

Kapitel 1	■ Sicherheit	3
<hr/>		
Kapitel 2	■ Einleitung	5
<hr/>		
Kapitel 3	■ Ein- und Ausgangsklemmen	9
<hr/>		
Kapitel 4	■ Grundlagen des SyncPos Programmes	21
	Schritt für Schritt den SyncPos	
	Motion Controller in Betrieb nehmen	29
	PID-Regelung optimieren	35
	Kurvenscheibensteuerung	42
	Nockenschaltwerk	52
<hr/>		
Kapitel 5	■ PC Software Benutzeroberfläche	53
	■ PC SW Benutzeroberfläche im MTC10 Modus	87
<hr/>		
Kapitel 6	■ Programmieren mit SyncPos	93
	Grundsätzliches zum Programmaufbau, zur	
	Befehlssprache, zum Compiler usw.	95
	■ Software-Referenz	102
	Alle Befehle in der Übersicht.....	102
	Alle Befehle von ACC bis #INCLUDE	105
	■ Parameter-Referenz	172
	VLT- und SyncPos-Parameter	172
	VLT-Parameter-Liste	176
	SyncPos-Parameter	177
<hr/>		
Kapitel 7	■ Meldungen und Fehler-Referenz	213
<hr/>		
Kapitel 8	■ Programmbeispiele	223
<hr/>		
Kapitel 9	■ Anhang	227
	Glossar	228
	Neues in der Software Version	229
	Technische Referenz	235
	Stichwortverzeichnis	239

Programmierbarer SyncPos Motion Controller für VLT[®]5000 und VLT[®]5000Flux

Die Gebrauchsanweisung beschreibt zwei Versionen des programmierbaren SyncPos Motion Controllers:

Bestellnummer	Beschreibung	Typ Code
175Z0833	SyncPos für VLT5000	A10 C0
175Z3029	SyncPos für VLT5000 mit Beschichtung	A10 C1
175Z3683	SyncPos für VLT5000Flux	A10 C0
175Z3684	SyncPos für VLT5000Flux mit Beschichtung	A10 C1

Die Grundfunktionen der beiden Versionen sind gleich, die einzigen Unterschiede sind:

- Die Flux-Version hat zwei zusätzliche Ausgänge auf der VLT5000Flux Steuerungskarte.
- Die Flux-Version unterstützt keine absoluten Drehgeber.

Dieses Handbuch gilt für folgende Software-Versionen:

SyncPos PC Software Version 6.5X
VLT5000/SyncPos Software Version 3.7X/5.0X
VLT5000Flux/SyncPos Software Version 5.XX/5.0X

© Danfoss A/S 2004

VLT ist eingetragenes Warenzeichen von Danfoss A/S.
IBM ist eingetragenes Warenzeichen von International Business Machines, Inc.
Microsoft, MS, MS-DOS, Windows, Windows NT und Wingdings sind entweder eingetragene Warenzeichen oder Warenzeichen der Microsoft Corporation in den USA und/oder anderen Ländern.



Der Frequenzumrichter steht bei Netzanschluss unter lebensgefährlicher Spannung. Durch unsachgemäße

Installation des Motors oder des VLT-Frequenzumrichters können ein Ausfall des Gerätes, schwere Personenschäden oder sogar tödliche Verletzungen verursacht werden.

Befolgen Sie daher stets die Anweisungen in diesem Handbuch sowie die jeweils gültigen nationalen bzw. internationalen Vorschriften und Sicherheitsbestimmungen.

■ Diese Bestimmungen dienen Ihrer Sicherheit

1. Bei Reparaturen muss die Stromversorgung des VLT-Frequenzumrichters abgeschaltet werden. Vergewissern Sie sich, dass die Netzversorgung unterbrochen und die erforderliche Zeit verstrichen ist, bevor Sie die Motor- und Netzstecker entfernen.
2. Die Taste [STOP/RESET] auf dem Bedienfeld des VLT-Frequenzumrichters unterbricht nicht das Versorgungsnetz und darf deshalb nicht als Notschalter bzw. Reparaturschalter verwendet werden.
3. Es ist dafür Sorge zu tragen, dass gemäß den örtlichen und nationalen Vorschriften eine ordnungsgemäße Erdung des Gerätes erfolgt, der Benutzer gegen Leitungsspannung geschützt und der Motor gegen Überlastung abgesichert ist.
4. Der Ableitstrom gegen Erde ist höher als 3,5 mA.
5. Ein Überlastungsschutz des Motors ist in der Werkseinstellung nicht enthalten. Wenn diese Funktion gewünscht wird, Parameter 128 auf den Datenwert *ETR Abschaltung* oder Datenwert *ETR Warnung* einstellen.
Achtung: Diese Funktion wird bei 1,16 x Motornennstrom und Motornennfrequenz initialisiert

Für den nordamerikanischen Markt: Die ETR-Funktionen beinhalten Motorüberlastungsschutz der Klasse 20 gemäß NEC.

6. Die Stecker für die Motor- und Netzversorgung dürfen nicht entfernt werden, wenn der VLT-Frequenzumrichter an die Netzversorgung angeschlossen ist. Vergewissern Sie sich, dass die Netzversorgung unterbrochen und die erforderliche Zeit verstrichen ist, bevor Sie die Motor- und Netzstecker entfernen.
7. Beachten Sie bitte, dass der VLT-Frequenzumrichter mehr Spannungseingänge als L1, L2 und L3 hat, wenn Zwischenkreiskopplung (Zusammenschalten des DC-Zwischenkreises) und extern 24 V-DC installiert sind. Kontrollieren Sie, dass vor Beginn der Reparaturarbeiten alle Spannungseingänge abgeschaltet sind und die erforderliche Zeit verstrichen ist.

■ Warnung vor unbeabsichtigtem Anlaufen

1. Der Motor kann mit einem digitalen Befehl, einem Bus-Befehl, einem Sollwert oder "Ort-Stop" angehalten werden, obwohl der VLT-Frequenzumrichter weiter unter Netzspannung steht. Ist ein unbeabsichtigtes Anlaufen des Motors gemäß den Bestimmungen zur Personensicherheit jedoch unzulässig, so sind die oben genannten Stoppfunktionen nicht ausreichend.
2. Während der Programmierung des VLT-Frequenzumrichters kann der Motor ohne Vorwarnung anlaufen. Daher immer die Stopp-Taste [STOP/RESET] betätigen, bevor Datenwerte geändert werden.
3. Ist der Motor abgeschaltet, kann er automatisch wieder anlaufen, sofern die Elektronik des VLT-Frequenzumrichters defekt ist oder falls eine kurzfristige Überlastung oder ein Fehler in der Versorgungsspannung bzw. am Motoranschluss beseitigt wurde.



Warnung:

Das Berühren spannungsführender Teile – auch nach der Trennung vom Netz – ist lebensgefährlich.

Achten Sie außerdem darauf, dass andere Spannungseingänge, wie z.B. 24 V DC, Zwischenkreiskoppelung (Zusammenschalten eines DC-Zwischenkreises) sowie der Motoranschluss beim kinetischen Speicher ausgeschaltet sind.

Bei VLT 5001-5006 220 und 500 V Geräten: mindestens 4 Minuten warten
 Bei VLT 5008-5500 220 und 500 V Geräten: mindestens 15 Minuten warten
 Bei VLT 5001-5005 550-600 V Geräten: mindestens 4 Minuten warten
 Bei VLT 5006-5022 550-600 V Geräten: mindestens 15 Minuten warten
 Bei VLT 5027-5250 550-600 V Geräten: mindestens 30 Minuten warten

Kapitel 2

■ Einleitung 6

■ Über dieses Handbuch: Was steht wo? 6

■ Konventionen 7

■ Über dieses Handbuch: Was steht wo?

Nehmen Sie diese Gebrauchsanweisung bitte vollständig zur Kenntnis und, um mit dem System sicher und fachgerecht arbeiten zu können, beachten Sie bitte besonders auch die Hinweise und Achtungsvermerke.



Bezeichnet eine allgemeine Warnung



Bezeichnet einen wichtigen Hinweis



Bezeichnet eine Warnung vor Hochspannung

■ Kapitel Ein- und Ausgangsklemmen

Informieren Sie sich im Kapitel Ein- und Ausgangsklemmen und im VLT5000 Handbuch über die technischen Daten der Optionskarte, über die Anschlussbelegung der Klemmen und über die allgemeinen Anschlussbedingungen.

■ Kapitel Grundlagen des SyncPos Programmes

Lassen Sie sich das Prinzip von SyncPos, der Makro-Programmiersprache zunächst kurz erklären und informieren Sie sich dann – sofern Sie weder SyncPos noch das Arbeiten mit Windows kennen – über die Grundlagen der SyncPos-Programmes.

Schritt für Schritt die Steuerung in Betrieb nehmen
Dann nehmen Sie den VLT mit dem SyncPos Motion Controller Schritt für Schritt in Betrieb. Dieser Abschnitt bietet Ihnen einen schnellen, generellen Einstieg: vom Einschalten und Kennenlernen über die Inbetriebnahme der Steuerung mit den vorbereiteten Testprogrammen sowie den wichtigsten Grundeinstellungen.

PID-Regelung optimieren

In diesem Kapitel erfahren Sie alles über die Bedeutung und den Einfluss der Regelparameter und wie Sie mit einer oder mehreren Testfahrten die Steuerung optimieren können, zum Beispiel um bessere Positionierungsergebnisse oder kürzere Zykluszeiten zu erreichen. Wie Sie im einzelnen vorgehen steht im Abschnitt „Zehn Schritte zur optimalen Regelung“.

Kurvenscheibensteuerung und Nockenschaltwerk

Dieser Abschnitt erläutert die Grundlagen einer Kurvenscheibensteuerung und eines Nockenschaltwerkes. Mit den Beispielen können Sie das Editieren der Kurven und Programmieren der Steuerung im Detail nachvollziehen.

■ Kapitel PC-Software Benutzeroberfläche

Alle Menüs und Funktionen sind detailliert beschrieben. Zum Beispiel das Menü Entwicklung zum Austesten der neuen Programme, das Menü Steuerung für das Verwalten der Programme und das Einstellen der Parameter oder der CAM-Editor mit allen Funktionen zum Kurven editieren.

■ Kapitel PC-Software Benutzeroberfläche im MCT10-Modus

SyncPos verhält sich anders, wenn es nicht direkt durch Klicken auf die SyncPos-Programm-Schaltfläche, sondern indirekt durch Öffnen einer Programm- oder Konfigurationsdatei über das Motion Control Tool MCT10 gestartet wird. Der Hauptunterschied besteht darin, dass das gesamte Dateihandling ausschließlich durch das MCT10 stattfindet. Informieren Sie sich in diesem Kapitel über die Unterschiede in der SyncPos Benutzeroberfläche im MCT10-Modus, zum Beispiel wie man Dateien öffnet und sichert oder wie man eine CNF-Kurve editiert bzw. Parameter ändert.

■ Kapitel Programmieren mit SyncPos mit Software-Referenz und Parameter-Referenz

In diesem Kapitel finden Sie alles über die Programmierung und zu den Befehlen: Vom grundsätzlichen Aufbau der Befehle bis zur detaillierten Beschreibung aller Befehle von ACC bis #INCLUDE und schließlich der Parameter.

Sie können sich zuerst in der Übersicht informieren – die Befehle sind hier nach Gruppen sortiert, zum Beispiel Kontrollbefehle oder Befehle für die Synchronisation – um dann die einzelnen Befehle in der alphabetischen Beschreibung genau kennenzulernen.

Im nächsten Abschnitt sind alle Parameter – zunächst alphabetisch in der Übersicht – und danach im Einzelnen mit den Werkseinstellungen und Wertebereichen beschrieben.

■ Kapitel Meldungen und Fehler-Referenz

Das Kapitel beschreibt die Meldungen von der Steuerung und von der SyncPos-Benutzeroberfläche: zuerst in einer Tabelle nach Fehlernummern aufsteigend geordnet und danach im Detail mit möglichen Ursachen und deren Behebung.

■ Kapitel Programmbeispiele

Das Handbuch zeigt einige Programmbeispiele (viel mehr finden Sie in der Online-Hilfe), die Sie zum Kennenlernen ausprobieren oder direkt in Ihr Programm kopieren können. Sie können als Arbeitserleichterung die Beispiele oder Teile davon markieren und über den Zwischenspeicher in Ihr Programm kopieren.

■ Anhang

Ein kurzes Glossar erläutert die im Handbuch verwendeten Begriffe und Abkürzungen. Verschaffen Sie sich einen schnellen Überblick über Neues in der PC-Version 6.5x und in den entsprechenden Versionen des SyncPos Motion Controllers im Anhang. Dies sind vor allem die erforderlichen Erweiterungen für den Einsatz des SyncPos-Programmes im MCT10-Modus. In der Technischen Referenz finden erfahrene Anwender Detailinformationen in englischer Sprache zum Beispiel die „Array Structure of CAM Profiles“. Und mit einem ausführlichen Stichwortverzeichnis endet dieses Handbuch.

■ Konventionen

Die Informationen in diesem Handbuch sind weitestgehend systematisiert und typografisch folgendermaßen beschrieben:

Menüs und Funktionen

Menüs und Funktionen sind in Großbuchstaben geschrieben, zum Beispiel "STEUERUNG" → "PARAMETER".

Befehle und Parameter

Befehle und Parameterkennungen sind ebenfalls in Großbuchstaben geschrieben, zum Beispiel AXEND und KPROP; Parameter sind fettkursiv hervorgehoben, zum Beispiel **Proportionalfaktor**.

Variablen

Variablen sind klein geschrieben und wenn sie zitiert werden kursiv hervorgehoben, zum Beispiel timeout bzw. *timeout*.

Tasten

Die Namen von Tasten und Funktionstasten sind in eckigen Klammern gesetzt, zum Beispiel die Steuerungstaste [STRG]-Taste (oder nur [STRG]), die [ESC]-Taste oder [F1]-Taste.

Verweise

Verweise auf andere Textstellen sind im Handbuch unterstrichen und in der Online-Hilfe zusätzlich farbig markiert, zum Beispiel Befehlshilfe.

Kapitel 3

■ Ein- und Ausgangsklemmen	10
VLT-Steuerkartenklemmen	10
Optionskartenklemmen	11
Versorgungsspannungen	13
Drehgeberüberwachung	13
Layout der Optionskarte, DIP-Schalter	13
Technische Daten	15
Anschlussbeispiele	16

Es gibt zwei Schnittstellen zum SyncPos-Motion-Controller:

- 36 Klemmen auf der Optionskarte
- 24 Klemmen auf der VLT-Steuerkarte

■ VLT Steuerkartenklemmen

Die Steuerkartenklemmen können für Synchronisierungs- und Positionierungsfunktionen vorgesehen werden, sofern folgende Parametereinstellungen vorgenommen wurden:

Digitaleingänge 16, 17, 18, 19, 27, 29, 32 und 33

Mit dem Befehl IN des SyncPos-Anwendungsprogrammes können Sie jederzeit den Status der digitalen Eingänge auslesen.

Wenn die Parameter 300–303 und 305–307 auf "ohne Funktion" (Werkseinstellung) eingestellt sind, werden die Eingänge von der VLT-Steuerkarte ignoriert und können als Eingänge für die Optionskarte benutzt werden.

Parameter 304 kann nicht auf "Ohne Funktion" gesetzt werden. Daher hat der Eingang 27 immer eine Stopfunktion. Sie können dies verhindern, wenn Sie den Parameter 304 als "Motor Freilauf invers" gesetzt lassen und in Parameter 502 den "seriellen Eingang" wählen.

Analogeingänge 53, 54 und 60

Mit dem Befehl INAD des Sync-Pos-Anwendungsprogrammes können Sie jederzeit den Status der analogen Eingänge auslesen.

Wenn Sie die Parameter 308, 311 und 314 auf "Ohne Funktion" einstellen, werden die Eingänge von der VLT-Steuerkarte ignoriert und sie können weiterhin als Eingänge der Optionskarte genutzt werden.

VLT5000: Digital-/Analogausgänge 42 und 45

Die Ausgänge 42 und 45 können vom SyncPos-Anwendungsprogramm mit dem Befehl OUTDA gesteuert werden, wenn die Parameter 319 und 321 auf eine der folgenden Einstellungen gesetzt werden:

OPTION DIGITAL	[90]	Digitalausgang
OPTION 0 – 20 mA	[91]	Analogausgang (Werkseinstellung)
OPTION 4 – 20 mA	[92]	Analogausgang
OPTION 0 – 32000P	[93]	Pulsausgang

VLT5000Flux: Digitale/puls Ausgänge 26 and 46

Die Ausgänge 26 und 46 auf der VLT5000Flux Steuerungskarte können vom SyncPos-Anwendungsprogramm mit dem Befehl OUTDA gesteuert werden, wenn die Parameter 341 and 355 auf eine der folgenden Einstellungen gesetzt werden:
OPTION DIGITAL [90] Digitalausgang
OPTION 0...50000P [91] Pulsausgang

VLT5000Flux: Analoge Ausgänge 42 und 45

Die Ausgänge 42 und 45 auf der VLT5000Flux Steuerungskarte können vom SyncPos-Anwendungsprogramm mit dem Befehl OUTDA gesteuert werden, wenn die Parameter 319 und 321 auf eine der folgenden Einstellungen gesetzt werden:
OPTION 0 ... 20 mA [90] Analogausgang
0...20mA
OPTION 4 ... 20 mA [91] Analogausgang
4...20mA

Relaisausgänge 01 und 04:

Die Relaisausgänge können vom SyncPos-Anwendungsprogramm mit dem OUT Befehl gesteuert werden, wenn die Parameter 323 und 326 auf folgende Werkseinstellung gesetzt sind:
Steuerwort bit 11/12

Technische Daten

Die technischen Daten bzgl. der Steuerkartenklemmen finden Sie im VLT5000-Produktbuch.

MK3A Digitale Eingänge

I1	I2	I3	I4	I5	I6	I7	I8	24 V	COM

Digitale Ausgänge

MK3C

O1	O2	O3	O4	O5	O6	O7	O8	24 V	COM

MK3B Master / virtueller Master

5 V	COM	A1	$\bar{A}1$	B1	$\bar{B}1$	Z1	$\bar{Z}1$

Istwert

MK3D

5 V	COM	A2	$\bar{A}2$	B2	$\bar{B}2$	Z2	$\bar{Z}2$

■ Optionskartenklemmen

Es gibt zwei Drehgeberschnittstellen, die folgende Funktionen abdecken:

- Istwert Drehgeber-Eingang
- Master Drehgeber-Eingang / Virtueller Master-Ausgang

Es gibt 8 Digitaleingänge, 8 Digitalausgänge und Klemmen für 5-V- und 24-V-Versorgung. Die Funktionen und technischen Daten der Klemmen sind im Folgenden beschrieben.

Beschreibung der Klemmen

Es gibt 4 Klemmenblöcke, 2 zehnpolige und 2 achtpolige.

(Siehe Abb. oben)

Digitaleingänge

MK3A ist ein zehnpoliger Klemmenblock mit 8 Digitaleingängen (I1 – I8) und 2 Klemmen für 24V-Versorgung (siehe Seite 13 Versorgungsspannungen).

Die Digitaleingänge werden im SyncPos-Anwendungsprogramm benutzt und können deshalb frei programmiert werden. Außerdem können allen Eingängen Funktionen zugeordnet werden, und zwar über folgende Parameter:

I_BREAK (105)	Unterbrechen eines laufenden Programmes
I_CONTINUE (106)	erneutes Starten eines unterbrochenen Programmes
I_ERRCLR (107)	Reset-Fehler
I_NEGLIMITSW (47)	Schalter für negativen Grenzwert
I_POSLIMITSW (46)	Schalter für positiven Grenzwert
I_PRGCHOICE (104)	Wahl eines SyncPos-Programmes
I_PRGSTART (103)	Start eines SyncPos-Programmes
I_REFSWITCH (45)	Festlegung des Referenzschalters (HOME)

Nur zweien der Eingänge sind spezifische Parameterfunktionen zugeordnet:

- **I5** dient als Eingang für den externen Marker des Masters, wenn im Parameter SYNCMTYPM "Externer Marker" gewählt wird.
- **I6** dient als Eingang für den externen Marker des Slaves, wenn im Parameter SYNCMTYPS "Externer Marker" gewählt wird.

Digitalausgänge

MK3C ist ein zehnpoliger Klemmenblock mit 8 Digitalausgängen und 2 Klemmen für 24V-Versorgung (*siehe folgende Seite [Versorgungsspannungen](#)*).

Die Digitalausgänge werden durch das SyncPos-Anwendungsprogramm gesteuert und können deshalb frei programmiert werden. Außerdem können allen Ausgängen über folgende Parameter Funktionen zugeordnet werden:

- O_AXMOVE (64) Welle läuft
- O_BRAKE (48) Aktivieren der mechanischen
 Bremsen
- O_ERROR (108) Fehleranzeige

Drehgeberschnittstelle 1

MK3B ist ein achtpoliger Klemmenblock mit 6 Klemmen für die Drehgebersignale und 2 Klemmen mit 5V-Versorgung (*siehe folgende Seite [Versorgungsspannungen](#)*).

Die Drehgeberschnittstelle 1 kann für eine der folgenden zwei Funktionen benutzt werden:

- Master-Drehgebereingabe (inkremental oder absolut) zur Synchronisierung
- Virtuelle Master-Drehgeberausgabe (inkremental)

Die *Abbildung 1* zeigt die Funktion jeder Klemme in den drei möglichen Betriebsarten.

ACHTUNG!


Bei Einsatz der Funktion Virtueller Master muss in allen Optionen – ausgenommen die erste und letzte an das Netzwerk angeschlossene Station – die Abspannung ausgeschaltet sein (Schalter 1.3). Siehe auch [Anschlussbeispiele](#) auf Seiten 16 bis 19.

Klemme	A1	$\overline{A1}$	B1	$\overline{B1}$	Z1	$\overline{Z1}$
Inkrementale Eingänge	A ein	\overline{A} ein	B ein	\overline{B} ein	Z ein	\overline{Z} ein
Absolute Eingänge	Clk aus	\overline{Clk} aus	Daten ein	\overline{Daten} ein	Nicht belegt	Nicht belegt
Virtueller Master	A aus	\overline{A} aus	B aus	\overline{B} aus	Z aus	\overline{Z} aus

Abb. 1

Klemme	A2	$\overline{A2}$	B2	$\overline{B2}$	Z2	$\overline{Z2}$
Inkrementale Eingänge	A ein	\overline{A} ein	B ein	\overline{B} ein	Z ein	\overline{Z} ein
Absolute Eingänge	Clk aus	\overline{Clk} aus	Daten ein	\overline{Daten} ein	Nicht belegt	Nicht belegt

Abb. 2

Drehgeberschnittstelle 2

MK3D ist ein achtpoliger Klemmenblock mit 6 Klemmen für die Drehgebersignale und 2 Klemmen mit 5V-Versorgung (*siehe folgende Seite [Versorgungsspannungen](#)*).

Die Drehgeberschnittstelle 2 kann für eine der folgenden zwei Funktionen benutzt werden:

- Slave-Drehgebereingabe (inkremental oder absolut) zur Synchronisierung
- Istwert-Drehgebereingabe (inkremental oder absolut) zur Positionierung

Die folgende Übersicht zeigt die Funktion jeder Klemme in den zwei möglichen Betriebsarten.

(*Siehe Abb. 2*)

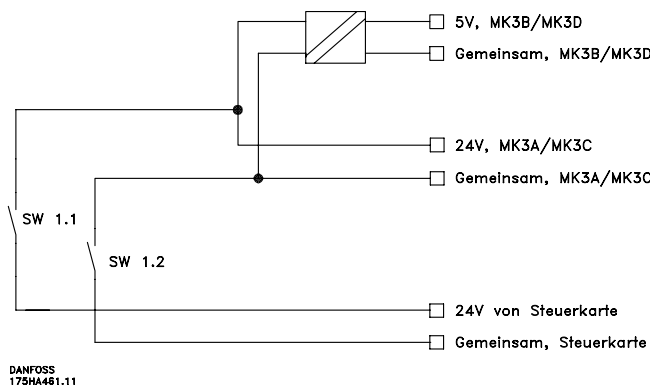
■ Versorgungsspannungen

Die Versorgung der Optionskarte erfolgt über die interne 24V-DC-Versorgung des VLT5000. Da jedoch die verfügbare Leistung begrenzt ist, kann es erforderlich sein, eine externe 24V-DC-Versorgung zu benutzen.

Die 24V-DC-Versorgung des VLT 5000 kann insgesamt 420 mA liefern, einschließlich der Last auf der Steuerkarte (Klemme 12, 13 und Ausgang 42 und 45).

Die 5V-Ausgangsleistung der Optionskarte wird von der 24V-Versorgung erzeugt. Die höchste Leistung auf der 5V-Seite ist $5\text{ V} \times 280\text{ mA} = 1,4\text{ W}$, dies entspricht etwa 60 mA auf der 24V-Seite.

Bei Einsatz einer externen 24V-DC-Quelle muss die interne 24V-Versorgung von der Steuerkarte abgeschaltet werden, und zwar durch Öffnen des Schalters SW 1.1 und 1.2.



Jeder Digitaleingang auf der Optionskarte nimmt 8 mA auf. Jeder Digitalausgang auf der Optionskarte kann je nach Last bis zu 0,7 A (bei externer 24V-Versorgung) liefern.

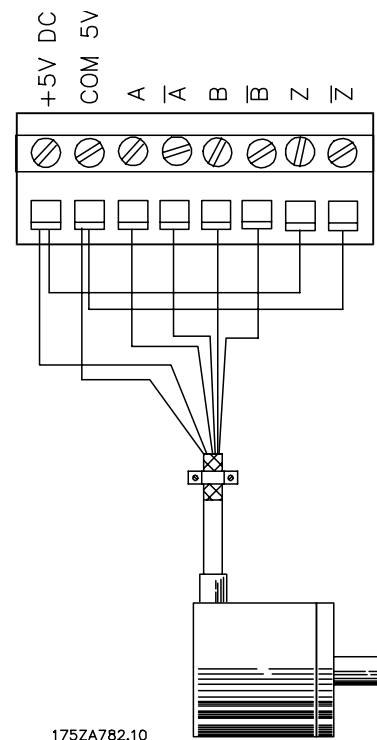
Die Belastung der 24V-Versorgung (intern oder extern) kann folgendermaßen berechnet werden:

$$\begin{aligned}
 &8\text{ mA} \times \text{Anzahl der Digitaleingänge} \\
 &+ \\
 &\text{Belastung der Digitalausgänge} \\
 &\quad (\text{mk3 C, O1 - O8}) \\
 &+ \\
 &\text{Belastung der 5V-Versorgung} \\
 &\quad (\text{mk3 B/D, 5 V/com}) \\
 &+ \\
 &\text{Belastung der Steuerkarte} \\
 &\quad (24\text{ V Versorgungsspannung, Klemmen 12/13} \\
 &\quad \text{und Ausgänge, Klemmen 42/45})
 \end{aligned}$$

■ Drehgeberüberwachung

Beide Drehgeber-Schnittstellen sind mit einem Überwachungsstromkreis versehen, der einen offenen Stromkreis wie auch einen Kurzschluss jedes Drehgeberkanals erkennen kann. Für jeden Drehgeberkanal zeigt ein LED den Status: Grün für OK und keine Anzeige für Fehler.

Die Überwachung des Null-Kanals kann mit Schalter 1.4 ausgeschaltet werden; das ist notwendig, wenn entweder inkrementale Drehgeber ohne Null-Kanal oder absolute Drehgeber benutzt werden. Der Schalter 1.4 schaltet die Überwachung der Null-Kanäle von Master- und Slave-Drehgeber aus. Wenn die Überwachung von nur einem der beiden Null-Kanäle ausgeschaltet werden soll (z.B. wenn ein inkrementaler Master-Drehgeber und ein absoluter Slave-Drehgeber benutzt werden), dann muss der nicht benutzte Null-Kanal-Eingang an 5V/common angeschlossen werden:



Die Anordnung der LED's und Schalter sehen Sie im nächsten Abschnitt [Layout der Optionskarte](#).

Nur wenn die Drehgeberüberwachung im Parameter MENCODER (Master) und/oder ENCODER (Slave) aktiviert ist, wird ein Drehgeber-Fehler ausgegeben, der dann als Option-Error die sog. ON ERROR Fehlerbehandlung auslöst.

Bitte beachten Sie: Die Überwachung des Master-Drehgebers ist ausgeschaltet, wenn der Schalter 1.3 auf "AUS" steht.

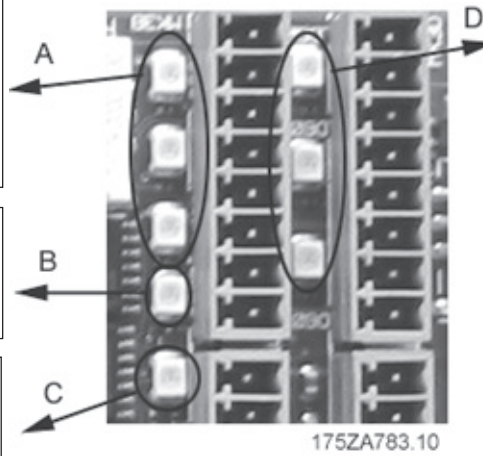
Layout der Optionskarte

Die Abbildung zeigt die Position der Anschlüsse und des Dip-Schalters.

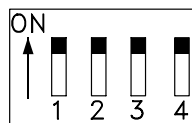
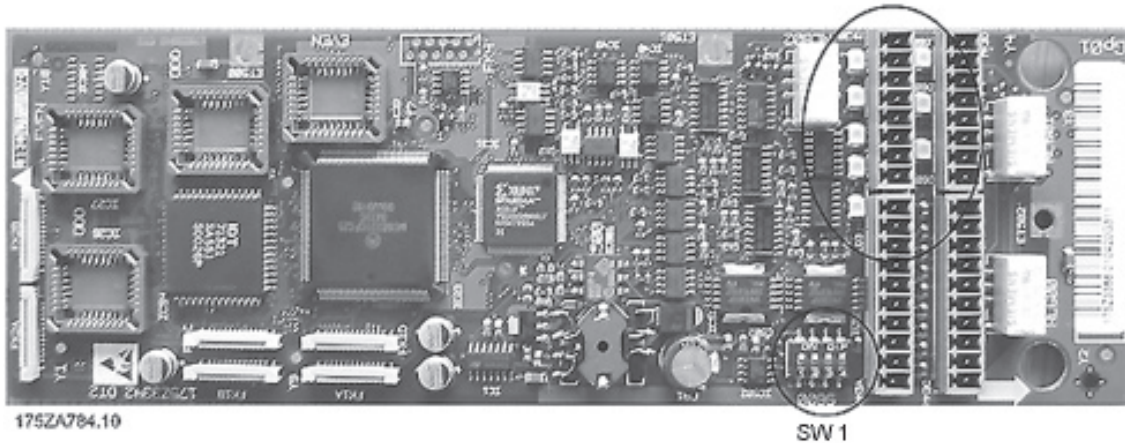
Master Drehgeber-Überwachung,
Kanal A, B und Z:
LED aus = Kurzschluss oder
offene Leitung
LED grün = Ok

5 V Überwachung:
LED aus = no 5V
LED grün = 5V ok.

CPU Überwachung:
LED muss mit 1 Hz blitzen, um ein
laufendes CPU System anzuzeigen



Drehgeber-Überwachung Slave,
Kanal A, B und Z:
LED aus = Kurzschluss oder
offene Leitung
LED grün = Ok



DANFOSS
175ZA068.10

- SW 1.1: Verbunden(ON)/nicht verbunden(OFF), 24 V von der Steuerkarte (siehe Beschreibung der Versorgungsspannung).
- SW 1.2: Verbunden(ON)/nicht verbunden(OFF), 24 V von der Steuerkarte.
- SW 1.3: Verbunden(ON)/nicht verbunden(OFF), Abschlusswiderstand für Master-Drehgeber (siehe Beschreibung der Funktion Virtueller Master).
ACHTUNG: Bei OFF ist die Master-Drehgeberüberwachung ausgeschaltet.
- SW 1.4: Schaltet für Master und Slave den Z-Kanal für die Drehgeber-Überwachung ON/OFF.

Die Werkseinstellung der Schalter 1.1. - 1.4 ist ON.



ACHTUNG!

Bei Einsatz der Funktion Virtueller Master muss in allen Optionen – ausgenommen die erste und letzte an das Netzwerk angeschlossene Station – die Abspannung ausgeschaltet sein (Schlt. 1.3). Siehe auch Anschlussbeispiele auf Seite 16 und 17.

■ Technische Daten
Klemmen:

Typ	Stecker mit Schraubverbindungen
Max. Kabelmaß	1,3 mm ² (AWG 16)

Digitaleingänge, MK3A:

Anzahl vom SyncPos-Programm benutzter Eingänge	8
Klemmenbezeichnungen	I1 – I8
Spannungsniveau	0 – 24 V DC (PNP positiv logisch)
Spannungsschwelle logisch "0"	5 V
Spannungsschwelle logisch "1"	10 V
Max. Spannung	28 V
Eingangsimpedanz	4 kΩ
Min. Signallänge (für ON INT)	1 ms

Galvanische Trennung: Alle Digitaleingänge sind mit Hilfe von Optoverbindern galvanisch isoliert, haben jedoch dieselbe gemeinsame Leitung wie die Digitalausgänge.

Digitalausgänge, MK3C:

Anzahl vom SyncPos-Programm benutzter Eingänge	8
Klemmenbezeichnungen	O1 – O8
Spannungsniveau	0 – 24 V DC
Max. Last	0,7 A (mit externer Stromversorgung)
Aktualisierungsintervall	1 ms

Galvanische Trennung: Alle Digitalausgänge sind mit Hilfe von Optoverbindern galvanisch isoliert, haben jedoch dieselbe gemeinsame Leitung wie die Digitaleingänge.

Externe 24 V DC Stromversorgung

(siehe VLT 5000 Handbuch, Seite 10)

Drehgebereingang 1, MK3B (Master):

Klemmenbezeichnungen	A1, A1, B1, B1, Z1, Z1
----------------------------	------------------------

Inkremental

Signalniveau	5 V differential
Signalart	Leitungstreiber, RS 422
Eingangsimpedanz	120 Ω (Dip-Schalter SW 1.3 = EIN/ON)
.....	> 24 kΩ (Dip-Schalter SW 1.3 = AUS/OFF)
Max. Frequenz	220 kHz (bei 50 % Arbeitszyklus)
Phasenverschiebung zwischen A und B	90° ±30°

Absolut

Signalniveau	5 V differential
Signalart	SSI
Protokoll	Gray code
Datenlänge	25 bit
Parität	keine
Taktfrequenz	105 oder 260 kHz
Max. Positionen je Umdrehung	8192
Max. Anzahl Umdrehungen	4096

Drehgebereingang 2, MK3D (Slave):

Klemmenbezeichnungen A2, A2, B2, B2, Z2, Z2

Inkremental:

Signalniveau 5 V differential

Signalart Leitungstreiber, RS422

Eingangsimpedanz 120 Ω

Max. Frequenz 220 kHz (bei 50% Arbeitszyklus)

Phasenverschiebung zwischen A und B 90° ±30°

Absolut:

Signalniveau 5 V differential

Signalart SSI

Protokoll Gray code

Datenlänge 25 bit

Parität keine

Taktfrequenz 105 oder 260 kHz

Max. Positionen je Umdrehung 8192

Max. Anzahl Umdrehungen 4096

Drehgeberkabel:

Kabeltyp Twisted pair und geschirmt. Bitte beachten Sie die Anweisungen des Drehgeber-Lieferanten.

Kabellänge Beachten Sie die Anweisungen des Lieferanten des Drehgebers.

*Der absolute Drehgeber wurde bis zu einer Kabellänge von 150 m mit 105 kHz Taktfrequenz und bis 100 m mit 262 kHz getestet. (Der Test wurde mit TR electronic Drehgeber Typ CE-65 M 8192*4096 und einem passenden Kabel nach TR electronic Vorschrift durchgeführt.)*

Maximal erlaubte Zeit zwischen Taktfrequenz und Datensignal, gemessen an den Steuerungsklemmen

beträgt 105 kHz Taktfrequenz = 9 µsec

..... 262 kHz Taktfrequenz = 3,5 µsec

Drehgeberausgang, MK3B:

Klemmenbezeichnungen A1, A1, B1, B1, Z1, Z1

Signalart Leitungstreiber, RS485

Max. Frequenz 150 kHz

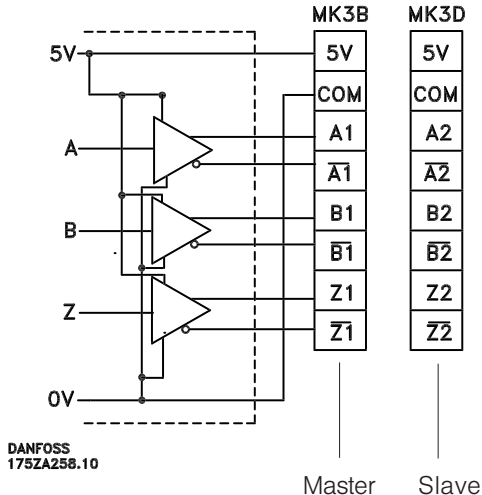
Min. Frequenz 150 Hz

Max. Anzahl Slaves 31 (mehr bei Einsatz von Repeatern)

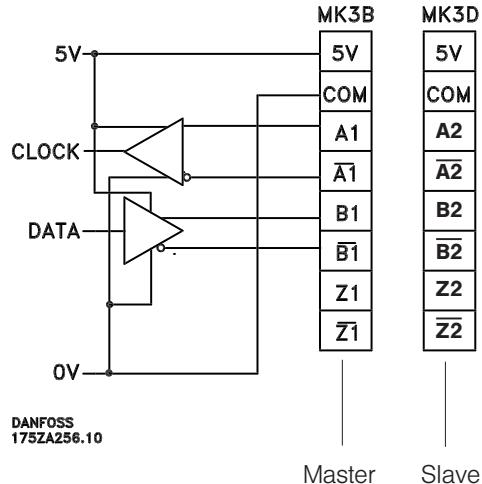
Max. Kabellänge 400 m

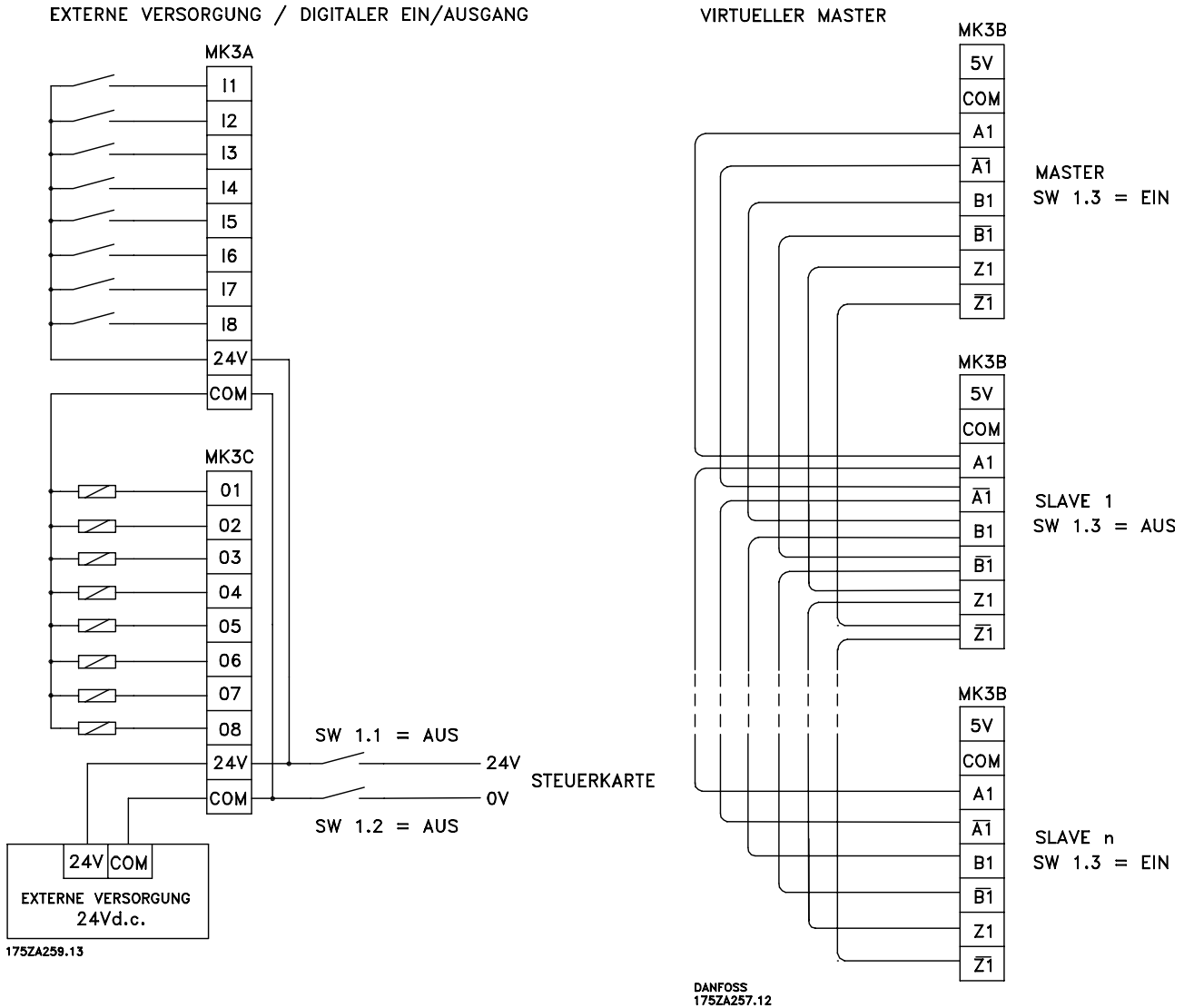
■ Anschlussbeispiele

INKREMENTALE DREHGEBER



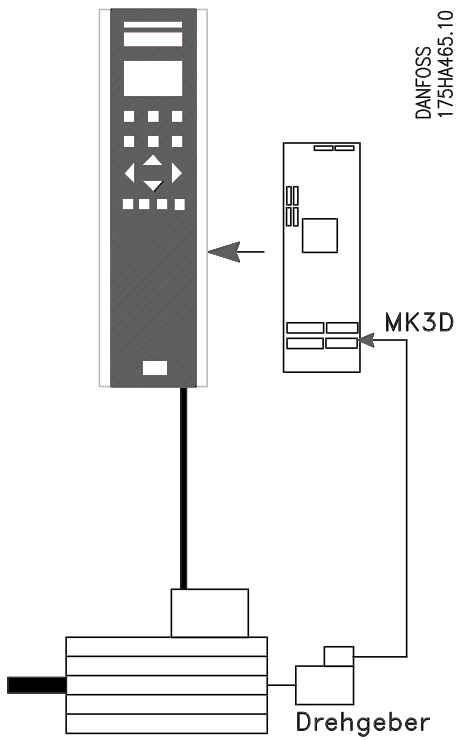
ABSOLUTE DREHGEBER





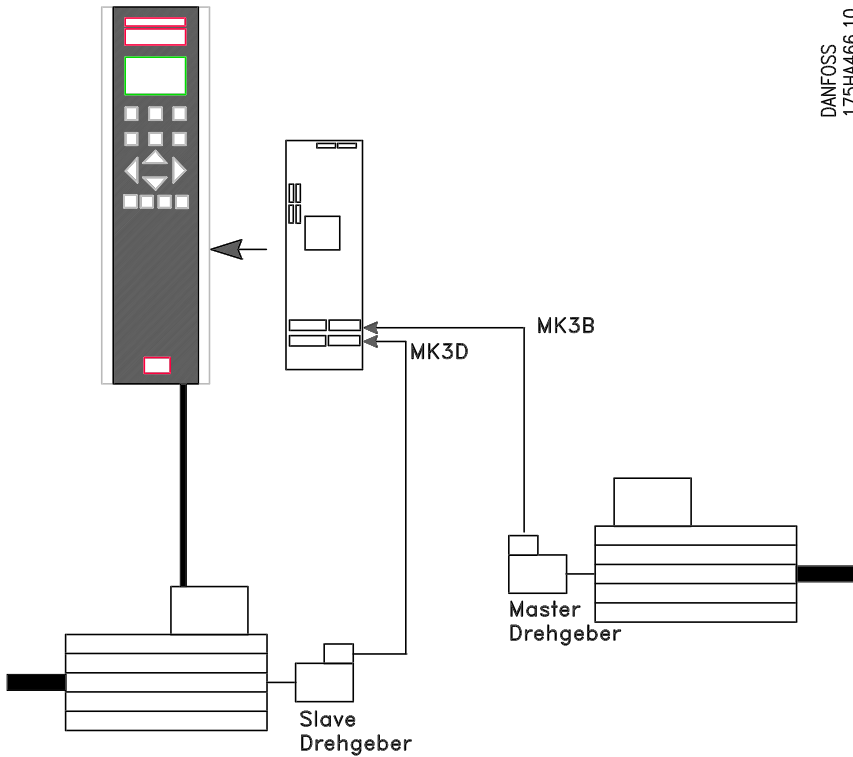
Die Abschlusswiderstände an den beiden Enden des Busses sind mit den Dip-Schaltern SW 1.2–4 zuzuschalten.

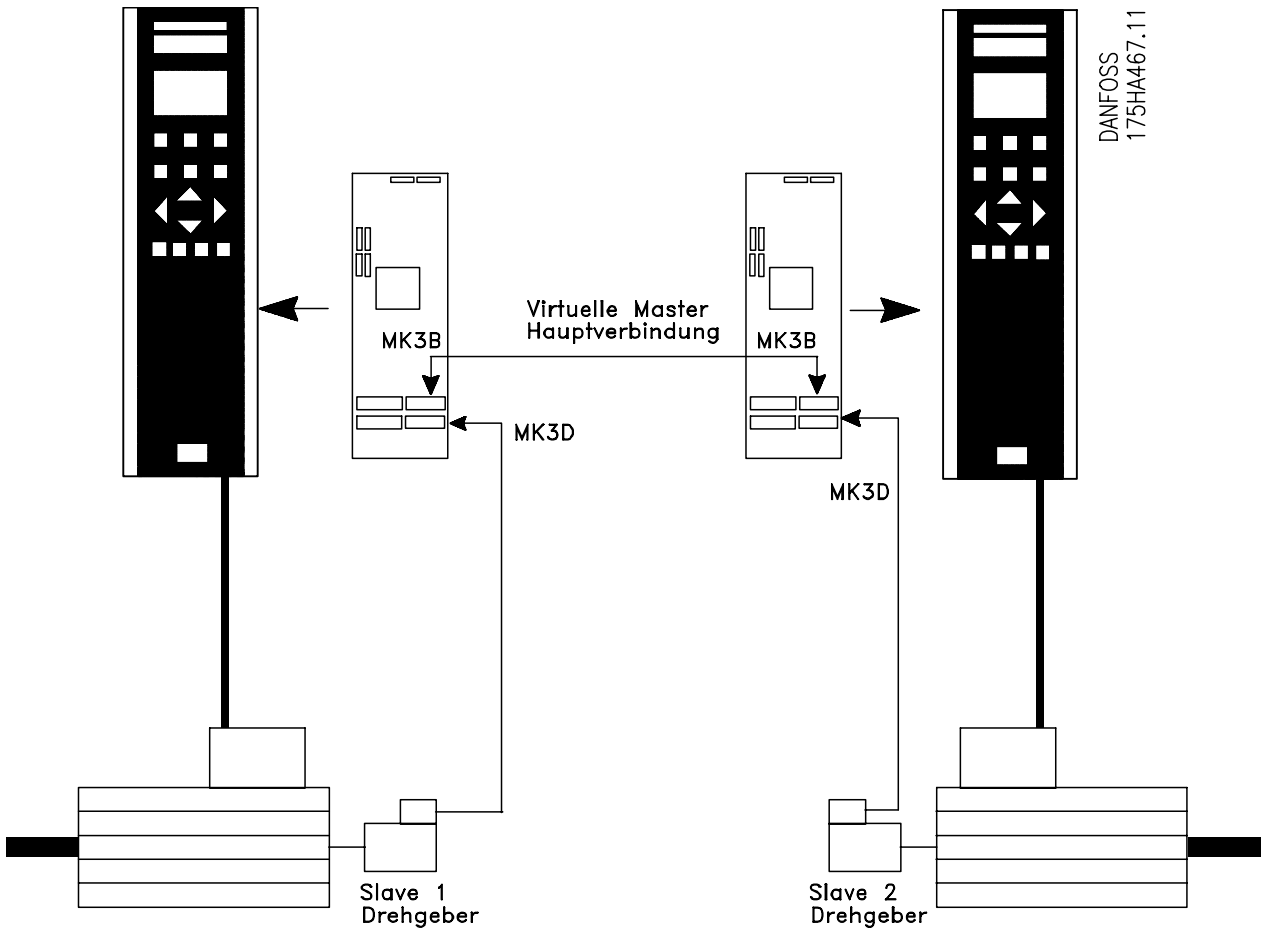
Ein- und Ausgangsklemmen



links: Drehgeberanschluss für Anwendungen für Positionierung

unten: Drehgeberanschluss für Master-/Slave-Synchronisation





Drehgeberanschlüsse für die Synchronisation mit virtuellem Master.

Kapitel 4

■ Grundlagen des SyncPos Programmes	
So funktioniert SyncPos	23
Voraussetzungen	24
Das SyncPos-Fenster	24
Mit der Maus arbeiten	26
Tastatur	26
Liste der Shortcuts	27
Liste der Funktionstasten	27
■ Schritt für Schritt den SyncPos Motion Controller Betrieb nehmen	
Sicherheitshinweise	28
SyncPos installieren	28
SyncPos starten	28
VLT Parameter einstellen	29
Kommunikation einrichten	29
SyncPos Parameter einstellen	29
Drehgeberanschluss und Motordrehrichtung prüfen	32
Fahrtst durchführen	33
■ PID-Regelung optimieren	
So funktioniert der Regelprozess	35
Bedeutung und Einfluss der Regelparameter	35
Schritt für Schritt die Reglerparameter optimieren	38
Was tun wenn	40
Beispiele zur Regleroptimierung	40
■ Kurvenscheibensteuerung (CAM-Mode)	
So funktioniert eine Kurvenscheibensteuerung mit SyncPos	42
Schnellkurs für Ungeduldige	43
Beispiel:	
Kartons mit Haltbarkeitsdatum stempeln	43
Beispiel:	
Kartons bedrucken mit Markerkorrektur	46
Wenn der Abstand des Sensors größer als eine Masterzykluslänge ist	48
Problemfälle bei der Festlegung des Markerabstandes	48
Beispiel: Slave-Synchronisation mit Marker	49
■ Nockenschaltwerk	
So funktioniert ein Nockenschaltwerk mit SyncPos	52
Beispiel eines Nockenschaltwerks	52

SyncPos ist ein auf Windows basierendes Entwicklungssystem mit einer speziell für Synchronisier- und Positioniersteuerungen konzipierten Programmiersprache, die leicht zu erlernen ist und sich bei den Befehlen an den bekannten Steuerungs-begriffen orientiert.

Diese Makrosprache ermöglicht es – selbst ohne genaue Kenntnis der Hardware-abläufe – komplexe Funktionen mit einfachen Befehlen zu realisieren und Fahrprogramme sowie allgemeine Steuerungsprogramme innerhalb kürzester Zeit zu erstellen.

Die Software SyncPos enthält alle für die Konfiguration, Programmierung, Optimierung und schließlich zur Befehlsübertragung an den SyncPos Motion Controller notwendigen Befehle und Menüs.

■ So funktioniert SyncPos

Lassen Sie sich das Prinzip von SyncPos kurz erklären:

Parameter bestimmen

Für alle Parameter sind im Programm Werkseinstellungen gespeichert, die nach der Auslieferung aktiv sind und durch ein Reset jederzeit wieder aktiviert werden können (siehe Seite 41).

Alle Parameter können Sie für Ihre Steuerung anpassen. Diese Benutzerparameter werden im EEPROM dauerhaft gespeichert und gelten für alle Programme.

Bevor Sie mit dem Programmieren beginnen, bestimmen Sie die grundlegenden Parameter des angeschlossenen VLT, wie **Maximalgeschwindigkeit** VELMAX (1) und **Kürzeste Rampe** RAMPMIN (31), stellen die PID-Filterwerte ein und definieren den **Benutzerfaktor** mit POSFACT_Z (23) und POSFACT_N (26).

Innerhalb eines Programmes können Sie mit dem Befehl, SET die Parameter vorübergehend ändern. Nach der Programmausführung werden sie wieder durch die gespeicherten Benutzerparameter ersetzt.

Programmieren mit der Makrosprache SyncPos

In dem Menü "BEARBEITEN" können Sie die Programme wie in einem Textprogramm erstellen und kommentieren. Alle Befehle sind im Kapitel Software-Referenz detailliert beschrieben.

Jeder Befehl besteht aus einem BEFEHLSWORT + ggf. **Parameter**, wobei der Parameter auch eine Variable, Konstante oder ein Array sein kann.

Der Kommentar steht zwischen /* ... */
oder nach //

zum Beispiel:

```

POSA 3000
/* Achse absolut auf Position 3000 fahren */
// Achse absolut auf Position 3000 fahren
    
```

Besonders einfach können Sie Ihr Programm mit Hilfe des Menüs "BEFEHLSHILFE" schreiben: Wenn Sie den Befehl auswählen, erhalten Sie sofort die notwendigen Eingabefelder eingeblendet. Nach der Eingabe der Werte wird automatisch die Syntax gebildet und Sie können den kompletten Befehl in Ihr Programm übernehmen.

Mit der Teach-in-Programmierung fahren Sie die Achse an die gewünschte Stelle und speichern einfach die erreichte Position. So können Sie schnell die kompliziertesten Verstell- und Bewegungsabläufe programmieren.

Programm ausführen und testen

Im Menü "ENTWICKLUNG" testen Sie neue Programme: Erst mit der Funktion "AUSFÜHREN" wird ein Programm in den VLT geladen und gestartet. Natürlich können Sie das Programm für den Test in "EINZELSCHRITTEN" ausführen oder erst ab einer bestimmten Stelle schrittweise ausführen lassen. Ein neues Programm wird vor jeder Ausführung automatisch auf korrekte Befehle geprüft. Oder Sie starten die "SYNTAXPRÜFUNG" zunächst ohne Programmablauf.

Programme im VLT speichern

Immer wenn Sie ein Programm ausführen oder die Syntaxprüfung starten, wird dieses Programm temporär im RAM in einem Bereich gespeichert, der mit jedem weiteren Test immer wieder überschrieben wird. Wenn Sie ein Programm für die SyncPos-Option fertiggestellt haben, können Sie das temporäre Programm im EEPROM dauerhaft speichern. Es erhält eine Nummer oder einem Namen und kann mit "AUTOSTART" gekennzeichnet werden, so dass es nach dem Einschalten des VLTs automatisch startet. Auf diese Weise kann die Steuerung offline betrieben werden.

Über die Programmnummern kann ein beliebiges Programm auch über die Eingänge, zum Beispiel von einer SPS aus, gestartet werden. Dazu sind die Eingänge mit "STEUERUNG" → "PARAMETER" → "GLOBAL" entsprechend zu setzen.

Steuerung mit den Reglerparametern optimieren

Die in der SyncPos-Option integrierte Lagereglereinheit berechnet bei jedem Bewegungsvorgang automatisch einen theoretischen Sollverlauf und versucht den VLT bzw. den Motor so zu steuern, dass eine möglichst gute Annäherung an den Sollverlauf erreicht wird. Mit den Reglerparametern können Sie direkt Einfluss darauf nehmen, wie stark und wie schnell einer Abweichung vom theoretischen Sollverlauf entgegengewirkt wird.

Diese PID-Filterparameter kann man auch theoretisch ermitteln, wenn man umfangreiche Kenntnisse über den gesamten Antrieb einschließlich der angeschlossenen Last hat. Wesentlich schneller und einfacher ist jedoch die experimentelle Methode mit den Funktionen im Menü "TESTFAHRT".

Nach jeder "TESTFAHRT" können Sie die Reglerparameter auch anhand von vier Grafiken beurteilen: Sie zeigen die Ist- und Sollkurven für die Geschwindigkeit, die Beschleunigung, die Position und den Stromverlauf. So können Sie sukzessive die PID-Filterparameter einstellen und die Steuerung optimieren.

Viel Erfolg bei der Arbeit mit SyncPos! Bei Fragen zur Programmierung wie zur Inbetriebnahme des VLT mit der SyncPos-Option wenden Sie sich bitte an Ihren Lieferanten.

■ Voraussetzungen

SyncPos ist auf handelsüblichen PCs mit Betriebssystemen ab Windows 95 bzw. NT 3.5 oder Windows 2000 einsetzbar.

Sie sollten die Grundlagen und Begriffe der Microsoft Windows-Oberfläche kennen, zum Beispiel die **Taskleiste** und den **Explorer**, wenn Sie mit Windows 9x arbeiten, bzw. Entsprechendes, wenn Sie WindowsNT einsetzen. Informieren Sie sich ggf. in den entsprechenden Windows-Handbüchern.

■ Das SyncPos-Fenster

Die folgende Abbildung erläutert die wichtigsten Elemente des SyncPos-Fensters.

Jedes Fenster stellt ein SyncPos-Programm dar, das mit einem VLT verbunden werden kann. Sie können also mindestens so viele Editierfenster öffnen, wie Sie Steuerungen bzw. VLTs auswählen.

Symbolleiste

Klicken Sie auf die Schaltflächen in der Symbolleiste, um schnell Funktionen auszuwählen.



Von links nach rechts: Datei neu, Datei öffnen, Datei sichern, Ausschneiden, Kopieren, Einfügen, Drucken, Info, Schnittstelle schließen und CAM-Editor.

Titelleiste zeigt den Namen der SyncPos-Datei und Nr. und Name des VLT und ggf. die Fehlernummer

Öffnen Sie die Menüs in der Menüleiste, um die SyncPos-Funktionen auszuwählen.

Menüs mit den Funktionen, die Sie mit der Maus markieren und auswählen.

Klicken Sie auf die Schaltflächen in der Symbolleiste, um schnell Funktionen auszuwählen.

Verwenden Sie die Bildlaufleisten (Scrollbars), um die Datei nach oben oder unten zu scrollen, bzw. nach links oder rechts zu scrollen.

Die blinkende Einfügemarke zeigt, wo der einzugebende Text erscheinen wird.

Dialogfeld

Kommunikationsfenster für Meldungen von Steuerung und Compiler

Die Statusleiste zeigt die Zeilennummer und Position, in der der Cursor steht, Informationen zu den Funktionstasten und ob die [NUM-Feststelltaste] oder die [Umschalt-Feststelltaste] gedrückt und damit aktiv ist.

Wenn Sie die Größe des Editierfensters bzw. des Kommunikationsfensters ändern wollen, stellen Sie den Cursor an den unteren Rand der Bildlaufleiste und klicken – sobald der Cursor seine Form ändert – und ziehen ihn in die gewünschte Richtung.

Menüs

Ein Häkchen bedeutet dass die Funktion aktiv ist, zum Beispiel im Menü "FENSTER" welche Dateien geöffnet sind.

Abgeblendete Funktionen sind nicht verfügbar bis Sie eine vorhergehende Aktion ausgelöst haben, zum Beispiel etwas in die Zwischenablage kopiert haben um es dann an anderer Stelle einzufügen, oder "VORBEREITEN EINZELSCHRITT" zum Auslösen jedes "EINZELSCHRITTS".

Sie können die meisten Menübefehle auch mit der Tastatur auswählen. Drücken Sie [ALT] und den im Menünamen unterstrichenen Buchstaben und anschließend den im Befehlsnamen unterstrichenen, z.B. [ALT] + [S] + [P] für "STEUERUNG" → "PROGRAMME".

Editierfenster

Im Editierfenster schreiben Sie Ihre Programme mit Hilfe der Funktionen des Menüs "BEARBEITEN" wie mit einem Texteditor. Verschiedene Farben erleichtern Ihnen die Unterscheidung zwischen Kommentaren, Programmteilen, Operatoren, Ziffern usw. Sie können die Farbzusammenhang mit "EINSTELLUNGEN" → "FARBEN EDITOR" ändern.

Dialogfelder

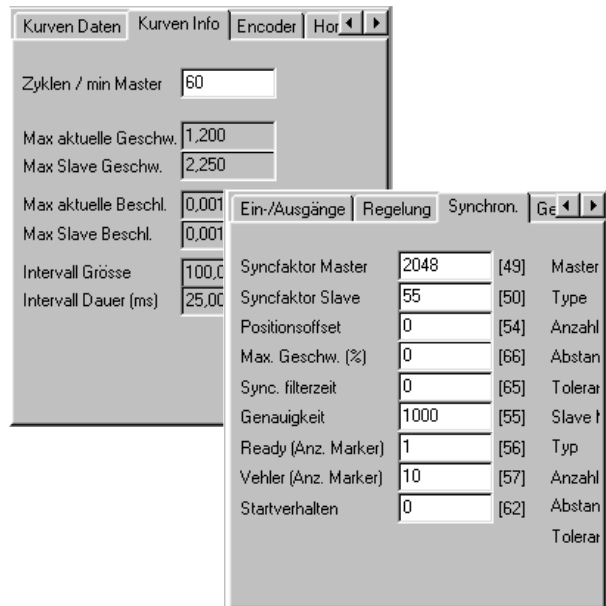
Nachdem Sie eine Funktion gewählt haben, erscheint häufig ein Dialogfeld, in dem Sie bestimmte Optionen festlegen können. Ist eine Option abgeblendet, ist sie für das aktuelle Verfahren nicht verfügbar.

Registerkarten

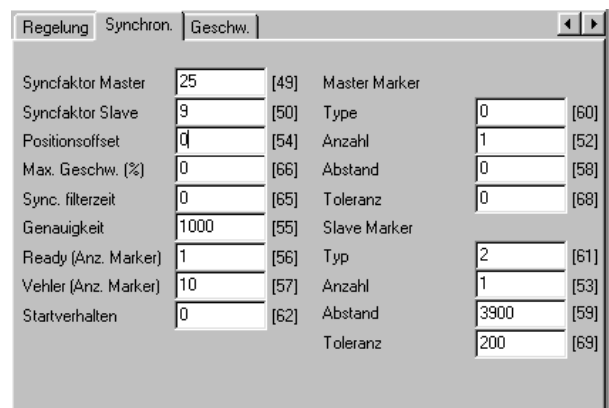
Mit den Registerkarten werden in der Windows-Benutzeroberfläche Dialogfelder in mehreren Ebenen für die Auswahl von Funktionen oder Eingabe von Werten bereitgestellt. Zum Beispiel im CAM-Editor die beiden Registerkarten "FIXPUNKTE" und "START-STOP-PUNKTE".

Klicken Sie auf den Namen der Registerkarte und diese Ebene wird nach vorne gestellt.

Bei den Registerkarten Kurven-Daten, Kurven-Info und Parameter klicken Sie auf die Bildlaufleisten (Scrollbars), um nach rechts zu scrollen und weitere Registerkarten darzustellen:



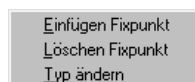
Wenn eine Registerkarte – zum Beispiel → "SYNCHRONISATION" nicht vollständig dargestellt wird, klicken Sie in ein beliebiges Eingabefeld und Sie sehen die komplette Registerkarte:



Dabei ändert sich automatisch die Einteilung des CAM-Editors. Wenn Sie auf "FENSTER" → "STANDARD", auf → Berechnen oder in ein anderes Feld im CAM-Editor klicken, wird sofort wieder das Standardfenster dargestellt.

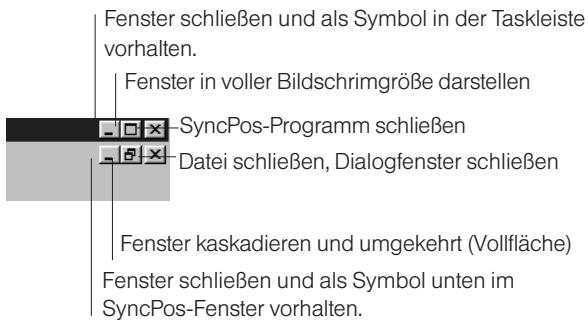
Popup-Menü

An manchen Programmstellen werden Popup-Menüs angeboten, wenn Sie auf die rechte Maustaste klicken. Zum Beispiel das Kontext-Menü im SyncPos-Editierfenster oder ein Auswahlmenü im CAM-Editor zum Bearbeiten von Fixpunkten. Die Popup-Menüs werden automatisch wieder verlassen, wenn die ausgewählte Funktion ausgeführt wird oder wenn Sie mit der linken Maustaste an eine beliebige andere Stelle im Bildschirm klicken.



Fenstergröße ändern und Fenster-Schalt symbole

Wenn Sie die Größe des SyncPos-Fensters ändern wollen, stellen Sie den Cursor an einen der äußeren Ränder des Fensters und ziehen – sobald der Cursor seine Form ändert – das Fenster mit gedrückter Maustaste in die gewünschte Größe. Die **Schalt symbole** rechts oben in jedem Fenster bewirken ...



■ Mit der Maus arbeiten

Falls Sie mit einer Maus mit mehreren Tasten arbeiten, ist die linke Maustaste grundsätzlich die „primäre“ Taste (sofern Sie die Konfiguration nicht geändert haben). „Klicken“ bedeutet, dass Sie die Maustaste kurz drücken und danach sofort wieder loslassen, ohne dabei die Maus zu bewegen. Wenn nicht anders angegeben, wird mit der linken oder primären Taste geklickt. „Ziehen“ bedeutet, dass Sie auf das Element zeigen und die Maustaste gedrückt halten, während Sie die Maus verschieben.

■ Tastatur

[ESC]-Taste

Neben den üblichen Funktionen einer [ESC]-Taste können Sie damit im Programm SyncPos jederzeit ein laufendes Programm abbrechen.



ACHTUNG!

Ein drehender Antrieb wird mit der maximal zulässigen Verzögerung abgebremst.

Pfeiltasten

Mit Hilfe der Pfeiltasten [↓], [↑], [←] und [→]-Taste bewegen Sie die Einfügemarke in einer Datei.

Richtungstasten

Mit den Richtungstasten [POS1] und [ENDE] stellen Sie den Cursor an den Anfang bzw. das Ende der Zeile und mit [Bild↑] und [Bild↓] an den Anfang bzw. Ende einer Bildschirmseite. Einige Richtungstasten können auch kombiniert werden, so stellen Sie zum Beispiel mit [STRG] + [POS1] den Cursor an den Dateianfang.

Zehnertastatur

Falls Sie über eine erweiterte Tastatur verfügen, können Sie Zahlen auch über die Zehnertastatur eingeben, wenn Sie vor der Zahleneingabe die [NUM-FESTSTELLTASTE] drücken.

Shortcuts

Tasten werden häufig als sog. Shortcuts mit anderen Tasten entweder als Tastenkombination oder als Tastenfolgen verwendet. Bei einer Tastenkombination müssen Sie die erste Taste gedrückt halten, während Sie die zweite drücken, z.B. [UMSCHALT] + [EINFG], um den Inhalt der Zwischenablage einzufügen. Bei Tastenfolgen drücken Sie die Tasten nacheinander, z.B. [ALT] + [D] um das Menü "DATEI" zu öffnen.

Funktionstasten

Häufig benötigte Funktionen sind auf die Funktionstasten gelegt, z.B. steuern Sie mit [F9] sehr effektiv das schrittweise Ausführen eines Programmes: Jedesmal, wenn Sie [F9] tasten, wird eine Zeile des Programmes abgearbeitet.

■ Liste der Shortcuts
Kopieren, Ausschneiden, Einfügen ...

Kopiert den markierten Text in die Zwischenablage	[STRG] + [EINFG]	oder [STRG] + [C]
Ausschneiden und Ablegen des markierten Textes in der Zwischenablage	[STRG] + [ENTF]	oder [STRG] + [X] oder [UMSCHALT] + [ENTF]
Fügt den Inhalt der Zwischenablage ein	[UMSCHALT] + [EINFG]	oder [STRG] + [V]
Zeile oberhalb Cursorposition einfügen	[STRG] + [UMSCHALT] + [N]	
Löschen des (restlichen) Wortes rechts von der Einfügemarke	[STRG] + [ENTF]	
Löschen des (restlichen) Wortes links von der Einfügemarke	[STRG] + [RÜCKTASTE]	
Ganze Zeile löschen	[STRG] + [Y]	

Cursor positionieren

Zum Dateiende springen	[STRG] + [ENDE]
Zurück zum Dateianfang	[STRG] + [POS1]
Gehe zu Zeile n	[STRG] + [G]
Bildlaufleisten zeilenweise nach oben oder unten schieben	[STRG] + [↑TASTE] bzw. [STRG] + [↓TASTE]
Bildlaufleisten nach links bzw. nach rechts schieben	[STRG] + [BILD↑] bzw. [STRG] + [BILD↓]

Erweitern einer Markierung ...

... um ein Zeichen nach rechts bzw. nach links	[UMSCHALT] + [→TASTE] bzw. + [←TASTE]
... bis zum Wortende	[STRG] + [UMSCHALT] + [→TASTE]
... bis zum Wortanfang	[STRG] + [UMSCHALT] + [←TASTE]
... bis zum Zeilenende, bzw. zum Zeilenanfang	[UMSCHALT] + [ENDE] bzw. + [POS1]
... um eine Zeile nach unten bzw. nach oben	[UMSCHALT] + [↓TASTE] bzw. + [↑TASTE]
... bis zum Dateianfang	[STRG] + [UMSCHALT] + [POS1] oder [UMSCHALT] + [BILD↑]
... bis zum Dateiende	[STRG] + [UMSCHALT] + [ENDE] oder [UMSCHALT] + [BILD↓]
Markiert den nächsten bzw. den vorherigen Befehl im Menü (bei angezeigtem Menü)	[↑TASTE] bzw. [↓TASTE]
Markiert das Menü auf der linken bzw. rechten Seite oder wechselt zwischen Haupt- und Untermenü, wenn ein Untermenü angezeigt wird	[←TASTE] bzw. [→TASTE]

Weitere Eingabehilfen

Rückgängigmachen der letzten Aktion (Datei speichern löscht den Undo-Speicher)	[ALT] + [RÜCKTASTE] oder [STRG] + [Z]
Im CAM-Editor: Rückgängigmachen der Eingaben jeweils bis zum vorhergehenden "BERECHNEN".	[ALT] + [RÜCKTASTE]
Im CAM-Editor: "WIEDERHERSTELLEN": Die Aktion des Befehls Rückgängig wird wieder zurückgenommen.	[ALT] + [UMSCHALT] + [RÜCKTASTE]

■ Liste der Funktionstasten

Programm abbrechen	[ESC]
Online-Hilfe aufrufen	[F1]
Im "SUCHEN"-Modus: Zwischen den markierten Fundstellen hin- und herspringen	[F2]
Im "SUCHEN"-Modus: Gleichen Begriff weitersuchen	[F3]
Startet "ENTWICKLUNG" → "AUSFÜHREN"	[F5]
Ganze Zeile markieren	[STRG] + [ALT] + [F8]
Startet im Modus "EINZELSCHRITT" jeweils eine Programmzeile bzw. im CAM-Editor das → "BERECHNEN" der Kurve	[F9]
"BEFEHLSHILFE" aufrufen	[F12]

■ Schritt für Schritt den SyncPos Motion Controller in Betrieb nehmen

Dieser Abschnitt bietet Ihnen einen schnellen, generellen Einstieg: vom Einschalten und Kennenlernen über die Inbetriebnahme des VLT5000 mit der SyncPos-Option mit den vorbereiteten Testprogrammen sowie den wichtigsten Grundeinstellungen.



ACHTUNG!

O.ERR 13 wird gleich nach dem Einschalten gemeldet, falls der VLT nicht bereit ist. Der VLT ist im Status „NICHT BEREIT“, wenn:

- der Motor abgeschaltet war und durch ein Reset wieder gestartet werden muss (Trip),
- der VLT nicht im lokalen Mode ist (Parameter 002 = Ort),
- der lokale LCP Stop aktiviert ist (die Anzeige leuchtet),
- kein Signal vom Eingang 27 kommt (Freilauf).

Ein Reset des Option Errors 13 ist nur mit dem ERRCLR Befehl oder durch "ABBRECHEN" [ESC] in der PC Software möglich und nur, wenn der VLT im Status „BEREIT“ ist, das bedeutet, dass keines der oben genannten Ereignisse vorliegt. Die VLT-Überwachung kann mit Parameter 700 [2] ausgeschaltet werden.

Bitte halten Sie sich an die folgende Schritt-für-Schritt-Anleitung:

1. SyncPos installieren und starten
2. VLT Parameter einstellen
3. Kommunikation einrichten
4. SyncPos Parameter einstellen
5. Drehgeberanschluss und Motordrehrichtung im Lokalmodus prüfen
6. Fahrttest durchführen
7. PID-Regelung optimieren

■ Sicherheitshinweise



Die Steuerung, bzw. der Motor müssen mit einem NOT-AUS jederzeit ausgeschaltet werden können.



Der Motor muss völlig frei drehen können, so dass auch ein plötzliches Rucken keinen Schaden anrichten kann.

Sie müssen außerdem die Sicherheitshinweise im Hardware-Handbuch kennen und beachten.

■ SyncPos installieren

Folgen Sie der Anweisung des Installationsprogrammes. SyncPos wird mit den Programmbeispielen im Verzeichnis „Program Files\Danfoss Drives\VLT Motion Control Tool“ installiert.

■ SyncPos starten

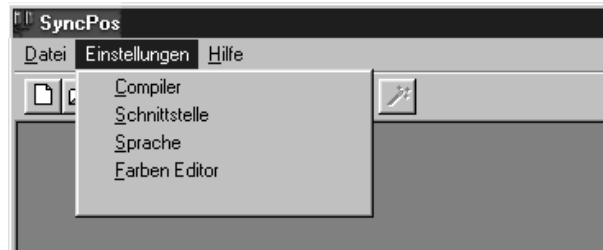


Schalten Sie die den VLT ein, jedoch auf jeden Fall ohne angeschlossenen Motor, bzw. muss der Motor spannungsfrei sein.

Klicken Sie in der Taskleiste auf "START" → "PROGRAMME" → SyncPos.

Dialogsprache ändern

Das SyncPos-Fenster ist nun – standardgemäß mit deutscher Dialogsprache – geöffnet.



Wenn Sie eine andere Sprache wünschen, klicken Sie – bevor Sie eine Datei öffnen – auf "EINSTELLUNGEN" → "SPRACHE" und wählen im darauf folgenden Dialogfeld zum Beispiel Englisch aus. Danach müssen Sie SyncPos schließen und neu starten.

Master Reset

Wenn Sie während des Einschaltens (Power-up) die Taste [CANCEL] am VLT drücken, wird die SyncPos-Optionskarte kein Programm starten, auch dann nicht, wenn entsprechende Startbedingungen vorliegen (Autokennung / Start-Eingang).

Die SyncPos-Optionskarte bleibt statt dessen im Idle-Modus und wartet auf neue Kommandos. Gleichzeitig wird der Fehler 19 Benutzer Abbruch ausgelöst.

SyncPos abbrechen

Ein Programm können Sie nur mit [ESC] abbrechen oder beenden. Dazu muss auch die Datei, die mit der Steuerung bzw. dem VLT verbunden ist, geöffnet sein bzw. wieder geöffnet werden.

■ VLT Parameter einstellen

Beim Booten werden die VLT-Parameter auf die Werkseinstellung eingerichtet. Es sind nur noch die Motorparameter einzustellen: Nutzen Sie dazu die Automatische Motor Anpassung AMA (siehe VLT5000/VLT5000 Flux Handbuch) oder passen Sie den VLT an den angeschlossenen Motor manuell an: Passen Sie die maximale Ausgangsfrequenz in Parameter 202 (Flux: maximale Ausgangsgeschwindigkeit) und den Maximalsollwert in Parameter 205 an die maximale Geschwindigkeit des Drehgebers an. Beachten Sie dabei, dass die maximale Ausgangsfrequenz höher sein muss, als die Frequenz der maximal zulässigen Geschwindigkeit der Achse, um Schlupf zu vermeiden.

Die Klemme 27 muss an 24 V angeschlossen oder der Parameter 502 auf "serieller Ausgang" eingestellt werden.

Wählen Sie die Funktionen der Ein- und Ausgänge in der Parametergruppe 3xx entsprechend den Anforderungen Ihrer Anlage aus. Beachten Sie, dass sich die Werkseinstellungen von einem Antrieb ohne Optionskarte unterscheiden.

Bitte beachten Sie auch, dass der Totzeitausgleich in Parameter 780 (Aktivierter Totzeitausgleich) auf AUS gesetzt ist (nicht bei Flux). Dieser Parameter verhindert Schwingungen im Stillstand.

■ Kommunikation einrichten

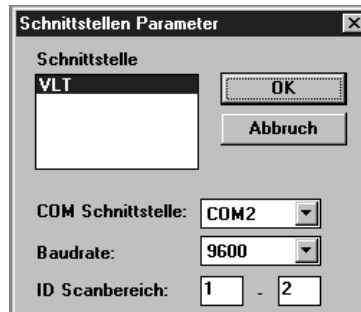
Bevor Sie beginnen ...

prüfen Sie, ob die Baudrate im VLT (Par. 501) eingestellt ist; bei einem seriellen Anschluss ist die Baudrate ab Werk fest eingestellt. Dann richten Sie die Kommunikation in Ihrem PC wie folgt ein:

Öffnen Sie eine vorhandene oder neue Datei. Klicken Sie auf "EINSTELLUNGEN" → "SCHNITTSTELLE".



Der VLT und die Baudrate sind voreingestellt. Klicken Sie auf OK.



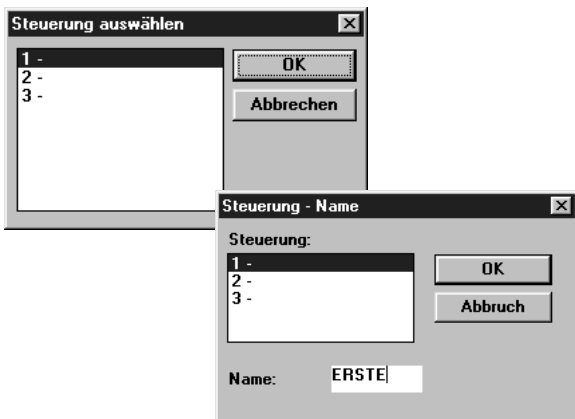
ACHTUNG!

Die Baudrate im VLT (Par. 501) und im Programm muss auf jeden Fall übereinstimmen.

"STEUERUNG AUSWÄHLEN"



Klicken Sie auf "ENTWICKLUNG" → "STEUERUNG AUSWÄHLEN", markieren Sie im folgenden Dialogfeld den VLT, den Sie in Betrieb nehmen wollen und klicken Sie auf "OK". Für jeden VLT, den Sie angeschlossen haben, erscheint automatisch die im Parameter 500 eingestellte Adresse im Dialogfeld.



Sie können zusätzlich zur Adresse im Menü "STEUERUNG" → "PARAMETER" → "NAME" für jeden VLT einen Namen eingeben.

RS485 Anschluss

Für einen RS485 Anschluss benötigen Sie die RS232-Standardschnittstelle im PC oder eine zusätzliche RS232-Schnittstellen-Karte und einen externen Wandler.

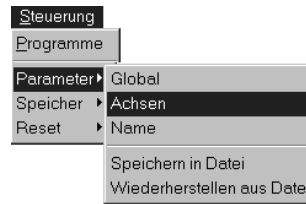
Kommunikation einrichten beenden

Im Kommunikationsfenster wird die erfolgreiche Verbindung gemeldet; in der Titelleiste der aktuellen Datei steht nun neben dem Dateinamen auch die Adresse und der Name des VLT.

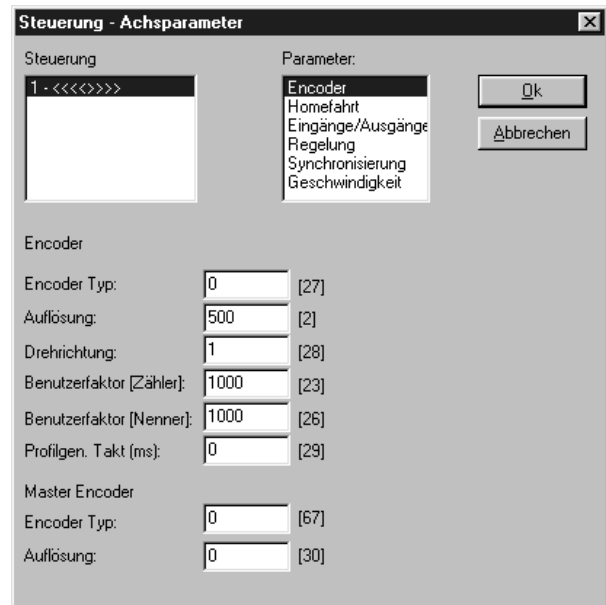
■ SyncPos Parameter einstellen

Die folgenden Parameter müssen immer geprüft und falls notwendig angepasst werden. Abhängig von den Anforderungen Ihrer Anwendung kann es notwendig sein, darüber hinaus noch weitere Parameter anzupassen.

Bei allen anderen Parametern können Sie zunächst die Werkseinstellungen benutzen und den VLT bei Bedarf später mit einer "TESTFAHRT" optimieren.



Klicken Sie auf "STEUERUNG" → "PARAMETER" → "ACHSEN" und wählen Sie die "STEUERUNG" aus, deren Einstellungen Sie gerade vornehmen. Dann markieren Sie im Feld **Parameter** die Parametergruppe, die Sie einstellen wollen, zum Beispiel **Encoder** und tragen die Werte in die entsprechenden Felder ein.



Klicken Sie auf "OK" um die neuen Parameterwerte in den VLT zu laden und gleichzeitig zu sichern.

Die detaillierte Beschreibung aller Globalen Parameter und Achsparameter lesen Sie im Kapitel Software-Referenz im Abschnitt Parameter-Referenz nach und über den Umgang mit den Dialogfeldern informieren Sie sich in Programmieren mit SyncPos im Abschnitt Menü "STEUERUNG" → "PARAMETER" → "ACHSEN".

■ SyncPos Parameter einstellen: Encoder

ENCODERTYPE (27)

Stellen Sie den Typ des verwendeten Drehgebers ein:

- 0 = inkremental
- 1 = absoluter Encoder, Standard ca. 262 kHz
- 2 = absoluter Encoder, ca. 105 kHz
- 3 = absoluter Encoder ohne Überlauf (linear), jedoch mit Fehlerkorrektur, ca. 262 kHz
- 4 = absoluter Encoder ohne Überlauf (linear), jedoch mit Fehlerkorrektur ca. 105 kHz
- 100 ... 104 = wie 0 ... 4, jedoch mit Überwachung des Encoders.

ENCODER (2)

Die Auflösung des Drehgebers (Geberstrichzahl) geben Sie in Pulsen pro Umdrehung ein.

Die folgenden zwei Parameter sind nur für Anwendungen mit Synchronisation relevant:

MENCODERTYPE (67)

Stellen Sie den Drehgebertyp des Masters ein:

- 0 = inkremental
- 1 = absoluter Encoder, Standard ca. 262 kHz
- 2 = absoluter Encoder, ca. 105 kHz
- 6 = Software-Simulation des Masters
- 100 ... 102 = wie 0 ... 2, jedoch mit Überwachung des Encoders

MENCODER (30)

Die Auflösung des Drehgebers des Masters geben Sie in Pulsen per Umdrehung ein.

■ SyncPos Parameter einstellen: Geschwindigkeit

Die beiden nächsten Parameter finden Sie in der Parametergruppe **Geschwindigkeit** im Menü "STEUERUNG" → "PARAMETER" → "ACHSEN":

VELMAX (1)

Maximalgeschwindigkeit in U/min der Achse, an der der Drehgeber angeschlossen ist.



ACHTUNG!

Zum Synchronisieren muss mindestens die Maximalgeschwindigkeit des Masters eingestellt werden.

Für eine Positionssynchronisation muss sie sogar höher sein als die Maximalgeschwindigkeit des Masters, damit der Slave die Position des Masters wieder aufholen kann.

Alle Geschwindigkeitsbefehle (VEL, CVEL) beziehen sich auf diesen Wert

RAMPMIN (31)

Die **kürzeste Rampe** gibt die Zeit von 0 bis zur Maximalgeschwindigkeit und umgekehrt an. Alle Beschleunigungs- und Bremsbefehle (ACC, DEC) beziehen sich auf diesen Wert.

■ SyncPos Parameter einstellen: Homefahrt

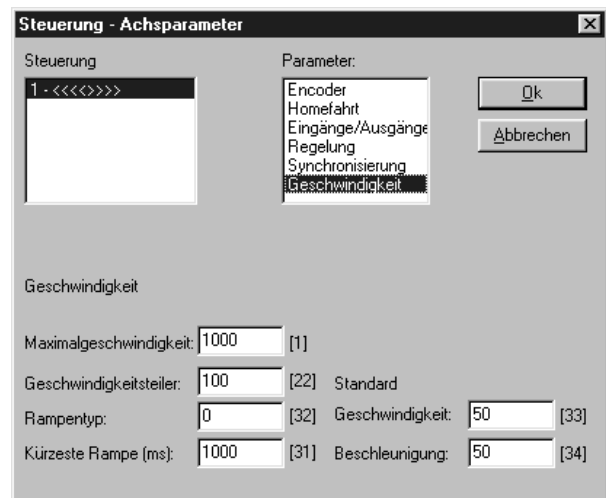
Bei Standard-Synchronisationen und Anwendungen mit Absolut-Drehgebern ist keine Homefahrt notwendig.

Bei Inkrementalgebern benötigt der Regler aber beim Einschalten eine Homefahrt. Dabei wird über einen Referenzschalter definiert, an welcher Stelle die Position 0 liegen soll und wie sich der VLT bei einer Homefahrt verhalten soll: Die Einträge sind von der Anwendung abhängig.

HOME_VEL (7)

Die **Home-Geschwindigkeit** wird in % bezogen auf die **Maximale Geschwindigkeit** des Antriebes eingegeben. Die Werte finden Sie in der Beschreibung des Motors.

■ SyncPos Parameter einstellen: Synchronisierung



Folgende zwei Parameter müssen nur für Anwendungen mit Synchronisation eingestellt werden:

Öffnen Sie im Menü "STEUERUNG" → "PARAMETER" → "ACHSEN" die Parametergruppe Synchronisierung.

Wenn externe Markersignale benutzt werden, können Sie den Markerabstand mit dem Programm "Marker count" (siehe Programmbeispiel im Kapitel 7) messen, falls er nicht bekannt ist.

SYNCFACTM (49) und SYNCFACTS (50)

Die **Synchronisationsfaktoren** für Master und Slave gleichen unterschiedliche Getriebefaktoren aus bzw. passen die Slavegeschwindigkeit im Verhältnis zur gegebenen Mastergeschwindigkeit an.

Beispiel: Die Geberstrichzahl von beiden Drehgebern ist 1024 (Pulse pro Umdrehung), der Masterantrieb läuft mit 305 U/min und der Slave mit 1220 U/min.

Dann beträgt SYNCFACTM = 305 und
 SYNCFACTS = 1220

Alternativ: SYNCFACTM = 1
 SYNCFACTS = 4

Alle nun folgenden Parameter müssen nur für Anwendungen mit Synchronisation mit Markerkorrektur (SYNCM Befehl) eingestellt werden:

SYNCMARKM (52) und SYNCMARKS (53)

Anzahl der Markerpulse des Masters und des Slaves.

SYNCMARKM und SYNCMARKS werden als Verhältnis zwischen der Anzahl der Markersignale des Masters und des Slaves eingegeben. Ein Verhältnis von 1:1 bedeutet, dass jeder Slavemarker mit jedem Mastermarker abgeglichen wird. Ein Verhältnis von 2:1 bedeutet, dass jeder Slavemarker auf jeden zweiten Mastermarker abgestimmt wird.

SYNCMPULSM (58) und SYNCMPULSS (59)

Wenn man den Encoder Indeximpuls als Markersignal benutzt, beträgt der Abstand zwischen zwei Markern die Auflösung (qc) des Drehgebers.

SYNCMTYPM (60) und SYNCMTYPS (61)

Markertyp für Master und Slave
 Master Markersignal: Input 5
 Slave Markersignal: Input 6

Wählen Sie den Signaltyp für Master und Slave:

- 0 = Indexpuls (positive Flanke)
- 1 = Indexpuls (negative Flanke)
- 2 = Externer Marker (positive Flanke)
- 3 = Externer Marker (negative Flanke)

■ Drehgeberanschluss und Motordrehrichtung prüfen

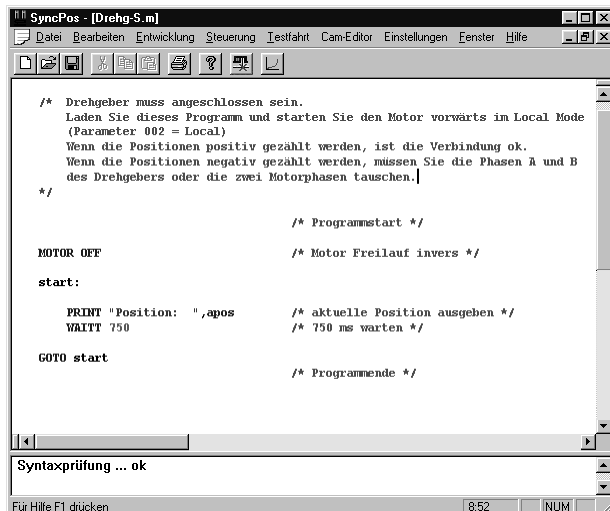
Schließen Sie nun – falls noch nicht geschehen – den Drehgeber an und testen Sie den Drehgeber.



ACHTUNG!

Schalten Sie aber die Versorgungsspannung aus, bevor Sie den Drehgeber anschließen.

Klicken Sie in der Menüleiste auf "DATEI" und "ÖFFNEN" Sie **Drehg-S.m**, das erste Testprogramm für die Inbetriebnahme.



Klicken Sie im Menü "ENTWICKLUNG" auf "AUSFÜHREN" und starten Sie so das Testprogramm. Wenn der Motor zum Beispiel im Local Mode (Parameter 002) vorwärts läuft, dann müssen die Positionen positiv gezählt werden. Falls die Positionen negativ gezählt werden, müssen Sie die A- und B-Kanäle des Drehgebers oder zwei Motorphasen tauschen.

Im Kommunikationsfenster wird die Position 0 gemeldet.

Wenn Sie jetzt mit der Hand den Motor drehen (der Motor darf nicht angeschlossen sein!) können Sie prüfen, ob der Drehgeber funktioniert: Im Kommunikationsfenster werden laufend die Positionen gemeldet. Bei einer vollen Umdrehung sollten Sie den 4-fachen Wert der Drehgeberauflösung erhalten, also 2000 wenn die Geberstrichzahl 500 ist.

Drehgeber prüfen bei einer Mastersynchronisation

Bei einer Anwendung mit Mastersynchronisation ändern Sie das Testprogramm wie folgt:
Ersetzen Sie den Befehl APOS durch MAPOS. Wenn nun der Master vorwärts läuft, müssen die Masterpositionen ebenfalls positiv gezählt werden.
Falls die Positionen negativ gezählt werden, müssen Sie die A- und B-Kanäle des Masters tauschen.

Drehrichtung prüfen

Auch hierfür ist der VLT-Parameter 002 einzustellen und der Motor mit lokalem Sollwert (Par. 003) zu fahren.

Durch Drehen an der Welle kann festgestellt werden, ob die Drehrichtung korrekt ist. Bei einer Rechtsdrehung von vorne auf das Wellenende gesehen, muss der Drehgeber hoch zählen.

Andernfalls müssen die Geberspuren A und B sowie \bar{A} und \bar{B} getauscht werden. Oft ist es auch einfacher, die zwei Motorphasen zu tauschen.

Oder Sie invertieren ganz einfach mit dem Parameter **Positive Drehrichtung** POSDRCT(28) die Bewertung der Drehgeberinformation.

Drehgebertest beenden

Beenden Sie den Drehgebertest mit der [ESC]-Taste und schließen Sie das Testprogramm mit "DATEI" → "SCHLIESSEN". Der erfolgreiche Drehgebertest ist Voraussetzung für die weitere Inbetriebnahme.

Wenn der Drehgeber nicht funktioniert ...

... könnte dies an einem falschen Kabelanschluss liegen. Messen Sie die Signale, die vom Drehgeber kommen und vergleichen Sie diese mit den in der Spezifikation geforderten Werten. Prüfen Sie, ob der Anschluss genau gemäß der Applikation erfolgte.)

■ **Fahrtst durchführen**

Schließen Sie nun den Motor an den VLT an und stellen Sie sicher, dass der Motor völlig frei drehen kann.



ACHTUNG!

Der Motor muss mit einem NOT-AUS versehen sein.

Klicken Sie auf "DATEI" und "ÖFFNEN" Sie die Datei **Fahrtst-S.m**.

```

DEF ORIGIN /* aktuelle Position als '0' definieren */
ACC 10 /* Beschleunigung auf 10% des Maximums setzen */
VEL 10 /* Geschwindigkeit auf 10% des Maximums setzen */

start: /* Schleife beginnt hier */
  POSA 500 /* Position '500' anfahren */
  WAITT 500 /* 0.5s warten, damit die Bewegung */
  /* abgeschlossen werden kann */
  PRINT "Position: ",apos /* aktuelle Position ausgeben */
  POSA 0 /* Position '0' anfahren */
  WAITT 500 /* 0.5s warten ... */

GOTO start /* das Ganze nochmal */
  
```

Syntaxprüfung ... ok

For Help, press F1 7:3 NUM

Klicken Sie auf "ENTWICKLUNG" und starten Sie mit "AUSFÜHREN" oder [F5] das Testprogramm.

Der Fahrtstest ist erfolgreich, wenn der Motor langsam hin- und herfährt und die Position 500 gemeldet wird.

Beenden Sie den Fahrtstest mit [ESC] und "SCHLIESSEN" Sie die "DATEI".

Motor fährt unkontrolliert los oder schwingt stark



ACHTUNG!

Schalten Sie den Motor sofort mit NOT-AUS aus, wenn er stark schwingt oder plötzlich unkontrolliert losfährt.

Wenn der Motor unkontrolliert losfährt, der Drehgebertest mit Prüfung der Drehrichtung des Motors vorher aber erfolgreich war, müssen Sie den **Proportionalfaktor** (Parameter 702) verringern. (Siehe PID-Regelung optimieren.)

Motor bewegt sich nicht

Wenn sich der Motor überhaupt nicht bewegt, ist wahrscheinlich der Proportionalwert des PID-Filters zu gering oder der VLT nicht freigegeben.

Wenn sicher ist, dass der VLT freigegeben ist (Klemme 27 = 24 V) und der VLT nicht am LCP gestoppt wurde, erhöhen Sie den **Proportionalfaktor** (Parameter 702). (Siehe PID-Regelung optimieren.)

Wenn der Motor stark schwingt ...

... müssen Sie die PID-Regelung optimieren und die anderen Parameter des Reglers anpassen:

Reduzieren Sie den **Proportionalfaktor** KPROP (11) oder erhöhen Sie den **Differentialfaktor** KDER (12).

Wenn Schleppfehler gemeldet wird

Wenn der Antrieb mit einer "Schleppfehler"-Meldung stehen bleibt, können Sie durch den Vergleich der Soll- und Istwertkurven ermitteln, ob der Antrieb in die falsche Richtung gedreht hat.

Prüfen Sie die Motor- und Drehgeberanschlüsse.

Wenn die Anschlüsse richtig sind, müssen Sie den **Tolerierten Positionsfehler** POSERR (15) erhöhen.

(Siehe PID-Regelung optimieren.)

■ PID-Regelung optimieren

■ So funktioniert der Regelprozess

Die im Kapitel PC Software Benutzeroberfläche erläuterte "TESTFAHRT" können Sie als Werkzeug benutzen, um die SyncPos-Steuerungsparameter zu optimieren und damit eine bestmögliche Leistung des System zu erreichen. Dazu müssen Sie nur einige Dinge über den Regelprozess von SyncPos wissen.

Die SyncPos Positioniersteuerung besteht aus zwei Teilen:

1. Der *Sollwert-Generator* wertet in SyncPos die verschiedenen Positionierbefehle aus und erzeugt eine Reihe von Sollpositionen, die schon in der gewünschten Position enden könnten. Alle Positioniervorgänge weisen normalerweise einen trapezförmigen Geschwindigkeitsverlauf auf. Das bedeutet, dass nach einer Phase mit konstanter Beschleunigung eine Phase mit konstanter Geschwindigkeit und zuletzt wiederum eine Phase mit konstanter negativer Beschleunigung folgt, die in der gewünschten Zielposition endet.
2. Die *PID-Regelung* erhält die Sollpositionen vom *Sollwert-Generator* und berechnet in Zyklen den Drehzahlsollwert, der für den Motor benötigt wird, damit er die aktuelle Sollposition erreicht. Mit den Reglerparametern können Sie direkt beeinflussen, wie stark und wie schnell einer Abweichung von dem theoretischen Sollverlauf entgegengewirkt werden soll.

Folgende Anzeichen deuten darauf hin, dass die Reglerparameter nicht optimal eingestellt sind:

- Antrieb schwingt
- Antrieb ist sehr laut
- häufiges Auftreten von Schleppfehlern
- schlechte Regelgenauigkeit



ACHTUNG!

Die Reglerparameter sind lastabhängig. Daher sollte der Antrieb unter den tatsächlichen Einsatzbedingungen optimiert werden.

In Ausnahmefällen kann es bei stark schwankenden Lastverhältnissen notwendig sein, verschiedene Sätze von Reglerparametern zu ermitteln und im späteren Anwendungsprogramm in Abhängigkeit vom Fahrvorgang umzuprogrammieren.

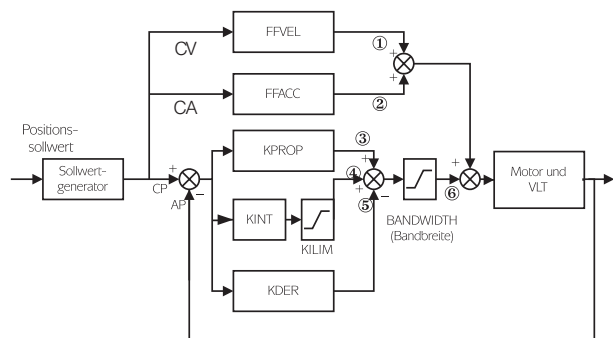
■ Bedeutung und Einfluss der Reglerparameter

Die Lagereglereinheit der SyncPos-Option gibt die notwendigen Ausgangsfrequenzen über einen internen Drehzahlsollwert an den VLT weiter. Dieser gesetzte Wert wird periodisch jede Millisekunde neu berechnet (das Intervall kann mit dem Parameter TIMER programmiert werden).

Die SyncPos-Option ist mit für die meisten Anwendungen passenden Regelparametern vorbereitet.

Die PID-Regelung arbeitet nach folgender Formel:

- ① = FFVEL * (Sollgeschwindigkeit)
- ② = FFACC * (Sollbeschleunigung)
- ③ = KPROP * (Positionsabweichung)
- ④ = KINT * (Summe aller vorhergehenden Positionsabweichungen) (begrenzt durch KILIM)
- ⑤ = KDER * (Geschwindigkeit der Positionsabweichung)
- ⑥ = ③ + ④ + ⑤ (begrenzt durch BANDWIDTH)



AV ist die aktuelle Geschwindigkeit

AP ist die aktuelle Position (berechnet durch den Drehgeber-Istwert) in qc (Quadcounts).

CP ist die aktuelle Sollposition in qc.

CV ist die Sollgeschwindigkeit in qc/ms.

Die Positionsabweichung wird durch CP-AP berechnet.



ACHTUNG!

Im Modus SYNCV arbeitet die PID-Regelung mit einer Drehzahl- statt einer Positionsänderung. Die Drehzahländerung wird durch CV-AV berechnet.

Zusammenfassung

Die Regelung in der SyncPos Optionskarte nutzt zwei Strategien gleichzeitig:

1. Eine Feed-forward-Regelung mit offener Regelschleife. Da ein asynchroner Motor von vornherein schon ein gutes Open-loop bewirkt, ist für die meisten Anwendungen die Feed-forward-Regelung ein wichtiger Teil der Steuerung.
Die Benutzung der Feed-forward-Regelung fördert eine sehr schnelle und exakte Reaktion auf die Änderungen der Sollposition.
2. Eine PID-Regelung mit geschlossener Regelschleife. Die PID-Regelung überwacht die Differenz zwischen der aktuellen Position und der Sollposition. Basierend auf dieser Information berechnet die PID-Regelung einen Sollwert um die Positionsabweichung zu minimieren. Daher ist die SyncPos-Option in der Lage, Änderungen der Last oder Reibung auszugleichen. Die PID-Regelung ist auch notwendig, um mögliche Positionsabweichungen auszugleichen, die durch eine ungenaue Open-loop Feed-forward-Regelung verursacht werden.

Die Feed-forward-Regelung wird eingesetzt, um Änderungen der Sollpositionen (besonders wichtig für Synchronisations-Anwendungen) zu handhaben, während die PID-Regelung benutzt wird, um Änderungen der Lastbedingungen oder Ungenauigkeiten der Feed-forward-Regelung auszugleichen.

Proportionalfaktor KPROP (11)

Der **Proportionalfaktor** wird mit der Positionsabweichung multipliziert und das Ergebnis zum Sollwert (dem internen Drehzahlsollwert für den VLT) addiert. Da der berechnete Sollwertanteil proportional zur Positionsabweichung (oder zum Positionsfehler) ist, wird diese Art der Regelung proportionale Regelung genannt.

Der **Proportionalfaktor** ähnelt in seinem Verhalten dem einer Feder, die um so weiter sie ausgedehnt wird, eine um so stärkere Gegenkraft erzeugt.

Einfluss des Proportionalfaktors:

- | | |
|----------------|---|
| KPROP zu klein | große Positionsabweichung durch nicht kompensierbare Last- und Reibungsmomente; |
| KPROP größer | schnellere Reaktion, kleinere statische Positionsabweichung, stärkeres Überschwingen, geringere Dämpfung; |
| KPROP zu groß | starke Schwingungen, Instabilität; |

Differentialfaktor KDER (12)

Der **Differentialfaktor** wird mit dem Differential der Positionsabweichung (der "Geschwindigkeit" der Positionsabweichung) multipliziert und das Ergebnis zum Sollwert addiert.

Der **Differentialfaktor** ähnelt in seinem Verhalten dem eines Dämpfers, der um so schneller er verstärkt wird, eine um so stärkere Gegenkraft erzeugt. Daher erhöht der **Differentialfaktor** die Dämpfung in Ihrem System.

Einfluss des Differentialfaktors:

- | | |
|--------------|---|
| KDER klein | keine Wirkung; |
| KDER größer | bessere Dämpfung, geringeres Überschwingen;
falls gleichzeitig KPROP erhöht wird: schnellere Reaktion auf Regelabweichung bei gleich starken Schwingungen; |
| KDER zu groß | starke Schwingungen, Instabilität; |

Integalfaktor KINT (13)

Die Summe aller Fehler wird jedes Mal berechnet, wenn der Sollwert erneuert wird. Der **Integalfaktor** wird dann mit der Summe aller Positionsfehler multipliziert und zum gesamten Sollwert addiert. Stellen Sie daher sicher, dass Sie den Integralanteil nutzen, wenn in Ihrer Anwendung statische Positionsabweichungen auftreten.

Statische Positionsabweichungen werden dadurch ausgeglichen, dass der aufaddierte Regelfehler so lange anwächst bis der Sollwert eventuell mit der Last übereinstimmt.

Durch das **Integrationslimit** kann der maximale aufaddierte Regelfehler begrenzt werden (anti-wind-up).

Einfluss des Integalfaktors:

- | | |
|-----------------|--|
| KINT sehr klein | statische Positionsabweichung wird sehr langsam zu Null ausgeglichen; |
| KINT größer | schnellere Regelung der statischen Positionsabweichung zu Null, stärkeres Überschwingen; |
| KINT zu groß | starkes Schwingen, Instabilität; |

Integrationslimit KILIM (21)

Das **Integrationslimit** begrenzt den Sollwert der durch den Integralfaktor erzeugt wird. Dadurch läßt sich das sogenannte "wind-up"-Problem verhindern, das typischerweise bei Anwendungen auftritt, bei denen der gesamte Sollwert (der interne Drehzahl-sollwert) so groß wird, dass es lange dauert, bis er wieder heruntergeregelt werden kann.

Das **Integrationslimit** ist auch sehr hilfreich, wenn die Motorspannung abgeschaltet ist und die Optionskarte den VLT steuert. Wenn man die Motorspannung während einer kleinen Positionsabweichung abschaltet (durch das Setzen der Klemme 27 auf low), kann ein enorm großer Sollwert entstehen, sobald die Motorspannung wieder anliegt.

Geschwindigkeits-Feed-forward FFVEL (36)

Der **Geschwindigkeits-Feed-forward** Faktor ist ein Skalierungsfaktor, der mit dem Differential der Sollposition (der Geschwindigkeit der Sollposition) multipliziert wird. Das Ergebnis wird zum gesamten Sollwert addiert.

Dieser Faktor ist besonders bei solchen Anwendungen nützlich, bei denen es eine gute Wechselbeziehung zwischen dem Sollwert (dem VLT Drehzahl-sollwert) und der Drehzahl des Motors gibt. Und das ist in den meisten Anwendungen mit einem asynchronen Motor der Fall.


ACHTUNG!

Die Skalierung des Parameters FFVEL ist von dem korrekten Setzen des maximalen Sollwertes (VLT Parameter #205) und der SyncPos Parameter VELMAX (1) und ENCODER (2) abhängig.

Beschleunigungs-Feed-forward FFACC (37)

Der **Beschleunigungs-Feed-forward** Faktor wird mit dem zweiten Differential der Sollposition (die Beschleunigung der Sollposition) multipliziert und das Ergebnis zum Sollwert addiert. Dieser Faktor sollte benutzt werden, um den Drehmoment auszugleichen, der zum Beschleunigen und Bremsen des Trägheitsmomentes benutzt wird.


ACHTUNG!

Die Skalierung des FFACC Faktors hängt von der eingestellten **Kürzesten Rampe** (31) ab. Sie sollten deshalb den FFACC Faktor entsprechend vergrößern, wenn Sie den SyncPos Parameter **Kürzeste Rampe** (31) verringern und umgekehrt.

Abtastzeit des gesamten Regelalgorithmus
TIMER (14)

Für besonders träge Systeme können Sie das gesamte Regelsystem verlangsamen, indem Sie für die Abtastzeit Vielfache von 1 ms eingeben. Allerdings gilt zu beachten, dass eine solche Änderung Einfluss auf sämtliche Regelparameter hat!

Daher sollte normalerweise von dem Wert 1 ms nicht abgewichen werden.

BANDWIDTH (35)

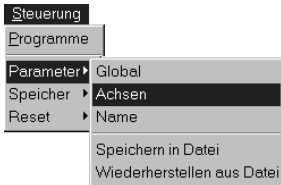
Eine **Bandbreite** von 1000 bedeutet, dass die Sollwerte zu 100 % ausgeführt werden, also **Differential-**, **Proportional-** und **Integralfaktor** wie definiert wirken.

Wenn Sie aber ein schwingungsgefährdetes System betreiben, zum Beispiel einen Kran mit schweren Lasten, können Sie die Bandbreite, in der der PID-Regler wirken soll, begrenzen. Ein BANDWIDTH von 300 bewirkt zum Beispiel eine Begrenzung auf 30 %: Das Aufschaukeln einer Schwingung wird dadurch verhindert, dass die Regelung nur mit 30 % des berechneten Sollwertes ausgeführt wird. Allerdings müssen Sie dann die Feed-forward-Anteile benutzen, um eine entsprechende Regelung zu erreichen.

■ Schritt für Schritt die Reglerparameter optimieren

Am besten nutzen Sie dazu auch die Funktionen im Menü "TESTFAHRT"; es hilft durch die grafische Darstellung der Soll- und Istkurven die PID-Regelung zu beurteilen und zu optimieren.

Sie sollten jedoch immer nur einen Wert ändern und die Besserung mit einer Testfahrt ermitteln.



Klicken Sie auf "STEUERUNG" → "PARAMETER" → "ACHSEN" und wählen Sie den VLT aus, den Sie gerade einstellen.

Regelverhalten festlegen

Bevor Sie die Reglerparameter anpassen, legen Sie fest, welches Regelverhalten erzielt werden soll.


ACHTUNG!

Die Antriebselemente dürfen auf keinen Fall außerhalb der technischen Spezifikation betrieben werden; damit wird die Maximalbeschleunigung vom „schwächsten“ Antriebselement bestimmt.

- „Steife“ Achse: Eine möglichst schnelle Reaktion wird hauptsächlich durch den **Proportionalfaktor** beeinflusst. Das Ergebnis beurteilen Sie anhand der Geschwindigkeitsgrafik.
- Die Dämpfung der Schwingungen wird hauptsächlich durch den **Differentialfaktor** beeinflusst. Das Ergebnis ist am besten in der Geschwindigkeitsgrafik zu beurteilen.
- Keine bleibende (statische) Positionsabweichung wird hauptsächlich durch den **Integralfaktor** erreicht und am besten anhand der Positionsgrafik beurteilt.

Zehn Schritte zur optimalen Regelung

Folgende Vorgehensweise optimiert Ihre Steuerung für meisten Anwendungen:

1. Stellen Sie sicher, dass Sie die korrekten Werte für den VLT Parameter #205 festgelegt haben und ebenso für die SyncPos Parameter VELMAX (1), ENCODER (2) und RAMPMIN (31). Falls Sie diese Einstellungen später einmal ändern, kann es sein, dass Sie die Steuerung erneut optimieren müssen.

2. Setzen Sie POSERR (15) sehr hoch, zum Beispiel auf 1000000, um zu verhindern, dass während der folgenden Testfahrten der Fehler O_ERR 8 auftritt.


ACHTUNG!

Sie dürfen POSERR aber nur so hoch setzen, wie es Ihre Anlage erlaubt (z.B. frei drehender Antrieb), denn bei diesen Werten ist die Schleppfehlerüberwachung außer Kraft gesetzt.

3. Optimieren Sie die Regelung des Geschwindigkeits-Feed-forwards:

Schritt 1)

Führen Sie eine "TESTFAHRT" mit den Parametern KPROP=0, KDER=0, KINT=0, FACC=0 und FFVEL=100 durch.

Schritt 2)

Beurteilen Sie das Geschwindigkeitsprofil.

Wenn die aktuelle Geschwindigkeitskurve niedriger ist, als der Sollkurve, klicken Sie auf "WIEDERHOLEN" und erhöhen im Dialogfeld **Testfahrt-Parameter** den Parameter FFVEL.

Wenn die aktuelle Geschwindigkeitskurve aber höher als die geforderte Sollkurve ist, verringern Sie natürlich FFVEL.

Schritt 3)

Führen Sie sukzessive Testfahrten durch, bis die zwei Geschwindigkeitskurven in der Grafik den gleichen Maximalwert aufweisen.

Schritt 4)

FFVEL ist nun optimiert; sichern Sie diesen aktuellen Wert.

4. In Systemen mit einem großen Trägheitsmoment und/oder schnellen Änderungen der Sollgeschwindigkeit ist es sinnvoll, den Beschleunigungs-Feed-forward zu nutzen (stellen Sie sicher, dass die Last angeschlossen ist, wenn Sie diesen Parameter optimieren):

Schritt 1)

Führen Sie eine "TESTFAHRT" mit KPROP=0, KDER=0, KINT=0, FFACC=0 und FFVEL mit dem bereits optimierten Wert durch. Verwenden Sie die höchstmögliche Beschleunigung. Falls RAMPMIN (31) korrekt eingestellt ist, sollte für die Beschleunigung und das Bremsen jeweils ein Wert von 100 ausreichend sein. Beginnen Sie mit einem niedrigen Wert für FFACC, etwa 10.

Schritt 2)

Beurteilen Sie die Geschwindigkeitsgrafik. Wenn während der Beschleunigung die aktuelle Geschwindigkeit konstant geringer als die Sollgeschwindigkeit ist, klicken Sie auf "WIEDERHOLEN" und geben Sie einen höheren Wert für FFACC ein. Wiederholen Sie dann die "TESTFAHRT".

Schritt 3)

Führen Sie sukzessive Testfahrten durch, bis die zwei Geschwindigkeitskurven in der Grafik die gleichen Beschleunigungs- und Bremsrampen aufweisen.

Schritt 4)

FFACC ist nun optimiert, sichern Sie den aktuellen Wert.

5. Als Nächstes muss der maximal mögliche stabile Wert für den **Proportionalfaktor** des PID-Reglers gefunden werden.

Schritt 1)

Führen Sie eine "TESTFAHRT" mit den Parametern KPROP=30, KDER=0 und KINT=0 durch. Benutzen Sie für FFACC und FFVEL die bereits gefundenen optimierten Werte.

Schritt 2)

Betrachten Sie die Geschwindigkeitskurve. Wenn sie nicht schwingt, klicken Sie auf "WIEDERHOLEN" und erhöhen Sie KPROP.

Schritt 3)

Führen Sie sukzessive Testfahrten durch, bis die aktuelle Geschwindigkeitskurve leicht schwingt.

Schritt 4)

Verringern Sie diesen 'leicht instabilen' Wert von KPROP auf etwa 70 %. Sichern Sie diesen neuen Wert.

6. Um die Schwingungen, die durch den KPROP erzeugt werden, zu dämpfen, sollten Sie nun den **Differentialfaktor** optimieren.

Schritt 1)

Führen Sie eine "TESTFAHRT" mit KINT=0, KDER=200 durch. Setzen Sie FFVEL, FFACC und KPROP auf die bereits gefundenen optimierten Werte.

Schritt 2)

Führen Sie sukzessive Testfahrten mit ansteigenden Werten für den KDER Faktor durch. Zuerst werden die Schwingungen schrittweise geringer. Hören Sie auf KDER zu erhöhen, sobald die Schwingungen beginnen, stärker zu werden.

Schritt 3)

Sichern Sie den letzten Wert von KDER.

7. In jedem System, in dem eine Statische Positionsabweichung zu Null ausgeregelt werden soll, muss der **Integalfaktor** benutzt werden. Um diesen Parameter zu setzen, müssen Sie zwischen dem schnellen Erreichen der geforderten Regelung zu Null der Statischen Positionsabweichung (was gut ist) und dem Anwachsen des Überschwingens und der Schwingungen im System (was schlecht ist) abwägen.

8. Wenn Sie den **Integalfaktor** der PID-Regelung benutzen, vergessen Sie nicht KILIM so stark wie möglich zu reduzieren (natürlich ohne den KINT-Effekt zu verlieren), um die Schwingungen und das Überschwingen so gut wie möglich zu verringern.

9. Verringern Sie BANDWIDTH so stark möglich. Mit einer korrekten optimierten offenen Regelschleife kann BANDWIDTH auf mindestens 6 oder 12 % verringert werden.

10. Setzen Sie POSERR (15) wieder auf einen normalen Wert von zum Beispiel 20000.

Testfahrtparameter → "SICHERN"

Wenn Sie die "TESTFAHRT" abgeschlossen haben → "SICHERN" Sie die neuen Parameter als Benutzerparameter. Damit werden sie im VLT gespeichert und künftig für alle Programme genutzt.

■ Was tun wenn ...

... Neigung zur Instabilität besteht

Bei einer verstärkten Neigung zur Instabilität reduzieren Sie wieder den **Proportional-** und **Differentialfaktor**, bzw. setzen den **Integralfaktor** zurück.

... stationäre Genauigkeit gefordert wird

Wenn eine stationäre Genauigkeit gefordert wird, müssen Sie den Integralanteil erhöhen.

... Schleppabstand überschritten wird

Wenn die Testfahrt ständig durch die Meldung „Schleppabstand überschritten“ unterbrochen wird, setzen Sie den Parameter für den **Tolerierten Positionsfehler** POSERR (15) – innerhalb der tolerierbaren Grenze – so groß wie möglich.

Wenn die Schleppfehler während der Beschleunigungsphase auftreten, deutet das darauf hin, dass bei den vorhandenen Lastverhältnissen die eingestellte Beschleunigung nicht erreicht werden kann. Erhöhen Sie den **Tolerierten Positionsfehler** POSERR (15) oder ermitteln Sie eine dem Gesamtsystem angepasste Maximalbeschleunigung. Treten Schleppfehler erst nach der Beschleunigungsphase auf und lassen sich auch durch das Erhöhen des **Tolerierter Positionsfehlers** nur verzögern aber nicht eliminieren, deutet das darauf hin, dass eine zu hohe Maximaldrehzahl gewählt wurde. Ermitteln Sie eine dem Gesamtsystem angepasste Maximalgeschwindigkeit.

... die maximale Beschleunigung nicht erreicht wird

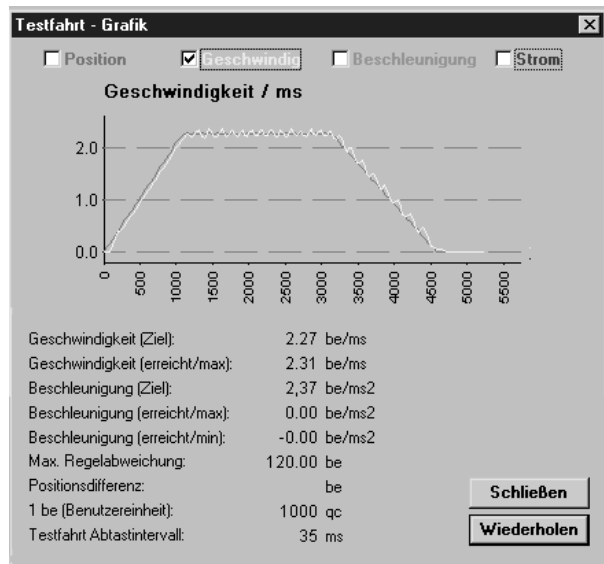
Die technischen Daten des Antriebes sind im allgemeinen nur für ein freidrehendes Wellenende gültig. Bei einem belasteten Antrieb verringert sich die maximale Beschleunigung.

Die theoretische Maximalbeschleunigung wird auch dann nicht erreicht, wenn zum Beispiel die PID-Regelung zu gering ist oder der VLT/Motor nicht passt und daher nicht genügend Spitzenstrom während der Beschleunigung liefert.

■ Beispiele zur Regleroptimierung

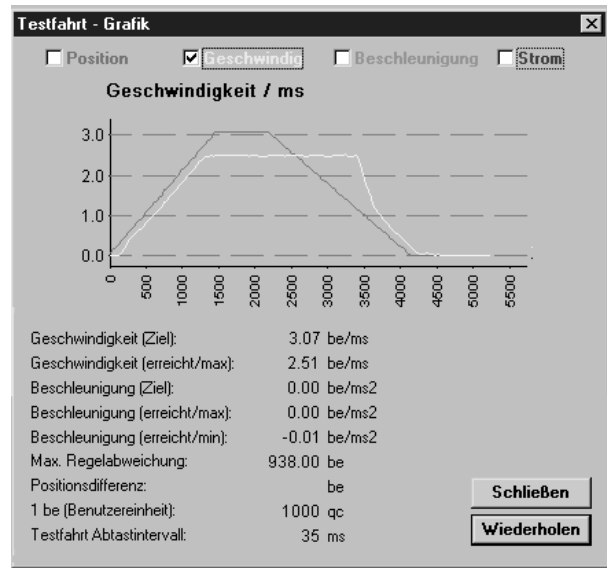
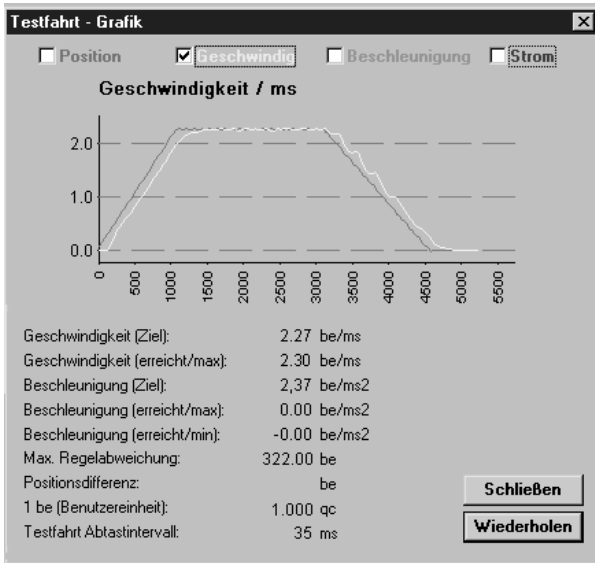
Dämpfung der Schwingungen

Der Antrieb schwingt:



Durch Erhöhen des **Differentialfaktors** auf 5000 und Reduzieren des **Proportionalfaktors** auf 3000 wird das Schwingen während der Fahrt und beim Abbremsen verringert.

Allerdings reagiert der Antrieb jetzt etwas träge auf die Geschwindigkeitsänderung, wodurch am Anfang der Beschleunigungsphase eine Abweichung von der Sollkurve auftritt:



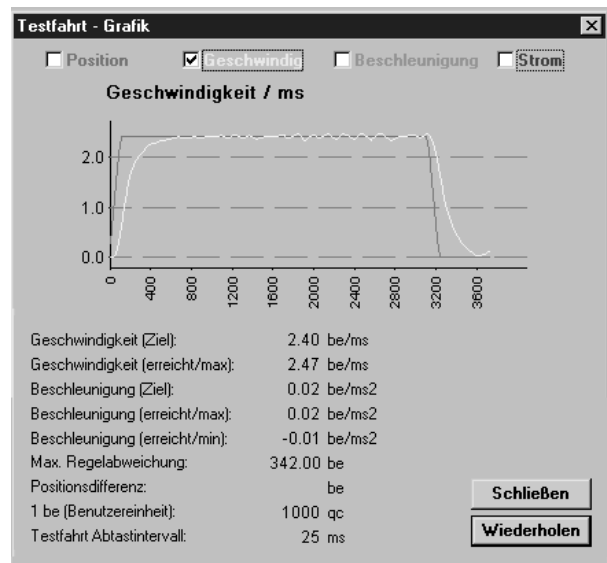
Durch Reduzieren der Geschwindigkeit auf 90 % der Maximalgeschwindigkeit und Erhöhen der Beschleunigung und Verzögerung auf 900 % erreicht der Antrieb die geforderten 2,47 Be/ms.

Bestimmung der Maximalgeschwindigkeit bzw. Maximaldrehzahl

Weg:	7500	Proportionalwert:	3000
Geschwindigkeit:	115	Differentialwert:	5
Beschleunigung:	80	Integralwert:	0
Brems-Beschl.:	60	Integrationslimit:	100
Abtastungen		Abtastzeit:	1
Anzahl:	150	Bandbreite:	1000
Intervall (ms):	35	FF Geschwindigkeit:	5
		FF Beschleunigung:	0

Weg:	7500	Proportionalwert:	3000
Geschwindigkeit:	90	Differentialwert:	5
Beschleunigung:	900	Integralwert:	0
Brems-Beschl.:	900	Integrationslimit:	100
Abtastungen		Abtastzeit:	1
Anzahl:	150	Bandbreite:	1000
Intervall (ms):	25	FF Geschwindigkeit:	5
		FF Beschleunigung:	0

Mit diesen Testparametern erreicht der Antrieb die Maximalgeschwindigkeit nicht und beginnt daher auch zu spät zu Bremsen, da er versucht die Sollposition zu erreichen:



■ Kurvenscheibensteuerung (CAM-Mode) und Nockenschaltwerk

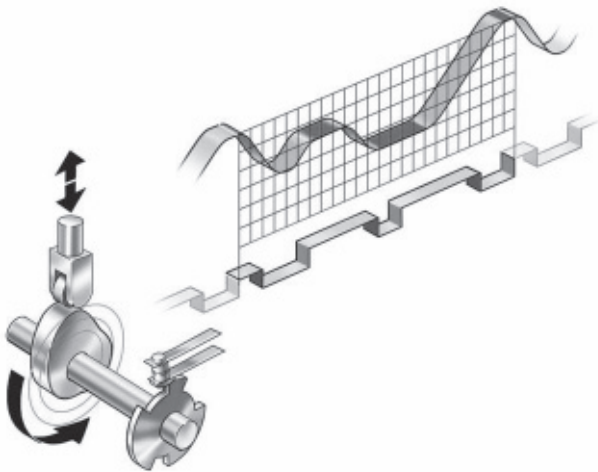
■ So funktioniert eine Kurvenscheibensteuerung mit SyncPos

Um Kurvenscheibensteuerungen zu realisieren, benötigen Sie je nach Anwendung mindestens eine Kurve, die die Slave-Position in Abhängigkeit von der Master-Position sowie das Ein- und Auskuppelverhalten beschreibt. Natürlich sind für eine Kurvenscheibensteuerungen weit mehr Parameter erforderlich, die zusammen mit den Fixpunkten der Kurve ein Kurvenprofil ergeben.

Die Synchronisation im CAM-Mode (SYNCC) können Sie auch mit Markerkorrektur durchführen (SYNCCMM und SYNCCMS). Dies wäre zum Beispiel erforderlich, wenn die Produkte unregelmäßig auf einem Band transportiert werden oder wenn addierende Fehler ausgeglichen werden müssen.

Prinzipskizze

Links die mechanische Kurvenscheibe und die mechanische Nockenwelle, rechts die Kurven für die elektronische Kurvenscheibensteuerung und das elektronische Nockenschaltwerk:



Für die Erstellung des Kurvenprofils nutzen Sie den → "CAM-EDITOR", in den Sie zuerst die bereits eingestellten VLT-Parameter laden. Dann setzen Sie die Fixpunkte der Kurve und definieren die für Ihre Anwendung erforderlichen Parameter. Alle Werte können Sie in physikalischen oder benutzerdefinierten Einheiten unter einer Windows-Oberfläche eingeben. Das Kurvenprofil können Sie ständig grafisch kontrollieren und so Geschwindigkeit und Beschleunigung der Slave-Achse prüfen.

Interpolation

Der CAM-Editor berechnet aus den Fixpunkten die Kurve mit Hilfe einer Spline-Interpolation. Diese ist auf minimalen Drehmoment optimiert. Um Drehzahlsprünge bei mehrmaligem Kurvendurchlauf zu verhindern, wird die Geschwindigkeit am Anfang und Ende gleichgesetzt. Für diese Berechnung können Sie zwischen zwei Kurventypen wählen. In beiden Typen berücksichtigt die Interpolation die Steigung der Kurve am Anfang und Ende: Bei Kurventyp 0 wird die Steigung am Anfang und Ende gemittelt, bei Kurventyp 1 wird die Steigung am Anfang der Kurve auch für das Ende der Kurve benutzt.

Tangentenpunkte für Geradestücke

Für Bereiche, in denen die Geschwindigkeit konstant und die Beschleunigung 0 sein muss, benutzen Sie Tangentenpunkte. Zwischen diesen Punkten wird statt eines Splines eine Gerade gelegt.

Genauigkeit

Die Fixpunkte werden direkt als Interpolationspunkte übernommen, sofern dies der Intervallabstand zulässt. Der CAM-Editor führt zwischen den Interpolationspunkten eine lineare Interpolation durch. Wird durch den gewählten Intervallabstand ein Fixpunkt nicht getroffen, ist der entsprechende Slave-Sollwert in der Interpolationstabelle nicht vorhanden. Wenn Sie → "AUSRICHTEN AN GITTER" aktivieren, können Sie solche Abweichungen vermeiden.

Interne Realisierung als Array

Intern werden die Kurvenprofile als Arrays realisiert, die Sie mit einer DIM-Anweisung und dem Befehl SETCURVE aufrufen.

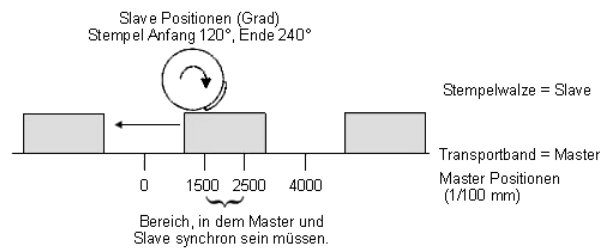
■ Schnellkurs für Ungeduldige

1. → "CAM-EDITOR" starten und VLT-Parameter als cnf-Datei in CAM-Editor laden: "DATEI" → "LADEN CNF".
2. Getriebefaktoren eingeben bzw. Benutzereinheiten MU und BE festlegen.
3. Mindestens drei → "FIXPUNKTE" für Master und Slave in der gleichnamigen Registerkarte eingeben.
4. → "START-STOP-PUNKTE" für das Ein- und Auskuppeln eingeben.
5. In der Registerkarte "KURVEN-DATEN" die → "ANZAHL INTERVALLE" für eine Masterzykluslänge definieren. Die → "INTERVALLZEIT" in der Registerkarte "KURVEN-INFO" sollte nicht kleiner als 20 ms sein.
6. In der Registerkarte "KURVEN-DATEN" die → "SLAVE-STOP-POSITION" definieren.
7. In der Registerkarte "KURVEN-INFO" die Anzahl der → "ZYKLEN / MIN MASTER" eingeben.
8. Mit Hilfe der grafischen Darstellung die Geschwindigkeit und Beschleunigung des Slaves prüfen.
9. Kurve als cnf-Datei → "SPEICHERN" und in den VLT laden.

■ Beispiel: Kartons mit Haltbarkeitsdatum stempeln

Das folgende Beispiel zeigt Ihnen, wie Sie Schritt für Schritt die Kurve für diese Anwendung der Kurvenscheibensteuerung editieren und anschließend in Ihr Steuerungsprogramm einbinden.

Eine Walze soll auf Kartons eine 10 cm lange Aufschrift stempeln. Der Stempel entspricht einem Walzenabschnitt von 120 Grad. Pro Minute werden 60 Kartons auf dem Band transportiert. Die Kartons werden exakt in immer gleichem Abstand (z.B. durch ein mechanisches Raster) auf dem Band transportiert. Während des Bedruckens müssen Stempelwalze und Karton synchron laufen:



Schritt für Schritt die Kurve editieren

1. VLT mit den erforderlichen Parameter einstellen und diese Benutzerparameter mit "PARAMETER" → "SPEICHERN IN DATEI" mit der Extension „cnf“ sichern.
2. Starten Sie den → "CAM-EDITOR" und laden Sie diese cnf-Datei mit "DATEI" → "LADEN CNF".
3. Ermitteln Sie den Getriebefaktor des Masters in MU-Einheiten:
Die Eingabe soll in 1/10 mm Auflösung möglich sein.
Der Antrieb ist mit dem Transportband mit einer Getriebeübersetzung von 25:11 verbunden; das heißt der Motor macht 25, das Zahnriemenrad 11 Umdrehungen.
Getriebefaktor = 25/11
Inkrementaldrehgeber direkt am Master-Antrieb;
Encoder-Auflösung = 4096
Das Zahnriemenrad hat 20 Zähne/Umdrehung, 2 Zähne entsprechen 10 mm, daher entspricht 1 Umdrehung = 100 mm
Transportbandvorschub bzw. 1000/10 mm
Skalierfaktor ist demnach 1000.

$$\frac{\text{Getriebefaktor} * \text{Encoderauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ MU}$$

$$\frac{25/11 * 4096 * 4}{1000} q_c = 1 \text{ MU}$$

$$\frac{25 * 4096 * 4}{1000 * 11} q_c = \frac{2048}{55} q_c = 1 \text{ MU} = \frac{\text{SYNCFACTM (49)}}{\text{SYNCFACTS (50)}}$$

Tragen Sie diese Werte in der Registerkarte → "SYNCHRONISATION" ein (die gewählten Einheiten sollten immer ganzzahlig sein):

Syncfaktor Master [49] = 2048

Syncfaktor Slave [50] = 55

4. Getriebefaktor des Slaves in BE-Einheiten eingeben:

Getriebefaktor = 5/1

Encoder-Auflösung (Inkrem.-Drehgeber) = 500
 Ein Umdrehung der Walze ist 360 Grad. Wir wollen mit einer Auflösung von 1/10 Grad arbeiten und teilen deshalb eine Umdrehung der Walze in 3600 Arbeitseinheiten ein:

Skalierfaktor = 3600

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} q_c = 1 \text{ BE}$$

$$\frac{5/100 * 500 * 4}{3600} q_c = 1 \text{ BE}$$

$$\frac{5 * 500 * 4}{3600} q_c = \frac{25}{9} q_c = 1 \text{ BE} = \frac{\text{POSFACT_Z (23)}}{\text{POSFACT_N (26)}}$$

Tragen Sie diese ganzzahligen Werte in die Registerkarte → "ENCODER" ein:

Benutzerfaktor Zähler [23] = 25

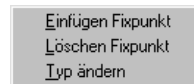
Benutzerfaktor Nenner [26] = 9

5. Damit die Fixpunkte auf den Interpolationspunkten liegen, bestimmen Sie in der Registerkarte → "KURVEN-DATEN" einen ganzzahligen Teiler für die Intervalle.
 Eine komplette Zykluslänge des Masters ist 400 mm; dies entspricht 4000 MU.
 Die → "ANZAHL INTERVALLE" = 40 ergibt eine vernünftige Intervallzeit von 25 ms.

6. Definieren Sie → "FIXPUNKTE" für das Transportband (Master) und die Walze (Slave) und den Punkttyp "1" für Kurvenpunkte. Klicken Sie auf den → "NEU RECHNEN"-Button, damit die Kurve dargestellt wird. Die Funktion → "AUSRICHTEN AN GITTER" sollte aktiviert sein.

Fix Punkte		Start Stop Punkte	
Punkt	Master	Slave	Typ
1	0	0	1
2	1500	1200	1
3	2500	2400	1
4	4000	3600	1
5			1
6			1
7			1

7. Zwischen der Position 1500 und 2500 müssen Master und Slave synchron mit gleicher Geschwindigkeit fahren. Dafür benötigen Sie eine Gerade, die mit zwei Tangentenpunkten bestimmt wird.
 Geben Sie für die beiden Positionen in der Registerkarte → "FIXPUNKTE" in der Spalte → "TYP" = 2 für Tangentenpunkt ein.
 Oder Sie bewegen den Cursor auf den Fixpunkt 2500, klicken auf die rechte Maustaste und wählen im darauf folgenden Popup-Menü → "TYP ÄNDERN". Da immer zwei Tangentenpunkte benötigt werden, wird der vorhergehende (linke) Punkt gleich mit geändert.



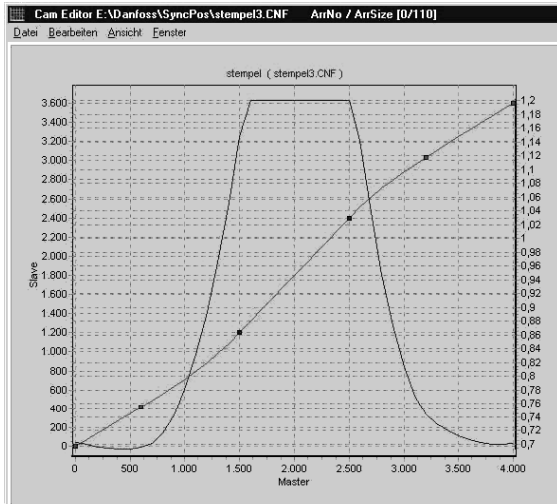
8. Die Kurve wird nicht angezeigt, weil zur Berechnung eines Splines immer drei Punkte benötigt werden. Vor und nach der Tangente (Geradestück) benötigen Sie in diesem Fall also noch jeweils einen Fixpunkt.

Punkte entweder in der Tabelle einfügen:

Sie klicken mit der rechten Maustaste in die Reihe der zweiten Fixpunkte und wählen im darauf folgenden Popup-Menü → "EINFÜGEN FIXPUNKT". Der Punkt wird zwischen diesem und dem vorhergehenden (links) eingefügt.

Oder in der Grafik einfügen: Sie bewegen den Cursor auf den (jetzt) fünften Punkt bis das Handsymbol erscheint. Sie klicken auf die rechte Maustaste und wählen im Popup-Menü → "EINFÜGEN FIXPUNKT".

Aktivieren Sie die grafische Darstellung der → "GESCHWINDIGKEIT" und verschieben Sie die beiden neu eingefügten Fixpunkte so lange mit der Maus, bis Sie einen gleichmäßigen Geschwindigkeitsverlauf erhalten:



9. Tragen Sie in der Registerkarte → "KURVEN-INFO" die → "ZYKLEN / MIN MASTER" = 60 ein. Das ist die Anzahl der Kartons, die (maximal) pro Minute bearbeitet werden.
10. Prüfen Sie, ob die Beschleunigung des Slaves innerhalb des Limits liegt. Aktivieren Sie dazu die Darstellung der → "BESCHLEUNIGUNG" und des → "BESCHL.LIMITS".
11. Um die Kurve in Ihre Steuerung zu laden, müssen Sie zuerst die "DATEI" → "SPEICHERN ALS .. CNF"-Datei. In der Titelleiste sehen Sie den Namen der Kurve und die Anzahl der Array-Elemente. Letzteres benötigen Sie für die DIM-Anweisung bei der Programmierung.

■ Kurve in den VLT laden

Mit "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI" laden Sie die cnf-Datei mit den veränderten Parametern und den – automatisch erzeugten – Kurvenarrays in den VLT.

■ Kurve in das Steuerungsprogramm einbinden

Da die Kurve intern als Array gespeichert wird, muss in Ihrem Programm als erstes die DIM-Anweisung stehen:

```
DIM stempel[92] // Anzahl der Elemente aus
                // Titelleiste des CAM-Editors
```

HOME

```
// Slave Achse führt eine Homefahrt durch
// (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der
// Nullposition (0 Grad)
// (entfällt bei Absolutgeber)
```

SETCURVE stempel

```
// Stempelkurve setzen
// angenommen ein Karton steht mit der
// Vorderkante am Bearbeitungspunkt
// und der Master steht still
```

```
DEFMCP0S 1000 // 1000 entspricht dieser
               // Position (Vorderkante Karton)
```

POSA CURVEPOS

```
// Slave auf die, der Master-Position
// entsprechenden Kurvenposition fahren
```

SYNCC 0

```
// In den CAM-Mode wechseln und bleiben
```

SYNCCSTART 0

```
// Walze sofort mit eingestellter max.
// Geschwindigkeit einkuppeln.
// Dies verursacht keine Bewegung, da
// Master steht und auf korrekter Position ist;
// jetzt kann der Master gestartet werden.
```

anf:

```
// leere Hauptschleife, damit Programm
// nicht beendet wird
// hier könnten weitere Verarbeitungen
// gemacht werden
```

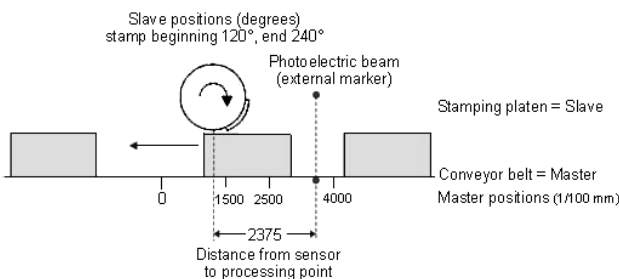
goto anf

■ Beispiel: Kartons bedrucken mit Markerkorrektur

Wenn die Kartons nicht in exakt gleichem Abstand transportiert werden, benötigen Sie Marker, mit denen ein Karton erkannt und die Synchronisation korrigiert werden kann.

Im Folgenden wird beschrieben, wie Sie die Kurve des vorgehenden Beispiels für diese Anwendung anpassen.

Wieder soll eine Walze auf Kartons eine 10 cm lange Aufschrift stempeln. Pro Minute werden maximal 60 Kartons auf dem Band transportiert. Während des Bedrucks müssen Stempelwalze und Karton synchron laufen.



■ Kurve für Synchronisation mit Marker editieren

- Schritte 1 bis 9 wie im vorhergehenden Beispiel.
- Definieren Sie in der Registerkarte ® "START-STOP-PUNKTE" die Punktpaare für das Ein- und Auskuppeln. Wir gehen davon aus, dass am Anfang des Kartons eingekuppelt und bis zum Ende des Kartons ausgekuppelt werden soll.

Fix Punkte		Start Stop Punkte	
Punkt	Master		
1 a	1000		
1 b	1500		
2 a	2500		
2 b	3000		
3 a			
3 b			

- Bestimmen Sie in der Registerkarte → "KURVEN-DATEN" die Position, in der die Walze stoppen soll, wenn im Programm keine andere Slave-Stop-Position definiert wird:
Die Walze soll immer auf Position 0 Grad zurückfahren: → "SLAVE-STOP-POSITION" = 0

- Die Lichtschranke (externer Marker) ist 237,5 mm vom Bearbeitungspunkt (= Stempel berührt den Karton) entfernt und erkennt den Anfang des Kartons (entspricht Master-Position 1000). Der Markerabstand beträgt demnach 2375. Tragen Sie diesen Wert in die Registerkarte → "SYNCHRONISATION" ein und definieren Sie die erlaubte Toleranz für das Auftreten der Marker und den externen Markertyp = 2 für den Master:

Markerabstand SYNCMPULSM (58) = 2375

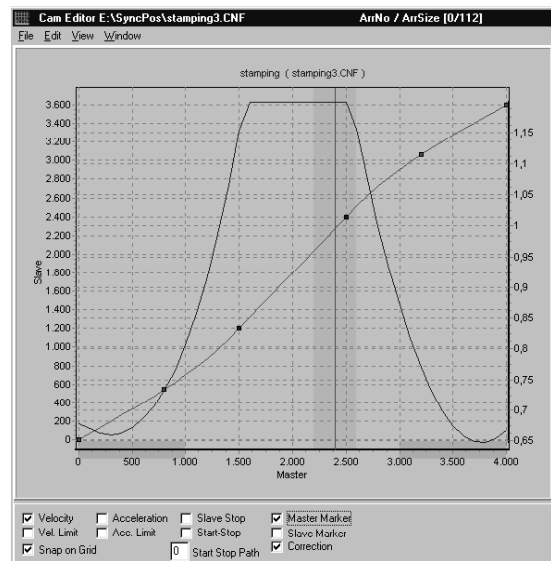
Toleranz SYNCMWINM (68) = 200

Markertyp SYNCTYPM (60) = 2

Tragen Sie die Master-Position in der Registerkarte → "KURVEN-DATEN" ein:

Master-Marker-Position = 1000

- Für die Festlegung, wann die Korrektur der Synchronisation frühestens beginnen kann und wann sie beendet sein muss, betrachten Sie das Kurvenprofil. Die grüne senkrechte Linie zeigt, an welcher Master-Position der Marker erkannt wird, der hellgrüne Bereich zeigt das Toleranzfenster für das Auftreten des Master-Markers. (Farbige Grafik siehe PC-Software)



Die Korrektur darf frühestens beginnen, wenn ein Karton fertig bedruckt ist, denn jede Änderung der Geschwindigkeit während des Bedrucks würde den Karton beschädigen. Und die Korrektur muss vollständig beendet sein, wenn der nächste Karton den Bearbeitungspunkt erreicht. In diesem Beispiel sind die Master-Positionen Ende und Anfang eines Kartons gut geeignet:

Korrektur Start = 3000

Korrektur Ende = 1000

Tragen Sie die Werte in die Registerkarte → "KURVEN-DATEN" ein; der Bereich wird im Kurvenprofil blau schraffiert gezeigt.

14. Prüfen Sie, ob die Geschwindigkeit und Beschleunigung des Slaves innerhalb des Limits bleiben. Aktivieren Sie dazu die Darstellung der → "GESCHWINDIGKEIT" und des → "GESCHW. LIMITS" und danach die Darstellung der → "BESCHLEUNIGUNG" und des → "BESCHL. LIMITS".

15. "DATEI" → "SPEICHERN ALS .. CNF"-Datei. In der Titelleiste sehen Sie den Namen der Kurve und die Anzahl der Array-Elemente. Letzteres benötigen Sie für die DIM-Anweisung bei der Programmierung.

■ Kurve in den VLT laden

Mit "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI" laden Sie die Kurve mit den veränderten Parametern und den – automatisch erzeugten – Kurvenarrays in den VLT.

■ Kurve in das Steuerungsprogramm einbinden

Da die Kurve intern als Array gespeichert wird, muss in Ihrem Programm als erstes die DIM-Anweisung stehen:

```
DIM marker[112] // Anzahl der Elemente aus
                // Titelleiste des CAM-Editors
```

HOME

```
// Slave Achse führt eine Homefahrt durch
// (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der
// Nullposition (0 Grad)
// (entfällt bei Absolutgeber)
```

SETCURVE marker

```
// Stempelkurve mit Marker setzen
```

dist = GET SYNCMPULSM

```
// Abstand zum Sensor
```

DEFMCPOS (1000-dist)

```
// Das ist die Stelle, die dem Sensorsignal
// entspricht
```

SET SYNCMSTART 2000

```
// Zählen des Masterpulses beginnt erst
// wenn nächste Flanke vom Sensor kommt
```

SYNCCMM 0

```
// Im CAM-Mode synchronisieren bis
// Motor Stopp
```

SYNCCSTART 1

```
// Walze mit Start-Punktepaar 1 einkuppeln
```

```
// Synchronbetrieb
```

WAITI 4 ON

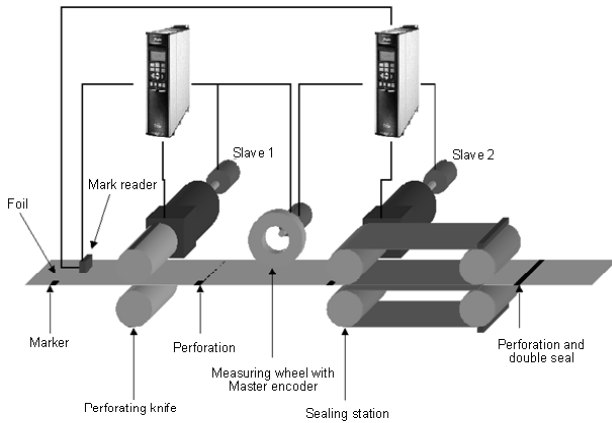
```
// Warten auf Eingangssignal, wenn
// Transportband abgeschaltet wird
```

SYNCCSTOP 2 0

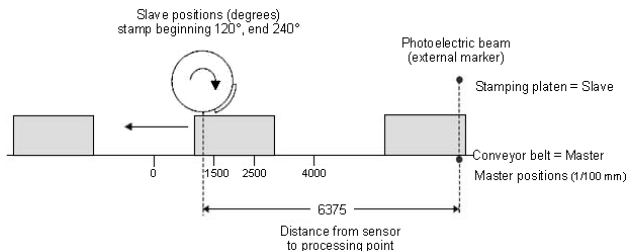
```
// Walze mit Stopp-Punktepaar 1 auskuppeln
// und bei Position 0 Grad anhalten.
```

■ Wenn der Abstand des Sensors größer als eine Masterzykluslänge ist

Bei vielen Anwendungen kann der Marker nicht innerhalb einer Masterzykluslänge angebracht werden, z.B. bei folgender Maschine zur Produktion von Plastiktüten:



Da hier zwischen den Slaves keine Marker eingebaut werden können, gibt es in dieser Anwendung nur einen Markenleser, die Schweißstation liegt aber viel weiter als eine Masterzykluslänge entfernt. Da der Abstand des Sensors größer als eine Masterzykluslänge ist, wird ein Puffer für die Markerabweichung angelegt. Bei Erscheinen des Markers wird der Wert in den Puffer geschrieben und mit Erscheinen des nächsten Markers ausgelesen:

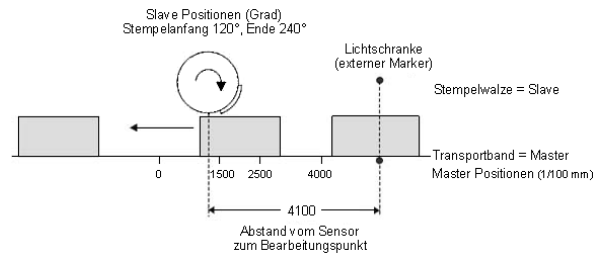


Um zu beurteilen, in welchem Bereich korrigiert werden darf, subtrahieren Sie so oft die Masterzykluslänge, bis der Wert < 1 Masterzykluslänge ist. Dies ist der maximal erlaubte Abstand zum Korrigieren. In diesem Beispiel ist dieser also $6375 - 4000 = 2375$ und damit der gleiche Korrekturbereich wie im vorangegangenen Beispiel.

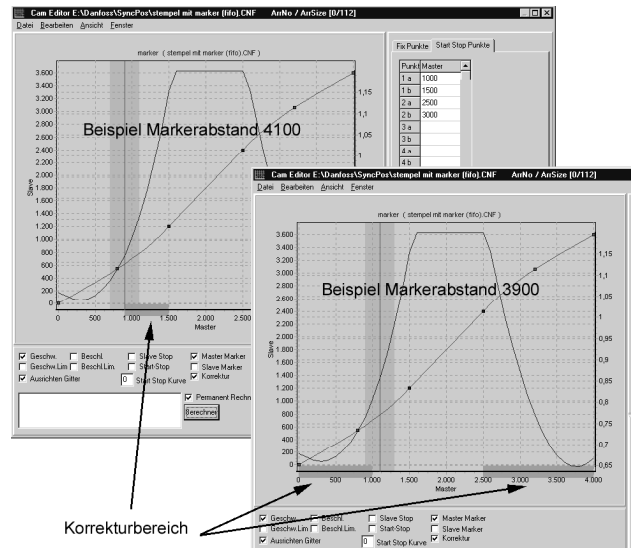
■ Problemfälle bei der Festlegung des Markerabstandes

Wenn der Marker so nah am Bearbeitungspunkt angebracht ist, dass nach Erkennen des Markers keine Zeit bleibt, die Synchronisation zu korrigieren, können Sie das Problem nur durch mechanische Veränderung des Markers beheben.

Der gleiche Effekt könnte aber auch auftreten, wenn der Markerabstand größer als die Masterzykluslänge ist und nach Subtraktion dieses Wertes ebenfalls ein zu geringer Abstand bleibt, zum Beispiel:



Bei Erscheinen des Markers wird der Wert in den Puffer geschrieben. Erst wenn der nächste Marker erkannt wird, wird der Puffer ausgelesen. Das bedeutet, dass der Marker erst bei der Master-Position 900 „erkannt“ wird und in unserem Beispiel nur noch wenig Zeit bleibt, den Fehler zu korrigieren. Es ist der gleiche Effekt, als wäre der Sensor um den Wert (Abstand - Mastertaktlänge) bzw. $(4100 - 4000)$, also nur 10 mm vor dem Bearbeitungspunkt montiert.



Daher wäre es besser, den Sensor so zu montieren, dass der Abstand zum Bearbeitungspunkt entweder kleiner oder wesentlich größer als eine Masterzykluslänge ist, hier zum Beispiel im Abstand von 3900. Dann kann man von 2500 bis 1000 korrigieren.

Oder man montiert den Sensor weiter weg, zum Beispiel im Abstand von 7900, dies wirkt genau so, als wäre der Sensor um Abstand – Masterzykluslänge (7900 – 4000), also 3900 vor dem Bearbeitungspunkt montiert. Genügend Zeit also, um die Synchronisation zu korrigieren.

Falls dies mechanisch nicht möglich ist, muss man die Werte etwas manipulieren, damit man die Lösung mit dem Puffer vermeiden kann. Gehen Sie folgendermaßen vor:

Subtrahieren Sie vom tatsächlichen Abstand einen Wert x, damit der Abstand < Masterzykluslänge wird, zum Beispiel 4100 – 200 = 3900. Den Wert x subtrahieren Sie auch von der Master-Position, also 1000 – 200 = 800.

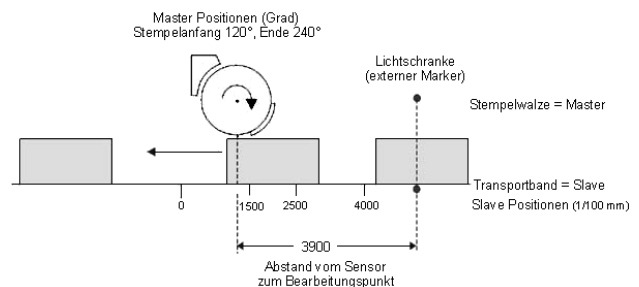
Tragen Sie beide Werte in die Registerkarten → "SYNCHRONIZATION" and → "CURVE DATA" ein:

Master Marker Abstand
 SYNCMPULSM (58) = 3900
 Master Marker Position = 800

Da nun kein Puffer erzeugt wird, könnte man zum Beispiel von 2500 bis 800 korrigieren.

■ Beispiel: Slave-Synchronisation mit Marker

In folgendem Beispiel ist das Transportband der Slave und die Stempelwalze der Master, da für einen gleichmäßig Druck, die Farbaufnahme und Farbabgabe kontinuierlich ablaufen müssen. Pro Minute werden maximal 20 Kartons auf dem Band transportiert. Der Abstand der Kartons ist nicht größer als eine Masterzykluslänge. Während des Bedruckens müssen Stempelwalze und Karton synchron laufen:



Im Gegensatz zur Synchronisation mit Master-Markerkorrektur wird hier die Slave-Position korrigiert und nicht die Kurve.

■ Kurve für Slave-Synchronisation editieren

1. VLT mit den erforderlichen Parameter einstellen und diese Benutzerparameter mit "PARAMETER" → "SPEICHERN IN DATEI" mit der Extension „cnf“ sichern.
2. Starten Sie den → "CAM-EDITOR" und laden Sie diese cnf-Datei mit "DATEI" → "LADEN CNF".
3. Ermitteln Sie den Getriebefaktor des Masters in MU-Einheiten:
 Getriebefaktor = 5/1
 Drehgeber-Auflösung (Inkremental) = 500
 Ein Umdrehung der Walze ist 360 Grad. Wir wollen mit einer Auflösung von 1/10 Grad arbeiten und teilen daher eine Umdrehung der Walze in 3600 Arbeitseinheiten ein:
 Skalierfaktor = 3600

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ MU}$$
 Tragen Sie die Werte in der Registerkarte → "SYNCHRONISATION" ein:
 Syncfaktor Master [49] = 25
 Syncfaktor Slave [50] = 9

4. Getriebefaktor des Slaves in BE-Einheiten eingeben:
 Der Antrieb ist mit dem Transportband mit einer Getriebeübersetzung von 25:11 verbunden; das heißt der Motor macht 25, das Zahnriemenrad 11 Umdrehungen.
 Getriebefaktor = 25/11
 Inkrementaldrehgeber direkt am Master-Antrieb;
 Encoder-Auflösung = 4096
 Das Zahnriemenrad hat 20 Zähne/Umdrehung, 2 Zähne entsprechen 10 mm, daher entspricht 1 Umdrehung = 100 mm Transportbandvorschub. Die Eingabe soll in 1/10 mm Auflösung möglich sein.
 Skalierfaktor ist demnach 1000

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ BE}$$

Tragen Sie die Werte in der Registerkarte ® "ENCODER" ein:

Benutzerfaktor Zähler [23] = 2048
 Benutzerfaktor Nenner [26] = 55

5. Damit die Fixpunkte auf den Interpolationspunkten liegen, bestimmen Sie in der Registerkarte → "KURVEN-DATEN" einen ganzzahligen Teiler für die Intervalle.
 Eine komplette Zykluslänge des Masters ist 360 Grad; dies entspricht 3600 MU.
 Die → "ANZAHL INTERVALLE" = 36 ergibt eine vernünftige Intervallzeit von 27,7 ms.

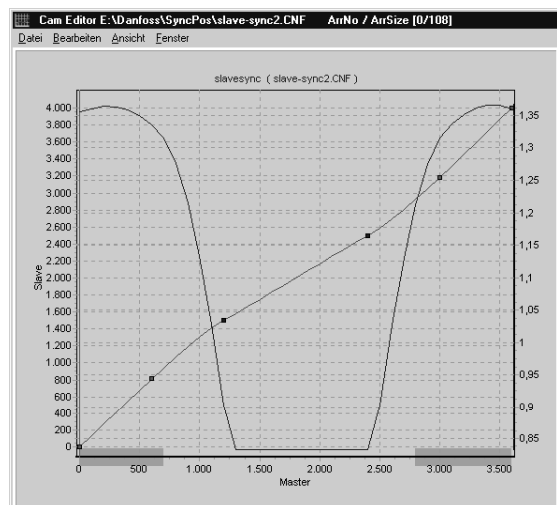
6. Definieren Sie → "FIXPUNKTE" für die Walze (Slave) und das Transportband (Master) und den Punkttyp „1“ für Kurvenpunkte. Die Funktion → "AUSRICHTEN AN GITTER" sollte aktiviert sein.

Punkte	Master	Slave	Typ
1	0	0	1
2	1200	1500	1
3	2400	2500	1
4	3600	4000	1

7. Zwischen der Master-Position 1200 bis 2400 müssen Master und Slave synchron mit gleicher Geschwindigkeit fahren. Dafür benötigen Sie eine Gerade, die mit zwei Tangentenpunkten bestimmt wird. Definieren Sie also für diese beiden Fixpunkte den → "TYP 2" für Tangentenpunkt und fügen Sie noch zwei Fixpunkte ein, damit die Kurve berechnet werden kann.

Punkte	Master	Slave	Typ
1	0	0	1
2	600	813	1
3	1200	1500	2
4	2400	2500	2
5	3000	3186	1
6	3600	4000	1

Aktivieren Sie die grafische Darstellung der → "GESCHWINDIGKEIT" und verschieben Sie die beiden neu eingefügten Fixpunkte so lange mit der Maus, bis Sie einen gleichmäßigen Geschwindigkeitsverlauf erhalten:



8. Tragen Sie in der Registerkarte → "KURVEN-INFO" die → "ZYKLEN / MIN MASTER" = 20 ein. Das ist die Anzahl der Kartons, die (maximal) pro Minute bearbeitet werden.
9. Prüfen Sie, ob die Beschleunigung des Slaves innerhalb des Limits liegt. Aktivieren Sie dazu die Darstellung der → "BESCHLEUNIGUNG" und des → "BESCHL. LIMITS".
10. Definieren Sie in der Registerkarte → "START-STOP-PUNKTE" um die Synchronisation am Anfang zu starten. Zwischen 20 und 100 Grad soll mit etwas Sicherheitsabstand eingekuppelt werden, denn bei 120 Grad muss auf-synchronisiert sein.

Punkte	Master
1a	200
1b	1000

11. Bestimmen Sie in der Registerkarte → "KURVEN-DATEN" die Position, bei der das Transportband stoppen soll, wenn im Programm keine andere Slave-Stop-Position definiert wird: Das Transportband soll immer auf Position 0 halten: → "SLAVE-STOP-POSITION" = 0

12. Die Lichtschranke (externer Marker) ist 390 mm vom Bearbeitungspunkt (= Stempel berührt den Karton) entfernt und erkennt den Anfang des Kartons (entspricht Slave-Position 1000). Der Markerabstand beträgt demnach 3900. Tragen Sie diesen Wert in die Registerkarte → "SYNCHRONISATION" ein und definieren Sie die erlaubte Toleranz für das Auftreten der Marker und den externen Markertyp = 2 für den Slave:
 - Markerabstand SYNCMPULSM (59) = 3900
 - Toleranz SYNCMWINM (69) = 200
 - Markertyp SYNCTYPM (61) = 2
 Tragen Sie die Slave-Position in der Registerkarte → "KURVEN-DATEN" ein:
 - Master-Marker-Position = 1000

13. Für die Festlegung, wann die Korrektur der Synchronisation frühestens beginnen kann und wann sie beendet sein muss, betrachten Sie das Kurvenprofil. Die grüne waagrechte Linie zeigt, an welcher Master-Position der Marker erkannt wird, der hellgrüne Bereich zeigt das Toleranzfenster für das Auftreten des Master-Markers.

Die Korrektur darf frühestens beginnen, wenn ein Karton fertig bedruckt ist, denn jede Änderung der Geschwindigkeit während des Bedruckens würde den Druckstempel und/oder den Karton beschädigen. Und die Korrektur muss vollständig beendet sein, wenn der nächste Karton den Bearbeitungspunkt erreicht. In diesem Beispiel sind die Slave-Positionen Ende und Anfang eines Kartons gut geeignet. Tragen Sie die Werte in die Registerkarte → "KURVEN-DATEN" ein:

 - Korrektur Start = 2800
 - Korrektur Ende = 750

14. Prüfen Sie, ob die Geschwindigkeit und Beschleunigung des Slaves innerhalb des Limits bleiben. Aktivieren Sie dazu die Darstellung der "GESCHWINDIGKEIT" und des → "GESCHW. LIMITS" und danach die Darstellung der → "BESCHLEUNIGUNG" und des → "BESCHL. LIMITS".

■ Kurve in den VLT laden

Mit "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI" laden Sie die Kurve mit den veränderten Parametern und den – automatisch erzeugten – Kurvenarrays in den VLT.

■ Kurve in das Steuerungsprogramm einbinden

Um die Master-Position zu bestimmen, wird ein Schalter am Master vorausgesetzt, der die Nullposition signalisiert. Um den Slave in die richtige Position zu bringen, wird dieser bis zur Lichtschranke vorwärts gefahren. Dies entspricht dem Kartonanfang = 1000. Dann fährt man den Slave um 2900 (= Markerabstand 3900–1000) weiter; damit steht der Slave mit dem Kartonanfang 1000 genau vor dem Bearbeitungspunkt, also an Slave-Position 0.

```
DIM slavesync[108] // Anzahl der Elemente
                    // siehe CAM-Editor
```

```
HOME
```

```
// Slave führt eine Homefahrt durch
// (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der
// Nullposition (0 Grad)
// (entfällt bei einem Absolutdrehgeber)
```

```
DEFMCPOS 0
```

```
// Kurve beginnt bei Master-Position 0
```

```
SET SYNCMSTART 2000
```

```
// Zählen des Masterpulses beginnt erst
// wenn nächste Flanke von Sensor kommt
```

```
SETCURVE slavesync
```

```
// Kurve für die Slave-Synchronisation setzen
// zum Start fahren
```

```
CSTART
```

```
CVEL 10 // langsam vorwärts fahren
        // bis Lichtschranke kommt
```

```
oldi = IPOS
```

```
// oldi = letzte Markerposition des Slaves
```

```
WHILE (oldi == IPOS) DO
```

```
// Warten bis Karton erkannt
```

```
ENDWHILE
```

```
POSA (IPOS + 2900)
```

```
// Karton um 2900 nach vorne fahren
```

```
SYNCCMS 0 // Im CAM-Mode synchronisieren
```

```
SYNCCSTART 1
```

```
// Mit Start-Stop-Punktepaar 1 einkuppeln
```

15. "DATEI" → "SPEICHERN ALS .. CNF".

■ Nockenschaltwerk
■ So funktioniert ein Nockenschaltwerk mit SyncPos

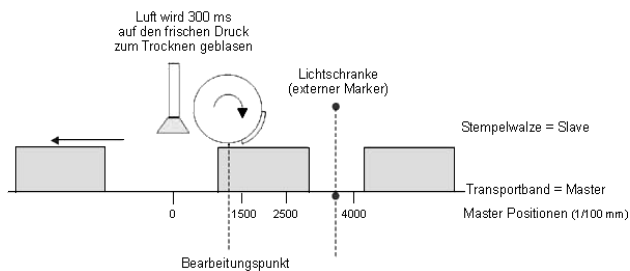
Die mechanische Nockenwelle wird ebenfalls durch eine (oder mehrere) Kurven nachgebildet. Um ein Nockenschaltwerk zu realisieren, muss es möglich sein, den Slave immer wieder an bestimmten Master-Positionen ein- und auszukuppeln.

Dies ist mit SyncPos mit den Interruptbefehlen ON MAPOS .. GOSUB und ON APOS .. GOSUB möglich. Man kann immer dann ein Unterprogramm aufrufen, wenn eine definierte Master-Position (und zwar in positiver oder negativer Richtung) passiert wurde.

In Verbindung mit einem Kurvenprofil, in dem mehrere Start-Stop-Punktepaare zum Aus- und Einkuppeln definiert wurden, kann man viele Anwendungen wie sie in der Verpackungsindustrie typisch sind realisieren.

■ Beispiel eines Nockenschaltwerks

Nach dem Bedrucken eines Kartons soll der frische Druck sofort im Luftstrom getrocknet werden:



```
ON MCPOS 2500 GOSUB trocknen
```

```
    // Unterprogramm aufrufen, wenn die
    // Master-Position 2500 in positiver Richtung
    // passiert wurde
```

```
SUBMAINPROG
```

```
  SUBPROG trocknen
```

```
    OUT 1 1    // Trockner einschalten
```

```
    DELAY 300 // 300 ms trocknen
```

```
    OUT 1 0    // Trockner ausschalten
```

```
  RETURN
```

```
ENDPROG
```

Kapitel 5
■ PC-Software Benutzeroberfläche
■ Menü "DATEI"

"NEU", "ÖFFNEN", "SCHLIESSEN"	55
"SPEICHERN" und "SPEICHERN ALS"	56
"DRUCKEN" und Druckereinstellungen	56
"PROGRAMMENDE"	56
Datei löschen	56

■ Menü "BEARBEITEN"

"RÜCKGÄNGIG"	57
"AUSSCHNEIDEN"	57
"KOPIEREN" und "EINFÜGEN"	57
"SUCHEN" und "ERSETZEN"	58

■ Menü "ENTWICKLUNG"

"AUSFÜHREN"	58
"ABBRECHEN"	59
Programm "FORTSETZEN"	59
Debug Modus	59
"VORBEREITEN EINZELSCHRITT"	59
Haltepunkte setzen	60
Variablen online ändern oder auslesen	60
"START" Debug und "EINZELSCHRITT"	60
"BEENDEN DEBUG"	60
"ÜBERWACHUNG ANZEIGEN"	60
"SYNTAXPRÜFUNG"	61
"IN DATEI COMPILIEREN"	61
"ABBRUCH ALLE"	61
"STEUERUNG AUSWÄHLEN"	61
"SCHNITTSTELLE SCHLIESSEN"	61
"BEFEHLSHILFE"	61
"POSITION ANFAHREN" (Teach-in)	62

■ Menü "STEUERUNG"

"PROGRAMME"	62
Sichern und Rücklesen Quellcode	63
"AUTOSTART"	63
"STEUERUNG" → "PARAMETER"	65
Globale Parameter	64
Achsparemeter	65
Achsparemeter online einstellen oder ändern	65
Achsparemeter einer Konfigurationsdatei ändern ...	66
"PARAMETER" → "NAME"	67
"PARAMETER" → "SPEICHERN IN DATEI"	67
"PARAMETER" → "WIEDERHERSTELLEN AUS DATEI"	67
"SPEICHER" → "RAM SPEICHERN"	68
"SPEICHER" → "EEPROM LÖSCHEN"	68
"RESET" →	68

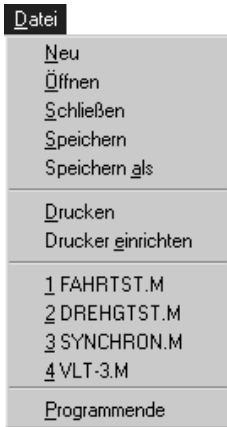
■ Menü "TESTFAHRT"	
Testfahrt-Parameter festlegen	69
"TESTFAHRT" → "AUSFÜHREN"	70
"FEHLER LÖSCHEN"	70
"AUFZEICHNUNG ANZEIGEN"	71
Fahrdiagramme auswerten	71
■ Menü "CAM-EDITOR"	
CAM-Editor-Fenster	73
Menü "DATEI" CNF	74
Menü Kurve "BEARBEITEN"	75
Menü "ANSICHT"	76
Menü "FENSTER" (CAM-Editor)	76
Kurvenprofil	76
Meldungen des CAM-Editors	77
Registerkarte "FIXPUNKTE"	77
Registerkarte "START STOP PUNKTE"	78
Registerkarten "KURVEN-DATEN", "KURVEN-INFO" und Parameter	79
Registerkarte "KURVEN-DATEN"	79
Registerkarte "KURVEN-INFO"	80
Registerkarten Encoder, Homefahrt, Ein-/Ausgänge und Regelung	81
Registerkarte Synchronisation	81
Registerkarte Geschwindigkeit	82
■ Menü "EINSTELLUNGEN"	
"COMPILER"	82
"SCHNITTSTELLE"	83
"FARBEN EDITOR"	83
"SPRACHE"	83
■ Menü "FENSTER"	
"KASKADIEREN"	84
"VERTIKAL" (Nebeneinander)	84
"HORIZONTAL" (Untereinander)	84
■ Menü "HILFE"	
"INHALT"	85
"PROGRAMMINFO"	85

■ PC Software Benutzeroberfläche

Das folgende Kapitel beschreibt die einzelnen Menüs der SyncPos Benutzeroberfläche. Wenn Sie mit der Windows-Oberfläche vertraut sind, können Sie natürlich die bekannten Menüs überspringen und sich sofort über die SyncPos-spezifischen informieren.

Die grundsätzlichen Erläuterungen über Programmaufbau, Befehlsaufbau, Interrupt, Sprachelemente, Arithmetik und Benutzereinheit finden Sie nach der Einleitung zur Software-Referenz. Alle Befehle und Parameter im Detail und alphabetisch geordnet können Sie in den Kapiteln [Software-Referenz](#) und [Parameter-Referenz](#) nachschlagen.

■ Menü "DATEI"



Das Menü "DATEI" enthält alle Befehle zum Erstellen, Öffnen, Speichern und Drucken eines Programmes. Alle Befehle erreichen Sie per Mausklick oder mit der Tastenkombination [ALT] und dem unterstrichenen Buchstaben, zum Beispiel [ALT] + [D] + [S], wenn Sie die "DATEI" → "SPEICHERN" wollen.

■ "NEU"

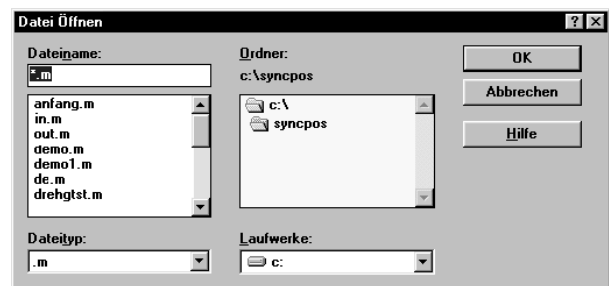
Klicken Sie in der Menüleiste auf "DATEI" und dann auf "NEU" oder klicken Sie auf die Schaltfläche , um ein neues Programm zu schreiben. Das Editierfenster wird mit der Bezeichnung „Unnamed Programm1“ geöffnet und Sie können beginnen, Ihr Programm zu schreiben.

■ "ÖFFNEN"

Klicken Sie auf "DATEI" → "ÖFFNEN" oder auf das Schaltsymbol um eine Programmdatei zu öffnen. Wählen Sie im folgenden Dialogfeld die gewünschte Datei mit Doppelklick oder mit Klicken auf "OK" aus. Wenn sich die gewünschte Datei noch in der Liste der zuletzt geöffneten Dateien (max. 9 Einträge) befindet, können Sie diese anklicken und öffnen.

Dateityp

Da alle SyncPos-Programmdateien die Extension ".m" verlangen, muss dieser Dateityp auch ausgewählt sein. Falls nicht, wählen Sie im Feld **Dateityp** ".m" oder **Alle Dateien (*.*)** aus.



Falls Sie die Datei nicht im linken Feld finden, klicken Sie im Feld **Ordner** auf ein anderes Verzeichnis oder klicken im Feld **Laufwerke** auf das Laufwerk, das die Datei enthält. Doppelklicken Sie so lange auf die Unterverzeichnisse, bis Sie die gesuchte Datei im linken Feld markieren können. Klicken Sie auf "OK".

Datei suchen

Wenn Sie eine Datei im System „verloren“ haben, können Sie zum Beispiel mit dem **Explorer** alle *.m suchen.

■ "SCHLIESSEN"

"DATEI" → "SCHLIESSEN"

Klicken Sie auf "DATEI" → "SCHLIESSEN", wenn Sie die geladene oder geschriebene Programmdatei schließen wollen, aber noch nicht das Programm SyncPos beenden wollen.




ACHTUNG!

Falls Sie die neu geschriebene Datei oder die Änderungen noch nicht gespeichert haben, beantworten Sie die nun folgende Abfrage auf jeden Fall mit "JA", andernfalls wird die neue Datei nicht gespeichert.

■ "SPEICHERN" und "SPEICHERN ALS"

"DATEI" → "SPEICHERN"

Klicken Sie auf "DATEI" → "SPEICHERN ALS" oder auf das Schaltsymbol  um ein neues oder geändertes Programm zwischendurch bzw. spätestens vor dem Schließen auf der Festplatte des PCs zu speichern. Falls Sie eine neue Programmdatei noch nicht gespeichert haben, wird automatisch das entsprechende Fenster "DATEI" → "SPEICHERN ALS" ... geöffnet.

Klicken Sie auf "DATEI" → "SPEICHERN ALS", um ein neues Programm oder ein geändertes Programm unter einem neuen Namen zu speichern. Alle Programmdateien werden mit der Extension .m versehen.



ACHTUNG!

Sie sollten Ihre Dateien immer im PC speichern. Denn obwohl das Programm bei jedem Ausführen und bei jeder Syntaxprüfung automatisch auch in der SyncPos-Option gesichert wird, ist es dann schon kompiliert und daher im PC nicht mehr zu verändern.

Für eine Kompatibilität zwischen Windows '95 und früheren Windows-Versionen wurde auf die Eingabe von mehr als 8 Zeichen für den Dateinamen verzichtet.

Speicherplatz

Sie können so viele Programmdateien sichern, wie der freie Speicherplatz auf der Festplatte des PCs zulässt.

■ "DRUCKEN" und Druckereinstellungen

"DATEI" → "DRUCKEN"

Zum Ausdrucken der Datei klicken Sie auf "DATEI" → "DRUCKEN", stellen den Druckbereich und die gewünschte Anzahl der Kopien ein und starten den Drucker mit "OK".

Je nach Betriebssystem – Windows 95 oder Windows NT – erscheint ein etwas anderes Dialogfeld. Informieren Sie sich in der Windows-Hilfe zur Systemsteuerung.


Klicken Sie auf "EINRICHTEN...", wenn Sie noch Ausgabedetails wie Papierformat oder Ausrichtung (Hoch- oder Querformat) bestimmen wollen. Diese Einstellungen erreichen Sie auch, wenn Sie auf "DRUCKEREINSTELLUNGEN" klicken.

Drucker einstellen

Klicken Sie auf → "DRUCKEREINSTELLUNGEN" oder im vorigen Dialog auf "EINRICHTEN...", wenn Sie das Papierformat, die Ausgabegröße (Skalierung) oder andere Optionen einrichten wollen. Über die Möglichkeiten des Fensters "DRUCKER EINRICHTEN" informieren Sie sich bitte in der Windows-Hilfe.

■ "PROGRAMMENDE"

SyncPos beenden

Mit Klicken auf "PROGRAMMENDE" oder auf das Symbol  wird das Programm SyncPos beendet. Falls Sie eine neue Datei oder eine Änderung noch nicht gespeichert haben, erhalten Sie mit einer letzten Abfrage die Chance, dies zu tun.



ACHTUNG!

"PROGRAMMENDE" beendet aber nicht ein laufendes Programm im VLT. Ein laufendes Programm können Sie nur mit [ESC] abbrechen oder beenden. Dazu muss auch die Datei, die mit der Steuerung bzw. dem VLT verbunden ist, geöffnet sein bzw. wieder geöffnet werden.




ACHTUNG!

Wenn die Steuerung bei "DATEI" → "PROGRAMMENDE" dennoch stehen bleibt, kann es daran liegen, dass vom VLT PRINT-Befehle geschickt werden, die nun nicht mehr im Kommunikationsfenster dargestellt werden können.

■ Datei löschen

Eine oder mehrere Dateien löschen Sie – wie unter Windows gewohnt – im **Explorer**.

■ Menü "BEARBEITEN"


Das Menü "BEARBEITEN" bietet die zum Programmieren notwendigen Editierhilfen, von denen Sie die meisten auch – wie in Windows gewohnt – über Tasten und Tastenkombinationen erreichen können. Schreiben Sie Ihr Programm und nutzen Sie den Tabulator  und die verschiedenen Farben, um das Programm optisch zu strukturieren. Die Tab-Schritte sind fest eingebaut.

Text markieren

Zeigen Sie auf die Stelle, an der die Markierung beginnen soll und ziehen Sie den Mauscursor über den Text. Soll nur ein einzelnes Wort markiert werden, doppelklicken Sie auf das betreffende Wort. Sie können eine Markierung mit den Shortcuts in alle Richtungen erweitern; zum Beispiel mit [UMSCHALT] + [→]-Taste um ein Zeichen nach rechts, mit [STRG] + [UMSCHALT] + [→-TASTE] bis zum Wortende, mit [UMSCHALT] + [ENDE] bis zum Zeilenende usw. Informieren Sie sich in der [Liste der Shortcuts](#) über alle Möglichkeiten.

Text löschen

Löschen Sie Einzelbuchstaben entweder mit der [ENTF]-Taste (sie löscht einzeln nach der Einfügemarke nach rechts) oder mit der [RÜCK-Taste] (sie löscht einzeln vor der Einfügemarke nach links). Wenn Sie markierte Wörter oder Zeilen löschen, wirken beide Tasten gleich.

Zeilennummer

Innerhalb Ihres Programmes können Sie sich an den Zeilennummern orientieren. Die Syntaxprüfung zum Beispiel stellt nicht nur den Cursor in die entsprechende Zeile, sondern nennt auch die Zeilennummer mit dem falschen Befehl.

Die aktuelle Zeilennummer finden Sie in der Statuszeile, zum Beispiel 13:1. Der Cursor steht dann in der Zeile 13 auf Schreibposition 1.

Kontextmenü


Klicken Sie auf die rechte Maustaste und das Menü "BEARBEITEN" steht dann als Kontextmenü zur Verfügung.

Kontextmenüs erscheinen im Fenster an der Stelle, an der Sie gerade arbeiten. Sie schließen das Kontextmenü, indem Sie außerhalb des Kontextmenüs klicken, ohne dabei einen Befehl zu wählen. Oder Sie tasten [ESC].

■ "RÜCKGÄNGIG"



Wenn Sie die letzte Eingabe rückgängig machen wollen, klicken Sie auf "BEARBEITEN" → "RÜCKGÄNGIG". Sie können mit jedem weiteren Klicken alle Eingaben bis zur letzten Speicherung rückgängig machen. Mit [STRG] + [Z] erreichen Sie das Gleiche.

■ "AUSSCHNEIDEN"

Mit "AUSSCHNEIDEN" können Sie Text kopieren und löschen. Markieren Sie die gewünschte Textstelle und klicken Sie auf "AUSSCHNEIDEN" oder auf das Schaltsymbol . Der Text wird für das Einfügen zwischengespeichert. Mit der nächsten Kopieraktion wird der Zwischenspeicher ohne Rückfrage überschrieben.

Die Tastenkombination [STRG] + [ENTF] bewirkt die gleiche Funktion.

■ "KOPIEREN" und "EINFÜGEN"

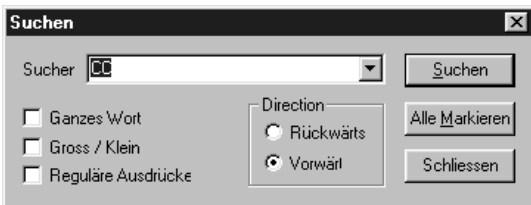
Markieren Sie die gewünschte Textstelle und klicken Sie auf "KOPIEREN" oder auf das Schaltsymbol , die Textstelle wird in den Zwischenspeicher kopiert. Stellen Sie den Cursor an die gewünschte Einfügestelle und klicken Sie auf "EINFÜGEN" oder auf das Schaltsymbol .

Die Tastenkombination [STRG] + [EINFG] erfüllt den gleichen Zweck.

■ "SUCHEN" und "ERSETZEN"

Suchen und Ersetzen ist gemäß den Windows-Konventionen realisiert und mit einigen nützlichen Funktionen ergänzt.

Klicken Sie auf "BEARBEITEN" → "SUCHEN" oder [STRG] + [F] und geben Sie im folgenden Dialogfeld den gesuchten Begriff ein. Mit [F3] können Sie dann von einer Fundstelle zur nächsten springen.



Klicken Sie auf → "ALLE MARKIEREN" statt auf → "SUCHEN" und es werden sofort alle Fundstellen am linken Rand mit einem blauen Dreieck markiert. Sie können dann mit [F2] zwischen den Fundstellen hin- und herspringen.

Reguläre Ausdrücke

Diese Funktion ist in Suchen und Ersetzen mit folgenden Syntax-Regeln realisiert:

- Wildcards: ? (für beliebiges Zeichen), + (für ein oder mehrere Suchbegriffe), * (für kein oder mehrere Zeichen).
- Zeichengruppe: Zeichen in eckigen Klammern werden als Gruppe gesucht; der Bereich wird mit Bindestrich angegeben, z.B. alle Zeichen von a bis c: [a-c].
- Logisches ODER: Unterausdrücke werden mit Hilfe des Pipeline-Symbols | mit ODER verknüpft.
- Unterausdrücke in Anführungszeichen: Ein regulärer Ausdruck sollte in Anführungszeichen gesetzt werden und wird als eine Einheit behandelt.
- Code-Umschaltzeichen: Abläufe wie \t, etc. werden durch ein äquivalentes einzelnes Zeichen ersetzt. \\ stellt den Backslash dar.

Ersetzen

Benutzen Sie Bearbeiten → "ERSETZEN" oder [STRG] + [R] wenn Sie die gefundene(n) Passage(n) ersetzen wollen.



■ Menü "ENTWICKLUNG"

Entwicklung	
Ausführen	F5
Abbrechen	ESC
Fortsetzen	
Vorbereiten Einzelschritt	
Start	
Einzelschritt	F9
Beenden Debug	
Überwachung anzeigen	
Syntaxprüfung	
In Datei kompilieren	
Abbruch alle	
Steuerung auswählen	
Schnittstelle schliessen	
Befehlshilfe	F12

Mit den Funktionen des Menüs "ENTWICKLUNG" können Sie die Programme ausführen, abbrechen, fortsetzen oder – besonders nützlich bei der Fehlersuche – schrittweise ausführen. Ein Debug-Mode und online Statusinformationen sowie die Möglichkeit, während der Programmausführung die Variablen zu ändern, erleichtern das Programmieren.

Neu geschriebene Programme sollten Sie vor dem ersten Ausführen einer "SYNTAXPRÜFUNG" unterziehen.

Bevor Sie jedoch beginnen, müssen Sie immer eine Steuerung bzw. einen VLT auswählen.

In der "BEFEHLSHILFE" verbergen sich mehrere hilfreiche Funktionen: Hier sind übersichtlich alle SyncPos-Befehle aufgelistet, die auch sofort ausgewählt und in das Editierfenster übernommen werden können. Und Sie können hier auch mit der Teach-in-Programmierung arbeiten.

■ "AUSFÜHREN"

Klicken Sie auf "ENTWICKLUNG" → "AUSFÜHREN" oder tasten Sie [F5] und es wird das Programm gestartet, das geöffnet und im Editor dargestellt ist. Dazu wird das Programm in den VLT geladen und zuvor kompiliert, also in die systeminterne Sprache übersetzt. Durch die Kompilierung reduziert sich die Programmgröße erheblich und der Speicher des Motion Controllers wird so sehr effizient genutzt. Gleichzeitig wird das Programm in den temporären Bereich des RAMs geladen, der mit jedem weiteren Ausführen überschrieben wird. Beim Programmieren haben Sie so einen schnellen unkomplizierten Arbeitsablauf zum Testen.

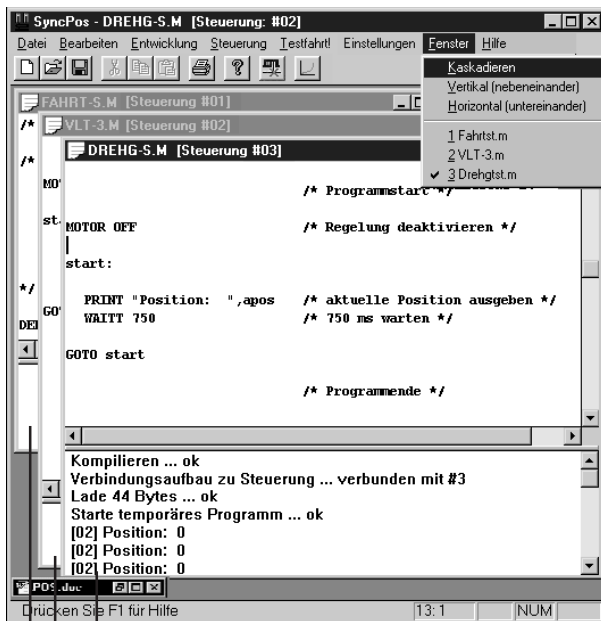


ACHTUNG!

Allerdings ist es nicht möglich, eine kompiliertes Programm wieder zurück in den PC zu holen, bzw. die Quelldatei wieder im PC zu bearbeiten. Daher sollten Sie alle Programme grundsätzlich auch auf der Festplatte des PCs speichern.

Programme in mehreren VLTs ausführen

Wenn Sie das Programm in mehrere VLTs laden wollen, verbinden Sie das Programm mit dem jeweiligen VLT und klicken auf → "AUSFÜHREN". Wenn Sie in jeden VLT ein anderes Programm laden wollen, öffnen Sie für jeden VLT ein eigenes Editierfenster, öffnen dort die gewünschte Programmdatei und verbinden es mit → "STEUERUNG AUSWÄHLEN" mit dem VLT. Dann starten Sie nacheinander jedes Programm mit → "AUSFÜHREN".



Steuerung, bzw. VLT 3
VLT 2
VLT 1

■ "ABBRECHEN"

Klicken Sie auf "ENTWICKLUNG" → "ABBRECHEN" oder tasten Sie [ESC] um das Programm sofort abzurechnen. Dabei werden auch eventuell aktive Fahrprozesse vorzeitig beendet.



ACHTUNG!

Es wird mit der maximal zulässigen Verzögerung abgebremst.

■ Programm "FORTSETZEN"

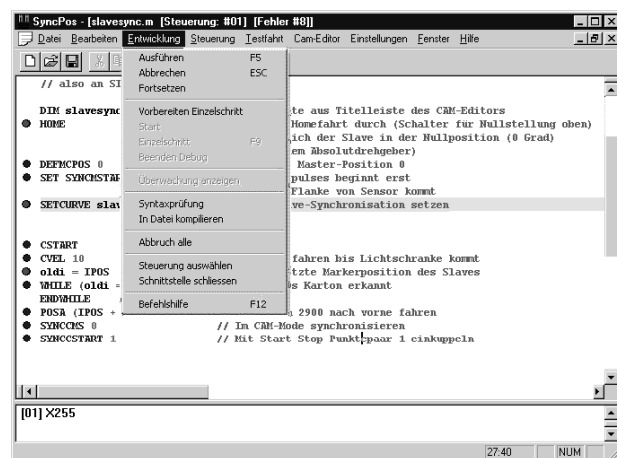
Klicken Sie auf "ENTWICKLUNG" → "FORTSETZEN", um das eben abgebrochene Programm fortzusetzen. Dabei werden auch die unterbrochenen Fahrprozesse zu Ende ausgeführt. Wenn ein Programm mit einer Fehlermeldung abgebrochen wurde, können Sie es – nachdem Sie den Fehler behoben und/oder die Fehlermeldung gelöscht haben – mit dieser Funktion wieder → "FORTSETZEN".

■ Debug Modus

■ "VORBEREITEN EINZELSCHRITT"

Das schrittweise Abarbeiten (Tracing) eignet sich vor allem für den Test von neu entwickelten Programmen und kann bei der Fehlersuche sehr hilfreich sein.

Klicken Sie auf "ENTWICKLUNG" → "VORBEREITEN EINZELSCHRITT" und das geöffnete Programm wird für den Debug-Mode vorbereitet: Es wird kompiliert und ein Debug-File erzeugt, das Programm wird in den VLT geladen und es werden alle ausführbaren Programmzeilen durch blaue Punkte gekennzeichnet. Nun sind auch die entsprechenden Menüpunkte aktiviert.



Haltepunkte setzen

Sie können vor jede mit einem blauen Punkt markierte Programmzeile durch Doppelklick einen Haltepunkt setzen. Dieser wird rot markiert. Die Programmausführung stoppt dann, bevor diese Programmzeile – die dann gelb markiert wird – ausgeführt wird.

Ein weiterer Doppelklick ändert die roten Haltepunkte wieder in blaue Markierungen für die Programmzeilen, die beim Tracing übersprungen werden sollen, im Debug-Mode also nicht angehalten wird.



ACHTUNG!

In Abhängigkeit von der Geschwindigkeit der Programmausführung und Kommunikation sollte die Anzahl der Haltepunkte auf ein vernünftiges Maß begrenzt werden. Maximal erlaubt sind 10 Haltepunkte.



ACHTUNG!

ON PERIOD Funktionen sollten Sie beim Debugging deaktivieren, da der interne Timer während der Pausen bei den Einzelschritten weiterläuft. Das Programm versucht dann die ON PERIOD Funktionen nachzuholen, was zu Problemen führen kann.

Kompatibilität

Die roten Haltepunkte ersetzen den bisherigen #DEBUG on/off Befehl, der ab

VLT5000/SyncPos Software Version 3.xx/4.2x

VLT5000/FluxSyncPos Software Version 5.xx/4.2x

nicht mehr ausgeführt wird.

Vorhandene #DEBUG Befehle müssen nicht unbedingt aus dem Programm entfernt werden, da sie ignoriert werden.

Wenn ein VLT mit einer früheren Version eingesetzt wird, wird die Programmausführung mit "ENTWICKLUNG" → "VORBEREITEN EINZELSCHRITT" gestartet und mit → "EINZELSCHRITT" bzw. [F9] schrittweise abgearbeitet. Die anderen Debug-Funktionen wie Start oder Überwachung anzeigen sind nicht anwählbar.

Variablen online ändern

Im Debug-Mode können Sie die Variablen während der Programmausführung ändern. Achten Sie dabei darauf, dass eine solche Änderung im Programm auch sinnvoll ist.

Klicken Sie mit der rechten Maustaste auf die Variable und setzen Sie im darauf folgenden Feld den gewünschten neuen Wert:



Variablen auslesen

Im Debug-Mode können Sie nach der Programmausführung den aktuellen Wert der Variablen auslesen.

Klicken Sie mit der linken Maustaste auf die Variable und der Wert wird so lange dargestellt, bis Sie den Mauscursor wieder bewegen.

```

● CVEL 10 // 1.
● oldi = IPOS // o.
● WH 1 (oldi = IPOS) DO
  ENDWRITE
  
```

■ Start (Debug) und Einzelschritt

Die Programmausführung stoppt beim ersten Haltepunkt und wartet auf eine Eingabe: Klicken Sie auf "ENTWICKLUNG" → "EINZELSCHRITT" oder tasten Sie [F9] um die nächste Programmzeile auszuführen.

Oder klicken Sie auf "ENTWICKLUNG" → "AUSFÜHREN" oder tasten Sie [F5] um das Programm bis zum nächsten Haltepunkt abzuarbeiten. Mit F9 hält das Programm also vor der nächsten Programmzeile, mit F5 vor jedem Haltepunkt.

Programmausführung im Debug-Mode Abbrechen
Klicken Sie auf "ENTWICKLUNG" → "ABBRECHEN" oder tasten Sie [ESC] um die Programmausführung sofort abzubrechen. Dabei werden auch eventuell aktive Fahrprozesse vorzeitig beendet.



NB!

Es wird mit der maximal zulässigen Verzögerung abgebremst.

Danach steht der Cursor in der Programmzeile, die als nächstes ausgeführt werden sollte. Sie können mit "ENTWICKLUNG" → "AUSFÜHREN" [F5] oder → "EINZELSCHRITT" [F9] fortfahren.

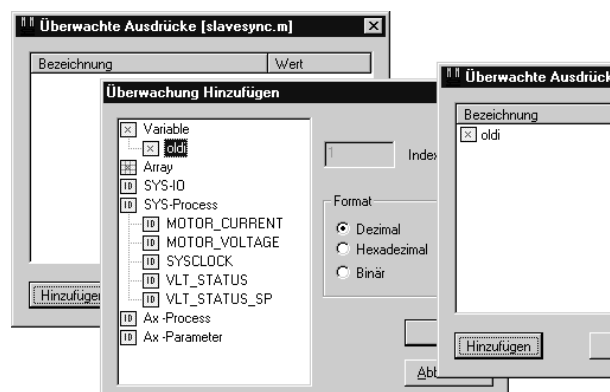
■ Beenden Debug

Mit "ENTWICKLUNG" → "BEENDEN DEBUG" wird die Programmausführung sofort beendet und der Debug-Modus verlassen. Die Markierung der Programmzeilen wird entfernt, die Haltepunkte weiterhin angezeigt, damit sie beim nächsten Debugging wieder benutzt werden können. Wenn Sie also Programmzeilen einfügen, "wandern" die Haltepunkte mit.

■ Überwachung anzeigen

Diese Funktion ermöglicht die Online-Überwachung der Variablen, Arrays, System- und Achsprozessdaten (gemäß der SYSVAR Indizes) und Achsenparameter.

Klicken Sie auf "ENTWICKLUNG" → "ÜBERWACHUNG ANZEIGEN" und im folgenden Dialogfenster auf → "HINZUFÜGEN". Das nächste Dialogfenster bietet die Variablen, Arrays und Parameter zur Auswahl:



Mit Doppelklick auf den gewünschten Typ, zum Beispiel Variable erhalten Sie alle im Programm verwendeten Variablen zur Auswahl. Markieren Sie den Ausdruck, der überwacht werden soll und wählen Sie aus, in welchem Format (Dezimal, Hexadezimal, Binär) er angezeigt werden soll. Dann klicken Sie auf OK.

Weitere Ausdrücke hinzufügen

Sie können weitere Ausdrücke zur "ÜBERWACHUNG" → "HINZUFÜGEN" und natürlich auch wieder → "Löschen". Maximal können 10 Ausdrücke gleichzeitig überwacht werden.



ACHTUNG!

Das Überwachungsfenster wird ständig aktualisiert. Daher sollte in Abhängigkeit von der Geschwindigkeit der Programmausführung und Kommunikation die Anzahl der überwachten Ausdrücke auf ein vernünftiges Maß begrenzt werden.



ACHTUNG!

Die Überwachung der Arrays ist auf die ersten 250 Elemente begrenzt.

Überwachungsfenster ändern

Wenn Sie die Größe dieses Dialogfensters ändern wollen, stellen Sie den Cursor an die untere rechte Ecke des Dialogfensters und klicken – sobald der Cursor seine Form ändert – und ziehen ihn in die gewünschte Richtung.

Überwachungsfenster schließen

Klicken Sie auf "ENTWICKLUNG" → "ÜBERWACHUNG SCHLIESSEN" oder auf das Schließen-Symbol im Dialogfenster. Wenn Sie es später erneut öffnen, werden die zuvor ausgewählten Ausdrücke wieder online überwacht und angezeigt.

■ "SYNTAXPRÜFUNG"

Prüfen Sie ein neu geschriebenes Programm bevor Sie es zum ersten Mal starten: Klicken Sie auf "ENTWICKLUNG" → "SYNTAXPRÜFUNG" und das Programm wird abgebrochen, sobald ein fehlerhafter Befehl gefunden wird. Im Kommunikationsfenster wird die Zeilennummer genannt und eine Fehlerbeschreibung ausgegeben. Automatisch wird der Cursor in die Zeile genau an die Position mit dem Syntaxfehler gestellt und das Programm stoppt an dieser Stelle.

Die Funktion "SYNTAXPRÜFUNG" erzeugt zusätzlich zum Check ein Debug-File und speichert es als „temp.ad\$“.

■ "IN DATEI KOMPILIEREN"

Mit → "IN DATEI KOMPILIEREN" wird ein aktuelles File kompiliert und als binäres File gesichert. Ein Dialogfenster "SPEICHERN UNTER" ermöglicht es dem Anwender, den Datei-Namen auszuwählen, unter dem die Datei gespeichert werden soll. Als Standard wird der Programmname mit der Extension „.bin“ angeboten.



ACHTUNG!

Diese Funktion steht nur Verfügung, wenn der CREATEBIN Parameter in der Datei APOS.DAT auf 1 gesetzt ist.

■ "ABBRUCH ALLE"

Falls Sie Programme in mehreren VLTs ausführen, klicken Sie auf "ENTWICKLUNG" → "ABBRUCH ALLE", um die laufenden Programme abzuberechnen.

■ "STEUERUNG AUSWÄHLEN"

Wenn Sie mehrere VLTs konfiguriert haben, wählen Sie mit "ENTWICKLUNG" → "STEUERUNG AUSWÄHLEN" im folgenden Dialogfeld den bzw. die VLTs aus, für die Sie Programme laden und starten wollen. Markieren Sie dazu die Nummer des VLTs und klicken Sie auf "OK".

■ "SCHNITTSTELLE SCHLIESSEN"

Mit → "SCHNITTSTELLE SCHLIESSEN" kann eine zurzeit offene Motorcontroller-Schnittstelle geschlossen werden. – Gibt es keine offene Schnittstelle, hat diese Funktion keine Auswirkung.

■ "BEFEHLSHILFE"

Die Befehlshilfe beinhaltet nicht nur alle Befehle alphabetisch geordnet, sondern zeigt auch zu jedem Befehl passend die notwendigen Eingabefelder und bildet automatisch die richtige Syntax für jeden Befehl, die Sie ganz einfach in Ihr Programm "EINFÜGEN". Außerdem können Sie den VLT per Teach-in programmieren.

Stellen Sie im Editierfenster die Eingabemarke an die Stelle, wo Sie den oder die neuen Befehle einfügen wollen, klicken Sie auf "ENTWICKLUNG" → "BEFEHLSHILFE" und wählen Sie im Dialogfeld den benötigten Befehl aus, z.B. POSA.

"HILFE"

Sie können sich in diesem Dialogfeld auch detailliert über jeden Befehl informieren: Klicken Sie auf "HILFE" oder drücken Sie [F1] und Sie erhalten die Informationen zum markierten Befehl.

Sobald Sie einen Befehl markieren zeigt die Vorschau die genaue Syntax des Befehls. Sie haben nun drei Alternativen zur Auswahl, die Sie beim Programmieren beliebig mischen können:

"EINFÜGEN"

Entweder Sie fügen den Befehl sofort in Ihr Programm ein: Klicken Sie auf "EINFÜGEN".

"JETZT AUSFÜHREN"

Oder klicken Sie auf "JETZT AUSFÜHREN" und testen diesen Befehl, bevor Sie ihn in Ihr Programm "EINFÜGEN".

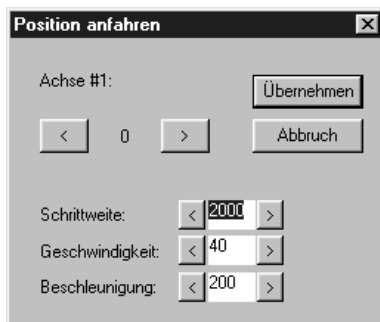


ACHTUNG!

Freigegebene Antriebe laufen an.

"POSITION ANFAHREN"

Oder Sie nutzen die Teach-in-Funktion und klicken auf "POSITION ANFAHREN": Im Dialogfeld wird die aktuelle Position der Achse angezeigt. Klicken Sie auf das Vorwärts- > oder auf das Rückwärtssymbol < und fahren Sie den Antrieb an die gewünschte Position: schrittweise mit einzelnen Mausklicks, Dauerfahrt durch Festhalten der Maustaste.



Wenn der Antrieb die gewünschte Position erreicht hat, klicken Sie auf "ÜBERNEHMEN" und der Wert wird in das Dialogfeld zur Achse eingetragen.



ACHTUNG!

Generell gilt, dass die Werte, die Sie beim Programmieren vorgeben, nicht auf zulässige Bereiche geprüft werden. Aufgrund der vielseitigen Anwendungsmöglichkeiten und verschiedenen Motorgößen ist dies nicht möglich und auch nicht erwünscht.

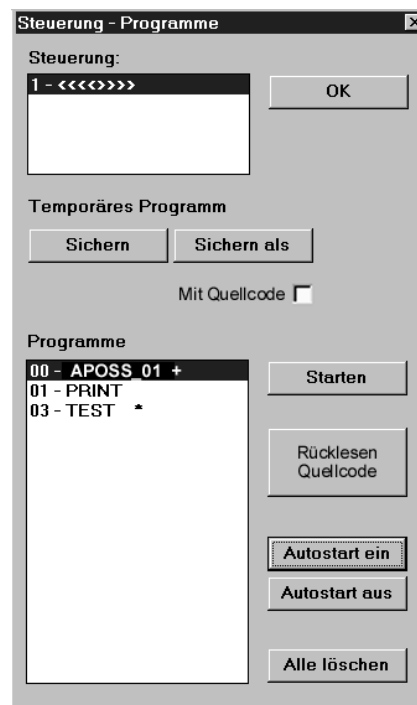
■ Menü "STEUERUNG"



Mit den Funktionen im Menü "STEUERUNG" verwalten Sie Ihre Programme: Sie speichern oder löschen die Programme im EEPROM der Option und kennzeichnen ein Programm für einen Autostart. Außerdem können Sie hier alle Parameter einstellen, verschiedene Resets durchführen und für jeden angeschlossenen VLT einen Namen vergeben.

■ "PROGRAMME"

Klicken Sie auf "STEUERUNG" → "PROGRAMME" und im Dialogfeld sehen Sie im ersten Feld alle VLTs. Es ist der VLT markiert, mit dem das Programm im Editierfenster gerade verbunden ist. Sie können natürlich auch einen anderen VLT markieren und bearbeiten.



"PROGRAMM" → "SICHERN"

Immer wenn Sie ein Programm ausführen, das heißt in den VLT laden, wird es in einen temporären Bereich im RAM geladen, der mit jedem weiteren Ausführen überschrieben wird. Sie können das zuletzt ausgeführte "TEMPORÄRE PROGRAMM" jetzt dauerhaft → "SICHERN".



ACHTUNG!

Vergessen Sie nicht, die Programmdatei entweder auch auf der Festplatte des PCs oder zusätzlich → "MIT QUELLCODE" zu speichern, da Sie die compilierte Quelldatei im VLT nicht mehr bearbeiten können.

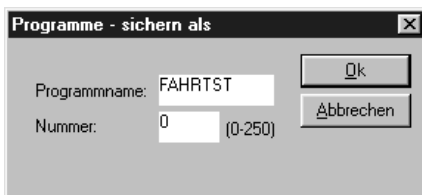
Klicken Sie auf "SICHERN" und geben Sie im folgenden Dialogfeld einen Namen ein oder bestätigen Sie den vorgeschlagenen Dateinamen. Die Programmnummer wird automatisch vergeben.



ACHTUNG!

Versucht man, ein Programm zu sichern während ein Programm bereits aktiv ist, dann kann das neue Programm nicht gesichert werden. In einem solchen Fall wird ein Dialogfenster geöffnet, das die Möglichkeit bietet, das aktive Programm zu → "UNTERBRECHEN". Das neue Programm wird dann gespeichert.

Klicken Sie auf "SICHERN ALS" und Sie können zusätzlich zum Namen auch die Programmnummer (0 bis 127) selbst bestimmen.



Über diese Programmnummer kann ein beliebiges Programm auch über die Eingänge, zum Beispiel von einer SPS aus, gestartet werden. Dazu sind die Eingänge mit "STEUERUNG" → "PARAMETER" → "GLOBAL" entsprechend zu setzen.

"SICHERN" → "MIT QUELLCODE"

Wenn das Kontrollkästchen aktiviert ist, wird zusätzlich zur compilierten und direkt ausführbaren Programmdatei der Quellcode im VLT gesichert. Und Sie können diesen bei Bedarf wieder zurücklesen und auf dem PC in einer Datei speichern.

Wenn der Programm-Quellcode in die Steuerung geladen wird, werden auch Included Files in den Quellcode geschrieben und mit diesem zusammen in die Steuerung geladen. So kann ein komplettes Programm in der Steuerung gespeichert werden.

Klicken Sie auf → "SPEICHERN UNTER" und geben Sie einen Namen in das folgende Dialogfeld ein oder bestätigen Sie den Dateinamen. Der vorgeschlagene Name enthält das Datum und die Uhrzeit, damit beim Zurücklesen keine aktuelle Datei überschrieben werden kann.

Der Quellcode wird im Flash-EPROM gespeichert. Falls dort nicht genügend Platz für den Quellcode ist, erhalten Sie eine Meldung und müssten dann andere Programmdateien löschen, bevor Sie die neue speichern.

Alle mit Quellcode gesicherten Programme werden mit einem + gekennzeichnet.

"PROGRAMM" → "STARTEN"

Sie können in diesem Dialogfeld ein Programm auswählen und direkt starten.

"RÜCKLESEN QUELLCODE"

Alle mit '+' gekennzeichneten Programme können Sie im Quellcode-Format wieder aus der Steuerung auslesen und auf Ihrem PC zur weiteren Verwendung ablegen.

Wählen Sie das gewünschte Programm aus und klicken Sie auf "RÜCKLESEN QUELLCODE". Sie können die Datei dann wie gewohnt bearbeiten oder für andere Steuerungen duplizieren.

"AUTOSTART EIN / AUS"

Mit "AUTOSTART" kennzeichnen Sie ein Programm, das künftig nach dem Einschalten des VLTs sofort gestartet wird. Markieren Sie das gewünschte Programm und klicken Sie auf "AUTOSTART EIN". Das ausgewählte Programm wird mit einem * gekennzeichnet.

Wenn Sie einen gesetzten Autostart aufheben wollen, klicken Sie auf "AUTOSTART AUS" oder Sie kennzeichnen gleich ein anderes Programm.

Um mehrere Programme mit Autostart ablaufen zu lassen, nutzen Sie den Parameter PRGPAR (102). Damit können Sie festlegen, welches Programm nach Ablauf des per Autostart ausgeführten Programmes gestartet werden soll.

Wenn in PRGPAR (102) oder I_PRGSTART (103) nichts anderes festgelegt ist, wird immer wieder das mit "AUTOSTART" gekennzeichnete Programm gestartet.

Ein gesetzter "AUTOSTART" wirkt sich wie folgt aus:

Wenn bei einem Kaltstart kein Fehler vorliegt (Ausnahme Schleppfehler, Endschalter-Fehler und SW-Endschalter-Fehler) wird das entsprechende Autostartprogramm gestartet.

Wird das Autostartprogramm durch einen Abbruch des Benutzers gestoppt wird es nicht wieder gestartet, es sei denn es findet ein neuer Kaltstart statt. In diesem Fall wird auch kein Programm auf Grund von Eingängen oder PRGPAR (102) gestartet.

Wird das Autostartprogramm durch einen Fehler abgebrochen (weil keine ON ERROR Routine definiert wurde) oder normal beendet, wird anschließend geprüft, ob ein Start durch Eingänge vorgesehen oder ob der Parameter PRGPAR (102) gesetzt ist. Wenn ja, wird das entsprechende Programm ausgeführt, bzw. auf den Start-Eingang gewartet. Wenn nicht, wird das Autostartprogramm wieder von vorne begonnen. Daraus folgt:

Einmalige Ausführung des Autostartprogrammes

Wenn prinzipiell ein Start von Programmen über den Parameter PRGPAR (102) oder über Eingänge vorgesehen ist, wird das Autostartprogramm nur einmal ausgeführt (zum Beispiel für HOME-Aufgaben).

Wiederholende Ausführung des Autostartprogrammes

In den anderen Fällen wird das Autostartprogramm immer wieder gestartet.

So kann man auch ein Programm mit einem EXIT Befehl einfach wieder von vorne starten. Dies ist dann nützlich, wenn man in einer Fehlersituation (ON ERROR) nicht mit RETURN fortfahren, sondern zum Beispiel eine erneute Homefahrt erzwingen will. Es sollte allerdings darauf geachtet werden, dass kein Fehler (außer Schleppfehler, Endschalte-Fehler und SW-Endschaltes-Fehler) vorliegt, da sonst das Autostartprogramm nicht wieder gestartet wird.

Verkettung von Autostartprogrammen

Das Starten über Parameter PRGPAR (102) kann natürlich auch zur Verkettung benutzt werden: Nachdem ein Programm gestartet wird, kann die mit PRGPAR (102) definierte Programmnummer umge-
setzt und so bestimmt werden, welches Programm als nächstes ausgeführt werden soll.



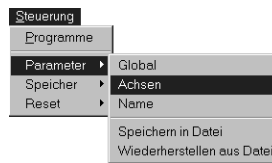
ACHTUNG!

Wenn kein Autostartprogramm definiert ist, kann auch kein Programm über PRGPAR (102) gestartet werden; dies erfordert immer ein beendetes Autostartprogramm.

"ALLE LÖSCHEN"

Klicken Sie auf "ALLE LÖSCHEN", wenn Sie alle Programme im VLT löschen wollen. Vergewissern Sie sich zuvor, dass Sie die Programme noch im PC zur Sicherheit oder für das Archiv gespeichert haben.

■ "STEUERUNG" → "PARAMETER"



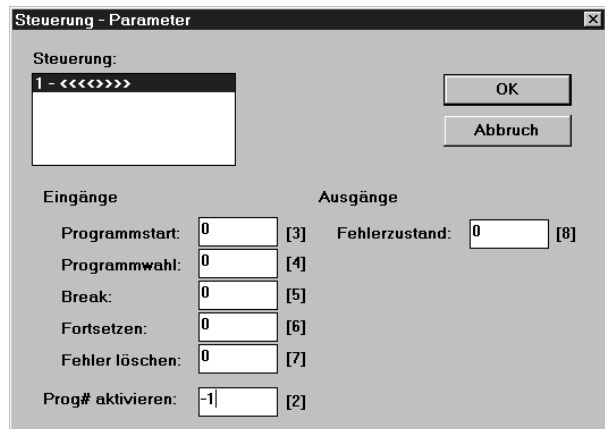
Die Parameter sind in zwei Gruppen unterteilt: den globalen Parametern und den Achsparametern. Sowohl die Übersicht als auch die Details aller Parameter und die Parameterkennungen mit den jeweiligen Werkseinstellungen finden Sie im Kapitel Software-Referenz, Abschnitt Parameter.

Die Dialogfelder der Parametergruppen bieten aber auch einen direkten Zugriff auf die Online-Hilfe: Stellen Sie den Mauscursor in eines der Eingabefelder, drücken Sie [F1] und die Informationen zu diesem Parameter werden eingeblendet.

■ Globale Parameter

Zu den globalen Parametern gehören die Funktionen der Ein- und Ausgänge (Gruppe GLI) und die Standard Parameter (Gruppe GLS). Die interne Parameter-Nummer führt Sie im nächsten Kapitel in aufsteigender Reihenfolge durch die Parameter-Referenz mit den Detail-Informationen.

Markieren Sie den VLT den Sie bearbeiten wollen. Sie können jeden voreingestellten Wert einzeln verändern. Klicken Sie auf "OK" um die Änderungen als Benutzerparameter in den VLT zu laden.



Mit "RESET" → "PARAMETER" im Menü "STEUERUNG" können Sie wieder auf die Werkseinstellungen zurücksetzen, allerdings werden dabei alle – also auch alle Achsparameter – auf die ab Werk eingestellten Werte zurückgesetzt.

■ Achsparameter

Die Achsparameter gelten immer für alle Programme einer Steuerung. Die Einheiten der Parameterwerte und alle anderen Informationen zu den Parametern finden Sie in der Parameter-Referenz nach der internen Parameternummer aufsteigend geordnet.

Es gibt zwei Möglichkeiten, die Parameter einzustellen oder zu ändern:

Achsparameter online einstellen oder ändern

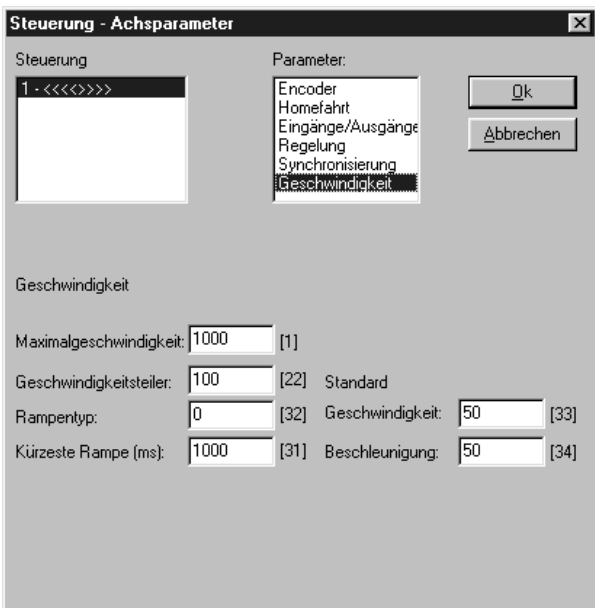
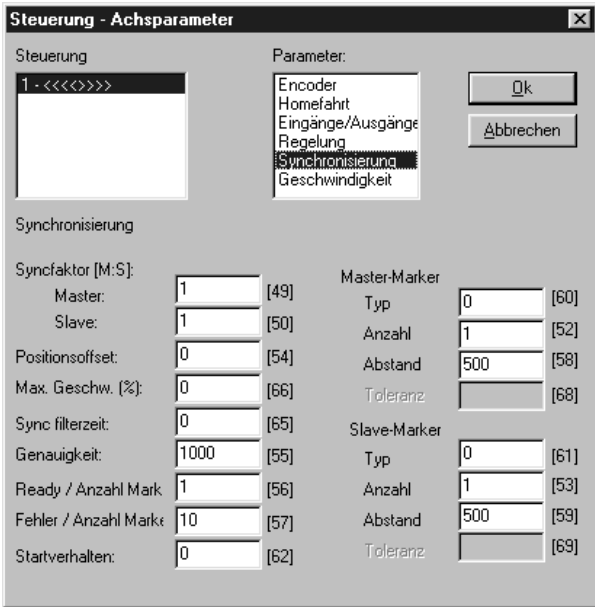
Klicken Sie auf "STEUERUNG" → "PARAMETER" → "ACHSEN" und markieren Sie im folgenden Dialogfeld die Steuerung, deren Parameter Sie sehen oder ändern wollen. Wählen Sie außerdem im Feld "PARAMETER" den Typ aus:

Encoder	AXE
Homefahrt	AXH
Eingänge/Ausgänge (I/O)	AXI
Regelung	AXR
Synchronisierung	AXS
Geschwindigkeit (Velocity)	AXV

Sie können jeden Parameter ändern und mit Klicken auf "OK" wieder in den VLT laden. Sie können aber auch sofort einen anderen VLT auswählen, die Parameter ändern und dann mit "OK" alle Änderungen gleichzeitig in den VLT laden.

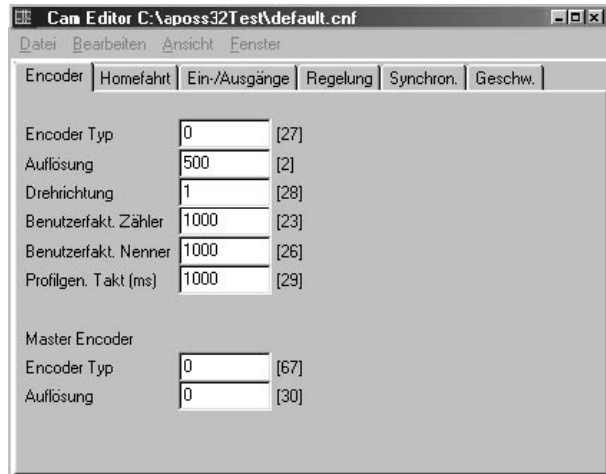
Achsparameter mit den Werkseinstellungen

In dieser Parametergruppe werden den Ein- und Ausgängen – in Abhängigkeit von der Anwendung – feste Funktionen zugewiesen. Falls erforderlich können Sie hier auch Software-Endschalter definieren.



Achsparameter einer Konfigurationsdatei ändern
Neben der Möglichkeit die Achs- und globalen Parameter online zu ändern, können auch alle Parametereinstellungen einer gespeicherten Konfigurationsdatei geändert werden. Dazu öffnen Sie den → "CAM-EDITOR" und ändern die Parameter in den entsprechenden Registerkarten.

Diese Änderungen betreffen jedoch nur die Konfigurationsdatei, nicht aber die Parameter in der Steuerung. Um die in der Konfigurationsdatei geänderten Einstellungen in die Steuerung zu übernehmen, muss die Konfigurationsdatei in die Steuerung geladen werden: "STEUERUNG" → "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI".



Achsparameter rücksetzen

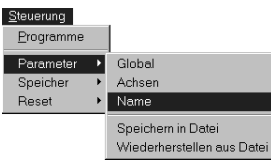
Wenn Sie für alle Achsparameter wieder die Standardeinstellungen wünschen, klicken Sie auf "STEUERUNG" → "RESET" → "PARAMETER".



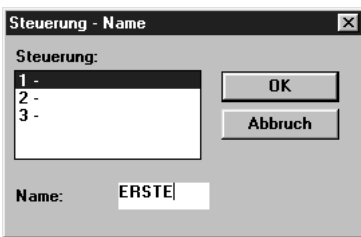
ACHTUNG!

Dabei werden aber auch die Globalen Parameter auf die Werkseinstellungen zurückgesetzt.

■ "PARAMETER" → "NAME"

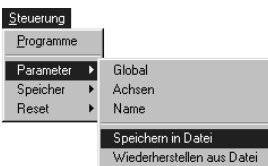


Sie können zusätzlich zur Adresse für jeden VLT einen Namen eingeben oder den vorhandenen ändern. Klicken Sie auf "STEUERUNG" → "PARAMETER" → "NAME" und wählen Sie im Dialogfeld den VLT aus, den Sie bearbeiten wollen:



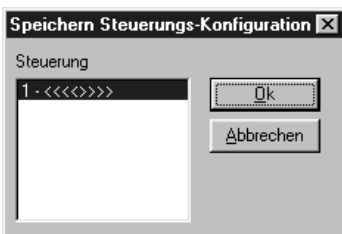
Geben Sie im Feld "NAME" einen maximal 8-stelligen Namen für den VLT ein oder überschreiben Sie den vorhandenen und klicken Sie auf "OK".

■ "PARAMETER" → "SPEICHERN IN DATEI"



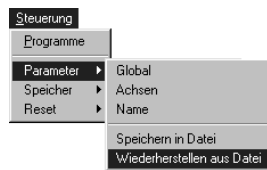
Mit "PARAMETER" → "SPEICHERN IN DATEI" sichern Sie die Benutzerparameter inklusive der Arrays in eine Datei mit der Extension „.cnf“. So können Sie schnell die Parameter in andere VLTs laden oder später jederzeit wieder in den VLT laden, zum Beispiel nach EEPROM löschen.

Klicken Sie auf → "SPEICHERN IN DATEI" und wählen Sie im folgenden Dialogfeld den VLT aus, dessen Parameter Sie sichern wollen.

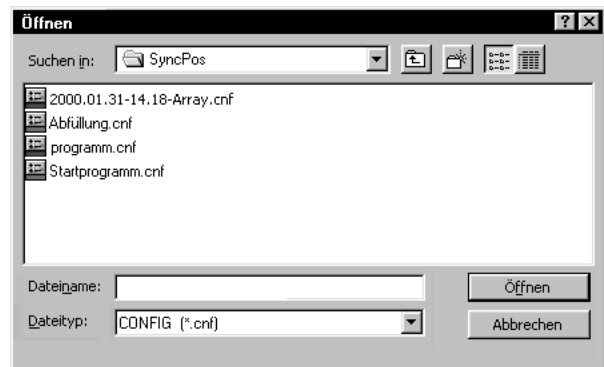
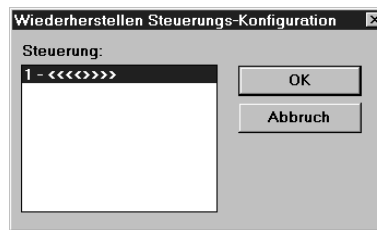


Geben Sie einen Namen ein oder bestätigen Sie den vorgeschlagenen. Dieser enthält das Datum und die Uhrzeit, so dass ein unbeabsichtigtes Überschreiben der Parameter beim Wiederherstellen ausgeschlossen wird.

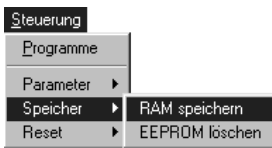
■ "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI"



Klicken Sie auf "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI" und wählen Sie den VLT aus, in den die Daten geladen werden sollen.

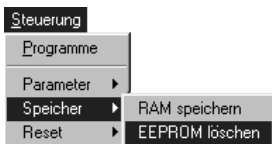


Dann wählen Sie die Datei aus und klicken auf "ÖFFNEN". Sofort werden die gespeicherten Benutzerparameter inklusive der Arrays in den VLT geladen.

■ "SPEICHER" → "RAM SPEICHERN"


Die Funktion RAM speichern wird normalerweise nicht benötigt, da Programme und Parameter automatisch gesichert werden.

Mit → "RAM SPEICHERN" können Sie zusätzlich aktuelle Array-Werte mit in das EEPROM sichern. RAM speichern entspricht dem Befehl SAVEPROM, denn es werden alle Programme, Parameter und Arrays gesichert.

■ "SPEICHER" → "EEPROM LÖSCHEN"


Das EEPROM in der SyncPos-Option werden Sie löschen, wenn Sie wieder die Werkseinstellungen benutzen wollen. Wenn Sie nur die Arrays zurücksetzen wollen, sollten Sie (ab Version 2.1x) "RESET" → "ARRAYS" benutzen.

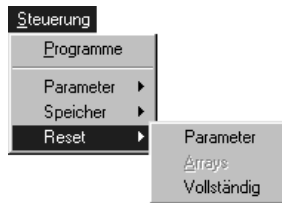

ACHTUNG!

Wenn Sie das EEPROM löschen, werden alle Parameter auf die Werkseinstellungen zurückgesetzt. Allerdings erst nach dem Ausschalten des VLTs.

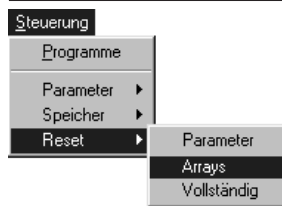

ACHTUNG!

Beachten Sie also Folgendes, wenn Sie das EEPROM löschen:

- Prüfen Sie, ob Sie alle noch benötigten Programme im PC gesichert haben, um sie nach dem Löschen des EEPROMs wieder in den VLT laden zu können.
- Prüfen Sie, ob die Parameter aller angeschlossenen VLTs in einer Datei im PC gesichert sind.
- Klicken Sie dann auf "SPEICHER" → "EEPROM LÖSCHEN".
- Laden Sie die Benutzerparameter und die benötigten Programme wieder in den VLT (bzw. in die VLTs).

■ "RESET" → ...

"RESET" → "PARAMETER"

Mit "RESET" → "PARAMETER" werden alle Globalen Parameter und alle Achsparameter im SyncPos Motion Controller auf die Werkseinstellungen zurückgesetzt.

"RESET" → "ARRAYS"


Mit "RESET" → "ARRAYS" können Sie alle Arrays im RAM löschen, ohne – wie bisher – auch die Parameter etc. zu löschen. Dieser neue Befehl bewirkt das Gleiche, wie der Befehl DELETE ARRAYS.


ACHTUNG!

Wenn Sie anschließend ein SAVE ARRAYS durchführen, werden auch die Arrays im EEPROM überschrieben!

"RESET" → "VOLLSTÄNDIG"

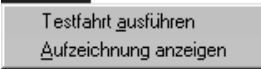
Mit "RESET" → "VOLLSTÄNDIG" werden nicht nur die Parameter, sondern auch die Programme und Arrays gelöscht und die SyncPos-Option auf die Grundeinstellungen ab Werk zurückgesetzt ...


ACHTUNG!

... und zwar sofort und nicht erst nach dem Aus- und Wiedereinschalten des VLTs wie bei EEPROM löschen.

Menü "TESTFAHRT"

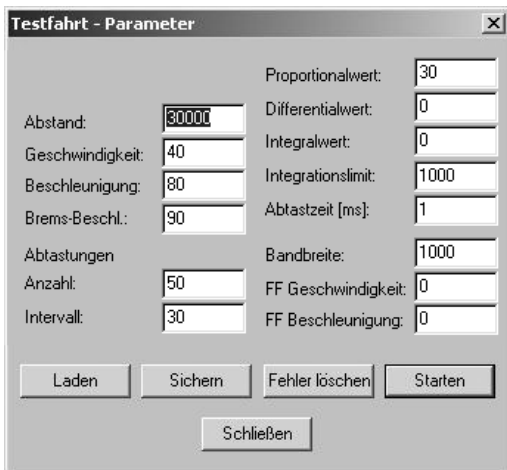
Testfahrt



Das Menü Testfahrt bietet die Funktion "TESTFAHRT" → "AUSFÜHREN" von der Eingabe der Testfahrtparameter bis zur grafischen Darstellung der Testfahrtergebnisse. Wenn Sie mit TESTSETP eine Testfahrt mit anderen Parametern definiert haben, können Sie sich diese Ergebnisse nach der Durchführung (TESTSTART) mit "TESTFAHRT" → "AUFZEICHNUNG ANZEIGEN" ebenfalls grafisch darstellen lassen.

Testfahrt-Parameter festlegen

Klicken Sie auf ""TESTFAHRT" → "AUSFÜHREN" und geben Sie im Dialogfeld die Testfahrt-Parameter ein:



Verändern Sie nach Möglichkeit von einer Messung zur anderen immer nur einen Parameter und prüfen Sie die Auswirkung.

Fahrweg

Bestimmen Sie den Fahrweg in Benutzereinheiten und nutzen Sie die gesamte Aufzeichnungsdauer bestmöglich: Die Zahl der Messpunkte multipliziert mit der Zeitdifferenz zwischen zwei Messungen ergibt die gesamte Aufzeichnungsdauer und bestimmt somit die grafische Darstellung. Für eine möglichst optimale Auswertung der Diagramme sollte der Verfahrensweg so gewählt werden, dass die Endposition ungefähr in 80 % der gesamten Aufzeichnungsdauer erreicht wird. So sind auch Überschinger in der Zielposition noch gut erkennbar.

Beispiel:

50 Messungen in 30 ms Intervallen = 1,5 sec Aufzeichnungsdauer

Geschwindigkeit, Beschleunigung und Verzögerung
Die Testfahrt-Parameter Geschwindigkeit, Beschleunigung und Verzögerung werden in Prozent der jeweiligen Maximalwerte eingegeben.

Führen Sie die Messvorgänge mit den für die Steuerung am häufigsten benötigten Werten für Geschwindigkeit, Beschleunigung und Verzögerung durch.

Um das Überschwingungsverhalten bei Erreichen der Endgeschwindigkeit bewerten zu können, ist ein trapezförmiger Geschwindigkeitsverlauf anzustreben. Eventuell muss hierzu das Abtastintervall erhöht oder die Endgeschwindigkeit reduziert werden.



ACHTUNG!

Prüfen Sie – bevor Sie mit der Optimierung des Regelverhaltens beginnen – ob die Maximalgeschwindigkeit und die Maximalbeschleunigung erreicht wird.

Anzahl der Messungen

Die Anzahl der Abtastungen sowie das Abtastintervall bestimmen die gesamte Messdauer. 50 bis 100 Messpunkte sind für eine optimale Bildschirmdarstellung ausreichend.

Die Zahl der maximal möglichen Messpunkte ist durch den internen Speicher des SyncPos Motion Controllers sowie durch eventuell darin abgelegte Programme begrenzt. Wenn der Speicherplatz für die gewünschten Abtastungen nicht ausreicht, müssen Sie zuvor die im SyncPos Motion Controller abgelegten Programme mit "STEUERUNG" → "PROGRAMME" → "ALLE LÖSCHEN".

Datenabtastintervall

Wählen Sie ein dem System angepasstes Abtastintervall, für Frequenzrichter zum Beispiel 20 bis 30 ms.

Für dynamische Anwendungen kann das Abtastintervall bis zu 1 ms verkleinert werden. Ein möglichst kurzes Abtastintervall ist für Servomotoren einzustellen.

Für das Aufzeichnen langsamer oder sehr langer Bewegungsvorgänge kann die Zeitdifferenz in Millisekunden natürlich erhöht werden, bis maximal 255 Millisekunden.



ACHTUNG!

Dieses Abtastintervall ist das interne Intervall zwischen den einzelnen Messungen, nicht das Abtastintervall der Steuerung.

■ "TESTFAHRT" → "AUSFÜHREN"
Sicherheitshinweise

Bringen Sie den Antrieb in die Ausgangsstellung, und zwar unbedingt bevor Sie das Testfenster öffnen.


ACHTUNG!

Bei der Regleroptimierung mit der Funktion "TESTFAHRT" wird der Antrieb nach dem Erreichen der Zielposition automatisch auf die Startposition zurückgefahren.

Falls Ihr Antrieb nicht reversieren darf, muss der Parameter REVERS (63) auf "keine Reversierung = 2" eingestellt werden.

Klicken Sie auf "STEUERUNG" → "PARAMETER" → "ACHSEN" und ändern Sie die Einstellung in der Parametergruppe Regelung.


ACHTUNG!

Achten Sie darauf, dass eventuell vorhandene Bremsen gelöst sind und keine Hindernisse innerhalb des Positionierweges sind.


ACHTUNG!

Falsch eingestellte Reglerparameter können den Motor und die Mechanik beschädigen. Eine Regleroptimierung darf deshalb nie ohne installierten NOT-AUS-Schalter durchgeführt werden.

"TESTFAHRT" starten

Klicken Sie auf "TESTFAHRT" → "AUSFÜHREN" und das Dialogfeld mit den Testfahrt-Parametern wird geöffnet: Die zuletzt gesicherten Testfahrt-Parameter und die aktuellen Achsparameter sind hier bereits eingetragen.



Beginnen Sie die Testreihe mit „stabilen“ Reglerparametern. Wenn die standardgemäß vorgegebenen Parameter bereits in der Ausgangsstellung zu einem stark schwingenden Antrieb führen, wählen Sie den **Proportional-** und **Differentialfaktor** klein (ca. 20) und setzen den **Integralfaktor** auf Null. Von diesen Werten ausgehend optimieren Sie dann die Steuerung.

Wenn die Möglichkeit besteht, sollten Sie die Steuerung zunächst mit Motor und Getriebe so optimieren, dass Sie einigermaßen unkritische Werte erhalten. Dann schließen Sie die Mechanik mit der Last an und führen die Feinoptimierung durch.

Klicken Sie auf "STARTEN": Die Testfahrt wird durchgeführt, die aktuellen Positionswerte etc. gespeichert und am Ende der Testfahrt an den PC zur Auswertung übergeben.


ACHTUNG!

Beobachten Sie das Motorverhalten und die Motortemperatur: Bei starkem Schwingungsverhalten oder übermäßiger Erwärmung des Motors muss der Bewegungsvorgang mit NOT-AUS vorzeitig abgebrochen und die Reglerparameter müssen anders gewählt werden.

Nach einem NOT-AUS müssen Sie den Antrieb für weitere Testfahrten erst wieder in die Ausgangsposition bringen. Reduzieren Sie den **Proportionalfaktor**, ggf. zusätzlich den **Differentialfaktor**, bevor Sie die nächste Testfahrt bzw. Messung starten.

Nach Beendigung der Messung werden die Messdaten automatisch an den PC übertragen und der Antrieb kehrt währenddessen mit reduzierter Geschwindigkeit in die Ausgangsposition zurück. Die Diagramme werden automatisch dargestellt.

Mit "WIEDERHOLEN" wird wieder das Dialogfeld Testfahrt-Parameter dargestellt und Sie können ohne Umwege die Parameter ändern und eine neue Testfahrt starten.

"FEHLER LÖSCHEN"

Es kann keine Testfahrt gestartet werden, wenn die Motorsteuerung Fehler meldet. Mit → "FEHLER LÖSCHEN" im Dialogfeld Testfahrt-Parameter können aber alle aktuell anliegenden Fehlerbedingungen in der Motorsteuerung gelöscht werden. Wenn Fehler anliegen, ist der "START"-Button grau und die Testfahrt kann nicht gestartet werden.

Testfahrt-Parameter sichern und laden

Klicken Sie auf "SICHERN", um die Testfahrt-Parameter im VLT zu speichern. Wenn Sie nun bei den weiteren Testfahrten schlechtere Regelungsergebnisse erzielen, können Sie die gespeicherten Parameter wieder → "LADEN".

■ "AUFZEICHNUNG ANZEIGEN"

Wenn Sie mit TESTSETP eine Testfahrt mit anderen Parametern definiert haben, können Sie sich diese Ergebnisse nach der Durchführung (TESTSTART) mit "TESTFAHRT" → "AUFZEICHNUNG ANZEIGEN" auch grafisch darstellen lassen. Soweit dies bei den von Ihnen gewählten Parametern sinnvoll ist, das heißt dass sich die Ergebnisse mit den vier Grafiken auch darstellen lassen.

Die vier Grafiken bzw. sieben Kurven werden dazu wie folgt genutzt:

- (1) Die Kurve Istposition zeigt die Werte des Index w1 (siehe TESTSETP),
- (2) die Kurve Sollposition zeigt die Werte des Index w2,
- (3) und die Stromkurve die Werte des Index w3.
- (4) Die Kurve Ist-Geschwindigkeit zeigt die Differenz der Aufzeichnungswerte zu den Werten von w1.

Im Fall der Aufzeichnung von Positionsdaten also die Änderung der Position in ms, das ist die Geschwindigkeit.

- (5) Die Kurve Soll-Geschwindigkeit zeigt die Differenz der Aufzeichnungswerte zu w2, im Fall der Aufzeichnung von Positionsdaten also die Änderung der Position in ms = Geschwindigkeit.
- (6) Die Kurve Ist-Beschleunigung zeigt die Differenz der Aufzeichnungswerte zur Ist-Geschwindigkeit (siehe 4), im Fall der Aufzeichnung von Positionsdaten also die Änderung der Geschwindigkeit in ms = Beschleunigung.
- (7) Die Kurve Soll-Beschleunigung zeigt die Differenz der Aufzeichnungswerte zur Soll-Geschwindigkeit (siehe 5), im Fall der Aufzeichnung von Positionsdaten also die Änderung der Geschwindigkeit in ms = Beschleunigung.

■ Fahrdiagramme auswerten

Prüfen Sie die maximale Position, die maximale Geschwindigkeit, die Anzahl der „Überschwinger“, sowie die Dauer des Einschwingvorganges.

Zu jeder Grafik werden die wichtigsten Testfahrt-Parameter, die eingestellten Maximalwerte und die tatsächlich erreichten Werte angezeigt:

- Geschwindigkeit in Benutzereinheiten/ms,
- Beschleunigung in Benutzereinheiten/ms²,
- die maximale Soll-Ist-Abweichung (Regelabweichung in Benutzereinheiten), die während der Fahrt herrschte,
- die tatsächliche Positionsabweichung im Ziel,
- Benutzereinheit in qc und das
- Datenabtastintervall in ms.

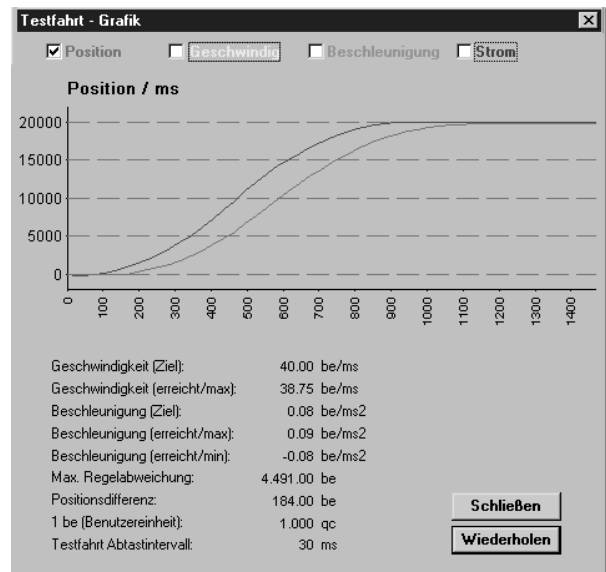
Klicken Sie in das jeweilige Eingabefeld um eine der anderen Grafiken zu sehen, z. B. vor Position. Sie können sich auch zwei oder alle vier Grafiken gleichzeitig ansehen. Dann werden allerdings keine Einheiten auf der x-Achse dargestellt.

Wiederholen

Wenn Sie auf "WIEDERHOLEN" klicken, wird wieder das Dialogfeld Testfahrt-Parameter dargestellt: Sie können die Parameter ändern und eine neue Testfahrt starten.

Testfahrt-Grafik Position

Die Grafik zeigt mit der dunkelroten (dunkleren) Kurve die Sollpositionen und mit der roten (helleren) Kurve die tatsächlich erreichten Positionen.



Testfahrt-Grafik Geschwindigkeit

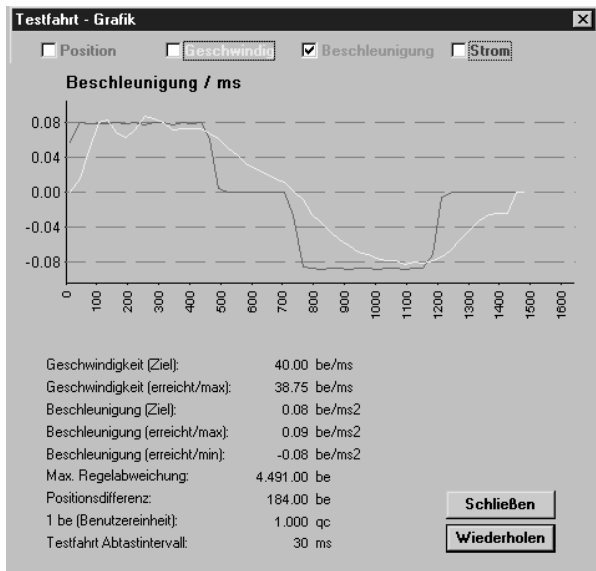
Die gelbe (helle) Kurve zeigt den erfassten Geschwindigkeitsverlauf, die braune (dunkle) Kurve die gewünschte trapezförmige Sollkurve.



In Sonderfällen kann der trapezförmige Geschwindigkeitsverlauf zu einer Dreiecksform entartet sein. Dieser Effekt tritt auf, wenn die Positionierdistanz zu gering ist, um bei der gewünschten Beschleunigung die Maximalgeschwindigkeit zu erreichen.

Testfahrt-Grafik Beschleunigung

Die hellgrüne Kurve zeigt den aktuellen Beschleunigungsverlauf, die dunkle die gewünschte treppenförmige Sollkurve beim Beschleunigen und Abbremsen.



Testfahrt-Grafik Strom

Die blaue Linie zeigt die aktuelle Stromaufnahme des Motors.



Menü CAM-Editor

Mit dem CAM-Editor werden die Kurvenprofile für beliebige Kurvenscheibensteuerungen erstellt. Die einzelnen Kurven werden durch Fixpunkte, Parameter für die Ein- und Auskuppelbewegung sowie Parameter für die Synchronisation mit Marker beschrieben. Für diese Eingabe und für andere Kurven-Informationen gibt es im CAM-Editor Registerkarten. In der Grafik werden die Kurven und Parameter visualisiert; außerdem können Sie die Fixpunkte auch interaktiv eingeben und manipulieren.

Während Sie mit dem CAM-Editor arbeiten, bleibt das SyncPos-Fenster geöffnet. Sie können beliebig hin- und herschalten, zum Beispiel um Parameter in der Steuerung nachzusehen oder die Anzahl der Array-Elemente in die DIM-Anweisung zu übernehmen.

Für eine Kurvenscheibensteuerung können eine oder mehrere Kurven erzeugt werden. Alle zu einer Anwendung erforderlichen Kurven werden in der cnf-Datei gespeichert und in die Steuerung geladen. Klicken Sie auf → "CAM-EDITOR" zum Starten des Editors. Das CAM-Editor-Fenster wird geöffnet und ...

Bevor Sie beginnen, eine Kurve zu editieren

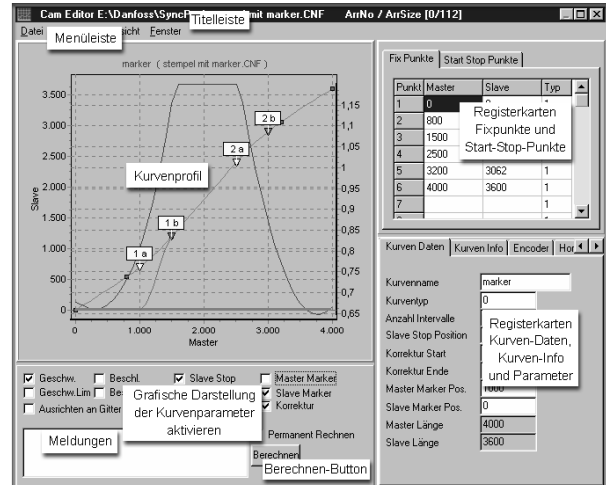
... sollten Sie die Parameter der Steuerung als cnf-File in den CAM-Editor laden. Denn dann werden die Parameter in der entsprechenden Registerkarte schon passend ausgefüllt. Sie erhalten diese Datei mit "STEUERUNG" → "PARAMETER SPEICHERN IN DATEI". Dabei werden auch evtl. vorhandene Arrays mit ausgegeben.

Wenn Sie kein cnf-File laden, werden die Werkseinstellungen des VLT eingestellt.

CAM-Editor-Fenster

Das CAM-Editor-Fenster ist in vier Bereiche aufgeteilt:

- Kurvenprofil-Grafik,
- Kontrollkästchen um die grafische Darstellung von Parametern und Berechnungsmodi auszuwählen und Berechnen-Button
- Registerkarten: Tabelle der "FIXPUNKTE" und der "START STOP PUNKTE",
- Registerkarten: "KURVEN-DATEN", "KURVEN-INFO" und alle Parameter gemäß den Dialogfeldern des SyncPos-Programmes: Encoder, Homefahrt, Eingänge/Ausgänge, Regelung, Synchronisation und Geschwindigkeit. Scrollen Sie nach links oder rechts, um alle Registerkarten zu sehen.



Das gesamte CAM-Editor-Fenster kann beliebig vergrößert oder verkleinert werden. Dabei bleiben die beiden rechten Bereiche für die alphanumerischen Eingaben gleich groß, während die Grafik vergrößert bzw. verkleinert wird und der linke untere Bereich bei starker Verkleinerung sogar beschnitten wird.

Wenn Sie in den zunächst nicht dargestellten Bereichen von Registerkarten, zum Beispiel in "SYNCHRONISATION" Eingaben vornehmen, wird vorübergehend die gesamte Karte dargestellt und die Grafik verkleinert. Klicken Sie auf "FENSTER" → "STANDARD", auf → "BERECHNEN" oder in ein anderes Feld im CAM-Editor, dann wird sofort wieder das Standardfenster dargestellt.

Sie können aber auch die einzelnen Bereiche in der Größe ändern, indem Sie mit dem Mauscursor auf die horizontale oder vertikale Trennlinie der Bereiche bewegen und klicken – sobald der Cursor seine Form ändert – und ihn dann in die gewünschte Richtung ziehen.

Titelleiste CAM-Editor

Die Titelleiste zeigt den Namen der cnf-Datei und den vollständigen Pfad.

Sobald eine Kurve zum ersten Mal mit → "SICHERN CNF ALS .." in eine cnf-Datei gespeichert wurde, sehen Sie rechts in der Titelleiste die Nummer (der Platz) des Arrays im cnf-File und die Anzahl der Array-Elemente, die Sie für die DIM-Anweisung im Programm benötigen:

Arr.Nr/ArrSize [0/350]

Menüleiste CAM-Editor

Die Menüleiste bietet die Menüs zum Laden und Speichern der Dateien, für verschiedene Einstellungen, zur Berechnung und Ansichtsoptionen.

■ Menü "DATEI" CNF


Das Menü Datei enthält die Befehle zum Erstellen, Öffnen und Speichern einer cnf-Datei mit den Kurven und Parametern; außerdem Funktionen zum Exportieren, Importieren und Drucken der cnf-Dateien. Alle Befehle erreichen Sie auch per Mausklick oder mit der Tastenkombination [ALT] und dem unterstrichenen Buchstaben, zum Beispiel [ALT] + [D] + [S], wenn Sie die "DATEI" → "SICHERN CNF" wollen.

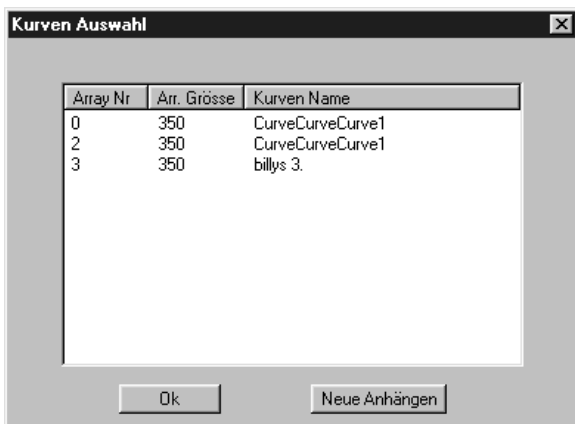
"NEUE CNF"

Mit "DATEI → NEUE CNF" wird eine „leere“ neue cnf-Datei bereitgestellt. Das heißt, hier sind in den Parametern die VLT-Werkseinstellungen eingetragen.

"LADEN CNF"

Mit "DATEI → LADEN CNF" bzw. öffnen Sie eine vorhandene cnf-Datei: Entweder die zuvor mit "PARAMETER" → "SPEICHERN IN DATEI" gesicherten Steuerungsparameter, wenn Sie neue Kurven editieren wollen. Oder eine cnf-Datei, die zu den Steuerungsparametern schon eine oder mehrere Kurven enthält, die sie bearbeiten wollen.

Wählen Sie im folgenden Dialogfeld die gewünschte Datei mit Doppelklick oder mit Klicken auf OK aus. Wenn die ausgewählte cnf-Datei mehr als eine Kurve enthält, folgt ein weiteres Dialogfeld, in dem Sie die zu bearbeitende Kurve auswählen:



Klicken Sie auf → "NEUE ANHÄNGEN", wenn Sie keine dieser Kurven bearbeiten, sondern eine weitere edieren wollen.

Enthält die ausgewählte cnf-Datei keine Kurve, wird eine leere Datei geöffnet.

Wird eine ältere Version einer cnf-Datei geladen (in der nicht alle neuesten Parameter definiert sind), ergänzt der CAM Editor diese mit geeigneten Default-Werten.

Ein Dialogfenster informiert den Anwender, wenn dies der Fall ist. Wird diese cnf-Datei wieder im CAM Editor gespeichert, den wird die Datei den kompletten Satz der Parameter enthalten.

"SICHERN CNF" und "SICHERN CNF ALS .."

Klicken Sie auf "DATEI → SICHERN CNF", um eine neue oder geänderte cnf-Datei zwischendurch bzw. spätestens vor dem Schließen zu speichern.

Falls Sie eine neue cnf-Datei noch nie gespeichert haben, wird automatisch das entsprechende Dialogfenster "DATEI" → "SICHERN CNF ALS.." angeboten.

Klicken Sie auf "DATEI" → "SICHERN CNF ALS..", um eine neue oder geänderte cnf-Datei unter einem neuen Namen zu speichern. Alle diese Dateien werden automatisch mit der Extension .cnf versehen.

"EXPORT"

Wenn Sie die Daten in ein anderes Format exportieren wollen, zum Beispiel um sie weiterzuverarbeiten oder auszudrucken, benutzen Sie → "EXPORT". Die Daten werden im ASCII-Format mit der Extension .dat gesichert.

"IMPORT"

Importieren können Sie nur dat-Dateien, die Sie mit → "EXPORT" erstellt haben.

"DRUCKEN" und "DRUCKEN SW"

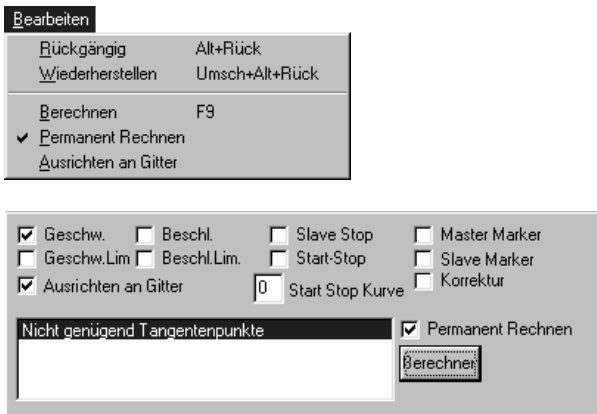
Je nach angeschlossenem Drucker können Sie die Kurvengrafik ausdrucken: Schwarzweiß oder farbig. Falls Ihr Drucker die Farbgrafik nicht in Schwarzweiß konvertieren kann, nutzen Sie → "DRUCKEN SW" für schwarzweiß.

"ENDE"

Den CAM-Editor schließen Sie mit → "ENDE" oder durch Klicken auf das Schließen-Schaltssymbol.

■ Menü Kurve "BEARBEITEN"

Im Menü "BEARBEITEN" finden Sie Funktionen, die zur Berechnung der Kurve erforderlich sind, und soweit sie häufig benötigt werden, auch direkt über einen Button im Editor gestartet oder in Kontrollkästchen ausgewählt werden können.



"PERMANENT RECHNEN"

Bei allen interaktiven und vielen anderen Aktionen wird die Kurve sofort neu berechnet und dargestellt. Schalten Sie → "PERMANENT RECHNEN" aus, wenn das interaktive Nachführen der Kurve beim Verändern der Punkte stört. Sie können dann den Punkt erst mit der Maus versetzen, bevor die Kurve nachgeführt wird.

"AUSRICHTEN AN GITTER"

Wenn die Fixpunkte automatisch auf das Interpolations-Raster gesetzt werden sollen – was empfohlen wird – aktivieren Sie → "AUSRICHTEN AN GITTER". Die Funktion wird nicht ausgeführt, wenn die Masterlänge nicht ganzzahlig durch die Anzahl der Intervalle teilbar ist.

"RÜCKGÄNGIG" und "WIEDERHERSTELLEN"

Wie oft und in welchen Arbeitsschritten Rückgängig wirkt, wird durch das Berechnen bestimmt. Jedes → "RÜCKGÄNGIG" stellt die Situation bis zum vorhergehenden Berechnen her. Sobald Sie die Datei sichern wird der Undo-Speicher gelöscht. Wählen Sie "BEARBEITEN" → "RÜCKGÄNGIG" oder benutzen Sie die Tastenkombination [ALT] + [RÜCKTASTE].

"BEARBEITEN" → "WIEDERHERSTELLEN" oder die Tastenkombination [ALT] + [UMSCHALT] + [RÜCKTASTE] stellt die Arbeitsschritte vor dem Befehl Rückgängig wieder her.

"BERECHNEN"

Bei vielen Aktionen – zum Beispiel allen interaktiven – wird die Kurve sofort neu berechnet und dargestellt. Wenn Sie Felder wie die Fixpunkte oder Parameter ausfüllen, bestimmen Sie mit dieser Funktion den Zeitpunkt selbst.

Wählen Sie "BEARBEITEN" → "BERECHNEN" oder klicken Sie auf den Berechnen-Button oder tasten Sie [F9] um das Berechnen und die neue Darstellung der Kurven auszulösen. Dies ist immer notwendig, wenn Sie zum Beispiel Fixpunkte hinzufügen, Parameter oder Kurven-Daten ändern.

Bei jeder Neuberechnung der Kurve wird auch ein Syntaxcheck durchgeführt. Das kann zu einer Reihe von Fehlermeldungen und Hinweisen führen, die Sie quittieren müssen. Zum Beispiel wenn Ende- und Startpunkt nicht verbunden sind oder nicht genügend Fixpunkte für ein Kurvenstück (mindestens drei) existieren.

■ Menü "ANSICHT"

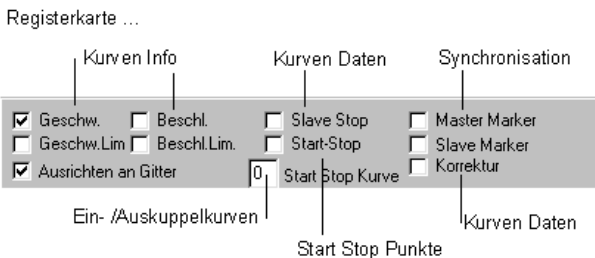
Zusätzlich zu den Fixpunkten können Sie die folgenden Kurven-Informationen und -Parameter grafisch darstellen. Wenn Sie zum Beispiel das CAM-Editor-Fenster so stark verkleinert haben, dass die Kontrollkästchen nicht mehr dargestellt werden, schalten Sie die grafische Darstellung der Parameter im Menü "ANSICHT" ein und aus. Nur die Darstellung der Ein- und Auskuppelkurven können Sie hier nicht aktivieren, da Kurve und Richtung durch eine Eingabe bestimmt werden.

Die ✓-Häkchen zeigen die gerade aktivierten Ansichten.



Die Menü-Auswahl wie auch die Kontrollkästchen sind in Gruppen unterteilt, die den Registerkarten entsprechen.

Oder Sie aktivieren die grafische Darstellung der Parameter durch Anklicken der Kontrollkästchen bzw. bei der Ein- und Auskuppelkurve durch die Eingabe der Punktepaar-Nummer.



Informieren Sie sich über die Bedeutung und die grafische Darstellung der Parameter in den Registerkarten "KURVEN-INFO", "KURVEN-DATEN", "SYNCHRONISATION" und "START-STOP-PUNKTE".

■ Menü "FENSTER" (CAM-Editor)



Da manche Registerkarten durch ihre Größe das CAM-Fenster temporär verändern, können Sie mit Klicken auf → "STANDARD" wieder das Standard-Fenster darstellen. Das Gleiche erreichen Sie auch mit → "BERECHNEN" oder wenn Sie in ein anderes Eingabefeld außerhalb der „großen“ Registerkarte klicken.

■ Kurvenprofil

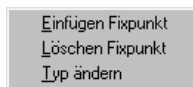
Im Kurvenprofil sehen Sie die grafische Darstellung der Kurve, Parameter und anderer Informationen und haben die Möglichkeit, interaktiv mit der Maus die Fixpunkte zu verändern, zu ergänzen und die Darstellung zu vergrößern. In blauer Schrift sehen Sie den Kurvennamen und den Datei-Namen der gerade dargestellten Kurven.

Fixpunkte mit der Maus einfügen oder löschen

Bevor Sie die Fixpunkte interaktiv mit der Maus bearbeiten können, müssen Sie mindestens zwei Fixpunkte mit Master- und Slave-Koordinaten in der Tabelle definiert haben. Dargestellt wird die Kurve aber nur, wenn mindestens drei Punkte definiert sind und die Kurve auch berechnet werden kann. Sie können Punkte direkt auf der Kurve einfügen oder an beliebiger Position im Koordinatenfeld.

Punkte auf die Kurve zwischen zwei bereits vorhandenen Punkten setzen:

Bewegen Sie den Mauscursor auf einen Fixpunkt bis das Handsymbol erscheint und klicken Sie auf die rechte Maustaste. Wählen Sie im folgenden Popup-Menü die gewünschte Aktion aus:



Mit → "EINFÜGEN FIXPUNKT" wird zwischen dem ausgewählten und dem vorherigen Punkt (nach links) ein weiterer Fixpunkt gesetzt und sofort auch in der Tabelle ergänzt. Vor dem ersten Punkt können Sie keinen weiteren mit der Maus einfügen.

Mit → "LÖSCHEN FIXPUNKT" können Sie den ausgewählten Punkt löschen; dieser wird auch in der Tabelle sofort gelöscht. Den ersten und letzten Fixpunkt können Sie nicht löschen.

Punkte beliebig in das Koordinatenfeld setzen:

Klicken Sie mit der rechten Maustaste an die gewünschte Position und wählen Sie im Popup-Menü → "EINFÜGEN FIXPUNKT".

Fixpunkte interaktiv mit der Maus verschieben

Bewegen Sie den Mauscursor auf einen Fixpunkt bis das Handsymbol erscheint, klicken Sie auf die linke Maustaste und ziehen Sie den Punkt auf die gewünschte neue Position. Die Kurve wird – wenn Permanent Rechnen aktiviert ist – simultan nachgeführt. Wenn dies auf Ihrem Rechner stört, deaktivieren Sie Permanent Rechnen; die Kurve wird dann neu berechnet und dargestellt, sobald Sie die Maustaste loslassen.

Die Fixpunkte sollten auf Intervallgrenzen liegen. Andernfalls ist nicht gewährleistet, dass die Kurve wirklich durch die Fixpunkte geht. Aktivieren Sie daher möglichst immer → "AUSRICHTEN AN GITTER". Vorausgesetzt, es ist bereits eine sinnvolle → "ANZAHL INTERVALLE" definiert.

Punkttyp in der Grafik ändern

Es gibt Kurvenpunkte (Typ 1, grün dargestellt) und Tangentenpunkte (Typ 2, blau dargestellt). Bewegen Sie den Mauscursor auf einen Fixpunkt bis das Handsymbol erscheint und klicken Sie auf die rechte Maustaste. Wählen Sie im folgenden Pop-up-Menü die gewünschte Aktion aus:

Einfügen Fixpunkt
Löschen Fixpunkt
Typ ändern

Mit → "TYP ÄNDERN" wird der ausgewählte und der vorherige Punkt (nach links) geändert, entweder von Kurven- zu Tangentenpunkten oder umgekehrt. Die Änderungen werden sofort auch in der Tabelle (Registerkarte Fixpunkte) ausgeführt.

Kurvenprofil vergrößern und verkleinern

Zum Vergrößern der Grafik klicken und halten Sie die linke Maustaste und ziehen ein Rechteck nach rechts unten. Die Grafik wird entsprechend dem Ausschnitt vergrößert. Sie können dies mehrmals bis zur extrem feinen Wiedergabe wiederholen. Zum Verkleinern klicken und halten Sie die linke Maustaste und ziehen ein Rechteck nach links oben. Die Grafik wird wieder in der Standardgröße dargestellt.

Kurvenprofil scrollen

Klicken und halten Sie die rechte Maustaste und bewegen Sie die Maus in die gewünschte Richtung. Um die Grafik wieder in der Standardposition darzustellen, klicken und halten Sie die linke Maustaste und ziehen ein Rechteck nach links oben. Ein evtl. Vergrößerung wird dabei auch zurückgesetzt.

Was tun ... wenn die Kurve nicht dargestellt wird

Wenn im Grafikenster nur Punkte statt einer Kurve dargestellt werden, konnte die Kurve nicht berechnet werden. Zum Beispiel haben Sie

- zu wenig Punkte eingegeben (mindestens drei sind für die Darstellung erforderlich),
- eine größere Anzahl Intervalle als Masterlänge gewählt,
- Tangentenpunkte falsch gesetzt. (Faustregel: zwei Tangentenpunkte müssen immer mit mindestens zwei Kurvenpunkten abwechseln.)

■ Meldungen des CAM-Editors

Das Display zeigt Eingabe- und Syntaxfehler. Sie können die Meldungen vor und zurück scrollen.



■ Registerkarte "FIXPUNKTE"

Geben Sie in die Kurventabelle die Fixpunkte für den Master und den Slave grundsätzlich aufsteigend ein. Wählen Sie nicht zu kleine Benutzereinheiten, damit die Kurve trotz der ganzzahligen Eingabe eine vernünftige Auflösung erhält. Empfehlenswert sind mindestens vierstellige Einheiten.

Fix Punkte		Start Stop Punkte		
Punkt	Master	Slave	Typ	
1	0	0	1	
2	1500	1200	1	
3	2500	2400	1	
4	4000	3600	1	
5			1	
6			1	
7			1	

Klicken Sie auf → "BERECHNEN" zur Darstellung der Kurve.

Es wird erst dann eine Kurve dargestellt, wenn Sie mindestens für Master und Slave je drei Punkte und den Punkttyp „1“ eingegeben haben.

Sobald Sie mindestens den Anfangs- und Endpunkt der Kurve definiert haben, können Sie die weiteren Fixpunkte auch interaktiv mit der Maus setzen.

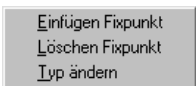
Die Fixpunkte sollten auf Intervallgrenzen liegen.

Aktivieren Sie daher möglichst immer → "AUSRICHTEN AN GITTER".

Fixpunkte in der Tabelle ergänzen oder löschen

In der Tabelle können Sie immer den letzten Punkt löschen oder einen weiteren ergänzen. Innerhalb der Tabelle können Sie mit Hilfe des Popup-Menüs Punkte ergänzen oder löschen.

Sie klicken mit der rechten Maustaste auf den Punkt, den Sie löschen oder vor dem Sie einen Fixpunkt einfügen wollen. Wählen Sie im darauf folgenden Popup-Menü → "EINFÜGEN FIXPUNKT". Der Punkt wird zwischen diesem und dem vorhergehenden eingefügt.



Oder Sie ergänzen oder löschen die gewünschten Punkte interaktiv mit der Maus in der Grafik. Innerhalb der Tabelle können Sie die gesetzten Punkte beliebig durch Überschreiben ändern.

"TYP": Kurven- und Tangentenpunkt

Typ 1 = Kurvenpunkt

Typ 2 = Tangentenpunkt für Geradestücke

Zwischen Kurvenpunkte wird die Kurve als Spline-Interpolation berechnet. Für Bereiche, in denen die Geschwindigkeit konstant und die Beschleunigung 0 sein muss, benutzen Sie Tangentenpunkte. Zwischen diesen Punkten wird statt eines Splines eine Gerade gelegt.

Zur Berechnung eines Splines werden immer drei Punkte benötigt, also mindestens zwei Fixpunkte und eine Tangente (Geradestück) aus zwei Tangentenpunkten.

Bestimmen Sie den Punkttyp entweder in der Tabelle durch numerische Eingabe oder ändern Sie den vorhandenen Typ mit Hilfe des Popup-Menüs. Sie klicken mit der rechten Maustaste auf den Punkt, von dem die Tangente (Geradestück) zum vorhergehenden führen soll. Wählen Sie im darauf folgenden Popup-Menü → "TYP ÄNDERN". Der Typ wird für beide Fixpunkte auf Tangentenpunkt = 2 geändert. Das Gleiche können Sie in der Grafik ausführen: Sie bewegen den Cursor auf den Punkt bis das Hand-symbol erscheint. Sie klicken auf die rechte Maustaste und wählen im Popup-Menü → "TYP ÄNDERN". Auch hier werden sofort beide Punkte geändert.

Registerkarte "START STOP PUNKTE"

In dieser Tabelle definieren Sie Punktepaare für das Ein- und Auskuppeln des Slaves bei der Synchronisation. Ein Punktepaar benötigen Sie, um zu bestimmen, an welcher Master-Position die Synchronisation starten und wo aufsynchronisiert sein soll. Mit einem weiteren Punktepaar legen Sie fest, ab welchem Punkt ausgekuppelt und wo die Synchronisation angehalten werden soll.

Punkt	Master
1 a	1000
1 b	1500
2 a	2500
2 b	3000
3 a	
3 b	

Sie können mehrere (max. 25) Punktepaare definieren, um zum Beispiel für mehrere Starts und Stopps in einem Zyklus verschiedene Situationen beim Starten zu berücksichtigen. Mit den Befehlen `SYNCCSTART pnum` und `SYNCCSTOP pnum slavepos` bestimmen Sie in Ihrem Programm, welches Punktepaar benutzt werden soll.

Wenn A- und B-Punkt identisch sind, wird der Slave mit der eingestellten Maximalgeschwindigkeit – also ohne Kurve – eingekuppelt, sobald der Master diesen Punkt erreicht hat.

Wenn keine Start-Stop-Punkte definiert sind, wird der Slave bei `SYNCCSTART` sofort mit der eingestellten Maximalgeschwindigkeit eingekuppelt.

Auch diese Punktepaare werden immer aufsteigend eingegeben. Denn die Fahrtrichtung wird automatisch durch das Programm berücksichtigt:

Beim Vorwärtsfahren startet die Synchronisation am A-Punkt und wird bis zum B-Punkt beendet. Beim Rückwärtsfahren wird sie am B-Punkt gestartet und bis zum A-Punkt beendet.

Wenn das Programm ohne expliziten Befehl `SYNCCSTOP pnum slavepos` verlassen wird, wird immer das zweite Punktepaar für das Auskuppeln benutzt.

Start-Stop-Punkte und Verlauf grafisch darstellen

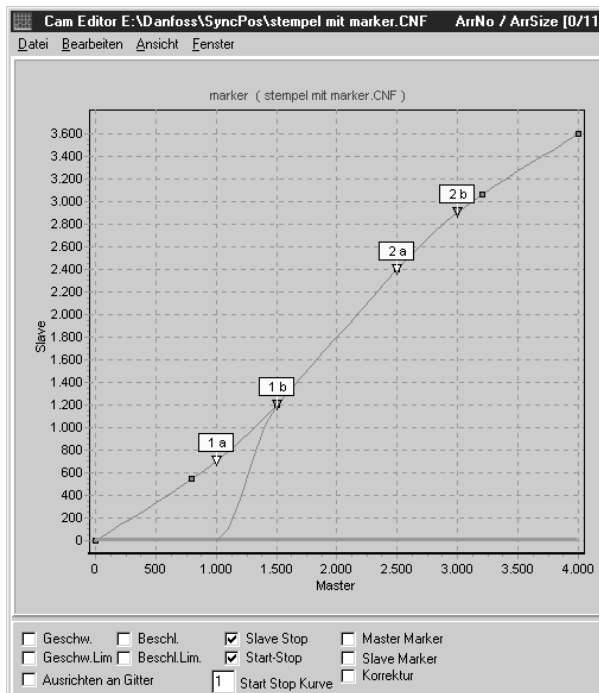
Sie können die Start-Stop-Punkte und die Ein- und Auskuppelkurven im Kurvenprofil visualisieren. Aktivieren Sie → "START-STOP". Gelbe Flags zeigen die Punktepaare für das Ein- und Auskuppeln bei der Synchronisation.

Für die Darstellung der Ein- und Auskuppelkurven geben Sie in → "START-STOP-KURVE" die Nummer des Punktepaars ein, dessen Verlauf Sie sehen wollen. Ohne Vorzeichen zeigt das Einkuppeln, wenn der Master vorwärts fährt. Nummern mit Minus-Vorzeichen zeigen das Auskuppeln, wenn der Master vorwärts fährt.

Wenn der Master rückwärts fährt, gelten die Vorzeichen für den umgekehrten Fall, also Plus für Auskuppeln und Minus für Einkuppeln.

■ Registerkarten "KURVEN-DATEN", "KURVEN-INFO" und Parameter

Bevor Sie eine Kurve editieren, sollten Sie immer zuerst die Parameter Ihrer Steuerung in den CAM-Editor laden. Mit "PARAMETER" → "SPEICHERN IN DATEI" sichern Sie die Benutzerparameter inklusive der Arrays in eine Datei mit der Extension „.cnf“. Diese "DATEI" → "LADEN" Sie in den CAM-Editor. Wenn Sie keine Parameter laden, finden Sie die Werkseinstellungen des VLT eingetragen. Wenn Sie im Verlauf der Kurverstellung die Parameter ändern, werden diese mit in der cnf-Datei gespeichert und mit "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI" in die Steuerung geladen und in den entsprechenden Dialogfeldern der Achsparameter eingetragen.



■ Registerkarte "KURVEN-DATEN"

In der Registerkarte → "KURVEN-DATEN" bestimmen Sie wichtige Eckdaten Ihrer Kurve:

Kurvenname

Wenn Sie mehrere Kurven editieren, geben Sie zu Ihrer eigenen Information hier den Kurven aussagefähige Namen. Wenn eine cnf-Datei mehrere Kurven enthält, werden beim "DATEI" → "LADEN CNF" diese Namen zur Auswahl angeboten.

Kurventyp

Um Drehzahlsprünge bei mehrmaligem Kurvendurchlauf zu verhindern, können Sie zwischen zwei Kurventypen wählen. In Beiden berücksichtigt die Interpolation die Steigung der Kurve am Anfang und Ende:

Wählen Sie den Kurventyp aus:

0 = Die Steigung der Kurve am Anfang und Ende wird gemittelt.

1 = Die Steigung am Anfang der Kurve wird auch für das Ende der Kurve benutzt.

Anzahl Intervalle

Mit der Anzahl der Intervalle definiert man die Anzahl der Geradestücke, die für die gesamte Kurve verwendet werden.

Wählen Sie nicht zu kleine Intervalle (das würde nur einen unnötigen Overhead verursachen) und möglichst einen ganzzahligen Teiler der Masterlänge. Zum Beispiel bei einer Masterlänge von 3000 ein Intervall von 30 oder 60. Dann können die Fixpunkte bei eingeschaltetem → "AUSRICHTEN AN GITTER" auch ohne Fehler darauf gesetzt werden.

Slave-Stop-Position

Bestimmen Sie die Position, zu der der Slave fahren und stoppen soll, wenn im Programm kein SYNCCSTOP Befehl mit der Variablen *slavepos* gesetzt wurde.

Diese Position wird auch verwendet, wenn SYNCC mit einer bestimmten Anzahl Zyklen startet und keinen SYNCCSTOP Befehl benutzt.

Eine graue Linie zeigt diese Position im Kurvenprofil. Aktivieren Sie dazu → "SLAVE-STOP".

Korrektur Start und Korrektur Ende

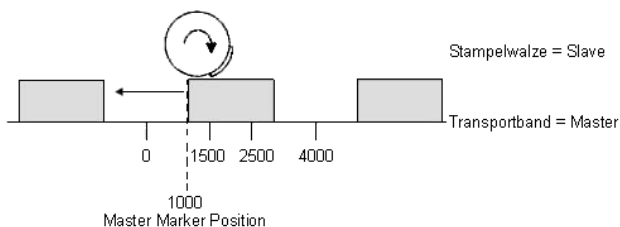
Geben Sie die Master-Positionen ein, bei welcher die Markerkorrektur beginnen und bei welcher sie enden soll. Achten Sie darauf, dass genügend Zeit bleibt, die Synchronisation zu korrigieren, bevor der Bearbeitungspunkt erreicht wird.

Der Korrekturbereich wird im Kurvenprofil blau dargestellt.

Aktivieren Sie dazu → "KORREKTUR".

Master-Marker Position und Slave-Marker Position

Tragen Sie die Master-Position (bzw. bei einer Slave-Synchronisation mit Marker die Slave-Position) ein, für die der Marker eingerichtet wurde, hier zum Beispiel der Anfang eines Kartons:



Aus der Master-Markerposition und dem Markerabstand wird die Position der Kurve errechnet, bei der der Marker erkannt wird. Diese Position wird im Kurvenprofil als grüne Linie dargestellt und ermöglicht Ihnen die Festlegung des Korrekturbereiches. Aktivieren Sie dazu → "MASTER-MARKER" bzw. "SLAVE-MARKER".

Master Länge / Slave Länge

Info über die in der Tabelle der Fixpunkte festgelegte Zykluslänge des Masters bzw. Slave.



ACHTUNG!

Die Slave Länge muss positiv sein; dies kann mit der Definition der positiven Drehrichtung im Parameter POSDRCT (28) sichergestellt werden.

Registerkarte "KURVEN-INFO"

In dieser Registerkarte bestimmen Sie im Eingabefeld die Anzahl der → "ZYKLEN / MIN MASTER". In den anderen Feldern finden Sie Kurven-Informationen, die sich aus den Parametern und der Kurvenanwendung berechnen.

Kurven Daten	Kurven Info	Encoder	Hor
Zyklen / min Master	60		
Max aktuelle Geschw.	1,200		
Max Slave Geschw.	2,250		
Max aktuelle Beschl.	0,001		
Max Slave Beschl.	0,001		
Intervall Grösse	100,000		
Intervall Dauer (ms)	25,000		

Zyklen / min Master

Geben Sie die Anzahl der Zyklen des Masters pro Minute ein. In den meisten Fällen ist dies die Anzahl der Produkte, die (maximal) pro Minute verarbeitet werden.

Max. aktuelle Geschwindigkeit

Der Wert gibt die maximale Geschwindigkeit des Slaves in dieser Kurvenanwendung an.

Den gesamten Geschwindigkeitsverlauf können Sie im Kurvenprofil grafisch darstellen. Aktivieren Sie dazu → "GESCHWINDIGKEIT".

Die blaue Kurve zeigt den Geschwindigkeitsverlauf des Slaves in dieser Kurvenanwendung. Auf der rechten Diagrammachse können Sie die Werte ablesen, und zwar in Einheiten, die die Änderung des Slaves zum Master beschreiben: Δ BE/MU

Max. Slave Geschwindigkeit und Geschwindigkeits-Limit

Dies ist die maximale Geschwindigkeit, die der Slave – in Abhängigkeit von der Geschwindigkeit des Masters – erreichen kann. Je schneller der Master fährt, bzw. je mehr Zyklen/min. er fährt, desto geringer wird das Geschwindigkeits-Limit. Übersteigt der Geschwindigkeitsverlauf dieses Limit, kann der Slave dem Master nicht folgen.

Sie können das Geschwindigkeits-Limit auch im Kurvenprofil grafisch darstellen. Aktivieren Sie → "GESCHW.-LIMIT".

Max. aktuelle Beschleunigung

Der Wert gibt die maximale Beschleunigung des Slaves in dieser Kurvenanwendung an, und zwar in Einheiten, die die Änderung des Slaves zum Master beschreiben: Δ BE/MU.

Den gesamten Verlauf der Beschleunigung können Sie im Kurvenprofil grafisch darstellen. Aktivieren Sie → "BESCHLEUNIGUNG". Die gelbe Linie zeigt den Beschleunigungsverlauf des Slaves.

Normalerweise ist die Beschleunigung im Vergleich zur Geschwindigkeit relativ klein. Daher empfiehlt es sich, die Darstellung der Geschwindigkeit und des Geschwindigkeits-Limits auszuschalten, um in der Grafik eine gute Repräsentation der Beschleunigung zu sehen.

Max. Slave Beschleunigung und Beschleunigungs-Limit

Dies ist die maximale Beschleunigung, die der Slave – in Abhängigkeit von der Beschleunigung des Masters – erreichen kann. Je schneller der Master fährt, bzw. je mehr Zyklen/min. er fährt, desto geringer wird das Beschleunigungs-Limit. Übersteigt der Beschleunigungsverlauf dieses Limit, kann der Slave dem Master nicht folgen.

Sie können das Beschleunigungs-Limit auch im Kurvenprofil grafisch darstellen. Aktivieren Sie → "BESCHL.-LIMIT".

Intervall-Größe

Die Intervall-Größe ergibt sich aus der Anzahl der Intervalle pro Masterzykluslänge.

Intervall-Dauer (ms)

Die Zeit in (ms) für ein Intervall ergibt sich ebenfalls aus der Anzahl der Intervalle pro Masterzykluslänge. Sie sollte nicht kleiner als 30 ms sein. (30 bis 100 ms sind geeignete Werte). Verändern Sie also in den "KURVEN-DATEN" die → "ANZAHL INTERVALLE", damit Sie eine vernünftige Größe erhalten.

■ Registerkarten Encoder, Homefahrt, Ein-/Ausgänge und Regelung

Informieren Sie sich im Kapitel [Parameter-Referenz](#) über die Inhalte, Einheiten, Eingabebereiche und Werkseinstellung dieser Parameter oder stellen Sie den Cursor in das Eingabefeld und tasten [F1].

■ Registerkarte Synchronisation

Informieren Sie sich im Kapitel [Parameter-Referenz](#) über die Inhalte, Einheiten, Eingabebereiche und Werkseinstellung dieser Parameter oder stellen Sie den Cursor in das Eingabefeld und tasten [F1].

Parameter	Wert	Einheit	Parameter	Wert	Einheit
Syncfaktor Master	2048	[49]	Master Marker		
Syncfaktor Slave	55	[50]	Type	2	[60]
Positionsoffset	0	[54]	Anzahl	1	[52]
Max. Geschw. (%)	0	[66]	Abstand	2600	[58]
Sync. filterzeit	0	[65]	Toleranz	200	[68]
Genauigkeit	1000	[55]	Slave Marker		
Ready (Anz. Marker)	1	[56]	Typ	0	[61]
Fehler (Anz. Marker)	10	[57]	Anzahl	1	[53]
Startverhalten	0	[62]	Abstand	0	[59]
			Toleranz	0	[69]

Für die Kurvenscheibensteuerung benötigen Sie folgende Parameter:

Syncfactor Master und Slave

Die beiden Parameter SYNCFACTM [49] und SYNCFACTS [50] werden benutzt, um bei einer Kurvenscheibensteuerung die MU-Einheiten zu bestimmen.

Master Marker Abstand und Slave Marker Abstand

Tragen Sie hier den Abstand des Sensors zum Bearbeitungspunkt ein; bei Master-Marker in SYNCPUISM [58] und bei Slave-Marker in SYNCPUISS [59].

Aus der Master-Markerposition und dem Markerabstand wird die Position der Kurve errechnet, bei der der Marker erkannt wird. Diese Position wird im Kurvenprofil als grüne Linie dargestellt und ermöglicht Ihnen die Festlegung des Korrekturbereiches. Aktivieren Sie dazu → "MASTER-MARKER" bzw. "SLAVE-MARKER".

Master-Marker Toleranz und Slave-Marker Toleranz

Toleranzfenster für das Auftreten der Master-Marker SYNCMWINM [68] bzw. der Slave-Marker SYNCMWINM [69]. Das Toleranzfenster wird im Kurvenprofil als grüner Bereich dargestellt. Aktivieren Sie dazu → "MASTER-MARKER" bzw. "SLAVE-MARKER".

■ Registerkarte **Geschwindigkeit**

Informieren Sie sich im Kapitel [Parameter-Referenz](#) über die Inhalt, Einheiten, Eingabebereiche und Werkseinstellung dieser Parameter oder stellen Sie den Cursor in das Eingabefeld und tasten [F1].

Parameter	Wert	Einheit	Standardwert
Max. Geschw. (1/min)	8000	[1]	Standard
Geschw. Teiler	100	[22]	Geschwindigkeit
Rampentyp	0	[32]	
Kürzeste Rampe (ms)	500	[31]	
Max. Geschw. (qc/st)	2184533	[9]	
Max. Beschl. (qc/st)	2184	[10]	

Außerdem wird hier auch die in der aktuellen Anwendung erreichte maximale Geschwindigkeit und Beschleunigung in qc/Abtastzeit berechnet. Die Darstellung im Kurvenprofil erfolgt in Einheiten. Aktivieren Sie dazu → "GESCHWINDIGKEIT" bzw. "BESCHLEUNIGUNG".

■ Menü "EINSTELLUNGEN"



In diesem Menü können Sie die Compiler Optionen und die Einstellungen der Schnittstelle ändern und die Farben des Editors nach Ihren Wünschen einstellen. Wie Sie eine andere Sprache auswählen, lesen Sie im Anschluss.

■ "COMPILER"

Die Default-Werte für die Compiler Optionen sind für die meisten Anwendungen passend gesetzt. So belegen sie einerseits nicht zu viel Speicherplatz und ermöglichen andererseits die erforderlichen Eingaben.

Max. Anzahl Variablen

Die Anzahl der Variablen hat direkten Einfluss auf die Menge des Speicherplatzes, der in der Option zur Verfügung steht. Dabei ist zu beachten, dass ein Array zusätzlich den Platz einer Variablen belegt. Wenn Sie mehr als 92 Variablen (inkl. Arrays) benötigen erhöhen Sie die Anzahl.

Max. Anzahl Labels

Die Maximale Anzahl der Labels bestimmt, wie viel Speicherplatz während der Compilierung für interne Sprungmarken zur Verfügung gestellt wird. Interne Sprungmarken werden für alle Programmverzweigungen (GOTO, IF, LOOP, REPEAT, WHILE, GOSUB) während der Compilierung automatisch erzeugt. Der empfohlene Einstellbereich liegt zwischen 100 und 500 interner Labels.

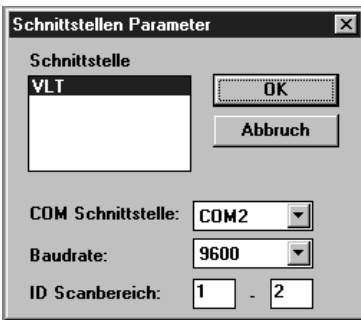
Erhöhen Sie die maximal zulässigen Werte, wenn die Anzahl der Labels zum Beispiel für Texteingaben nicht genügt.

■ "SCHNITTSTELLE"

Die Schnittstelle haben Sie schon bei der Inbetriebnahme des VLT festgelegt.

Wenn Sie eine der Einstellungen ändern müssen, zum Beispiel die Baudrate, klicken Sie auf "EINSTELLUNGEN" → "SCHNITTSTELLE" und geben die Änderung ein.

Für die VLT-Schnittstelle stehen die beiden Baudraten 9600 und 19200 zur Verfügung.



ACHTUNG!

Beachten Sie aber, dass bei einer Änderung der Schnittstellen-Einstellungen die Kommunikation zwischen PC und VLT verloren gehen kann.

ID-Scanbereich

Legen Sie den Bereich fest, in welchem die im Netz angeschlossenen VLTs gesucht werden sollen. Wenn Sie einen oder mehrere VLTs hinzufügen, müssen Sie den Bereich entsprechend der vergebenen Nummern erweitern.

RS485 Anschluss

Für einen RS485 Anschluss benötigen Sie eine RS232-Standardschnittstelle oder eine zusätzliche RS232-Schnittstellen-Karte im PC und einen externen Wandler.


■ "FARBEN EDITOR"

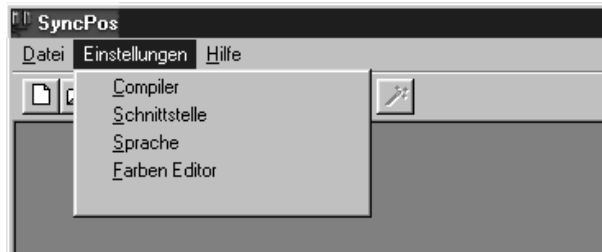
Zur besseren Übersicht können den verschiedenen Programmteilen wie Kommentar, Schlüsselwort, Ziffer usw. unterschiedliche Farben zugeordnet werden. Dazu öffnen Sie im Menü "EINSTELLUNGEN" den "FARBEN EDITOR". Markieren Sie den Typ, z.B. Kommentar und wählen Sie dazu die gewünschte Farbe.

Mit OK speichern Sie die neuen Einstellungen.



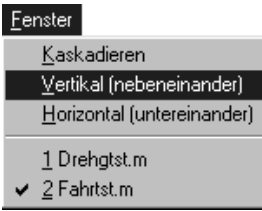
■ "SPRACHE"

Wenn Sie das Menü "EINSTELLUNGEN" aufrufen wollen um die Sprache zu ändern, müssen alle Dialogfelder und das Editierfenster geschlossen sein. Dazu klicken Sie auf das Symbol Schließen  in der rechten oberen Ecke des geöffneten Editierfensters bzw. Dialogfeldes.



Wenn Sie eine andere Sprache wünschen, klicken Sie auf "EINSTELLUNGEN" → "SPRACHE" und wählen im darauf folgenden Dialogfeld zwischen Englisch und Deutsch. Danach müssen Sie SyncPos schließen und neu starten, damit die Änderung wirksam wird.

■ Menü "FENSTER"



Sie können mehrere Editierfenster bzw. Dateien öffnen und jedes Editierfenster mit einem VLT verbinden.

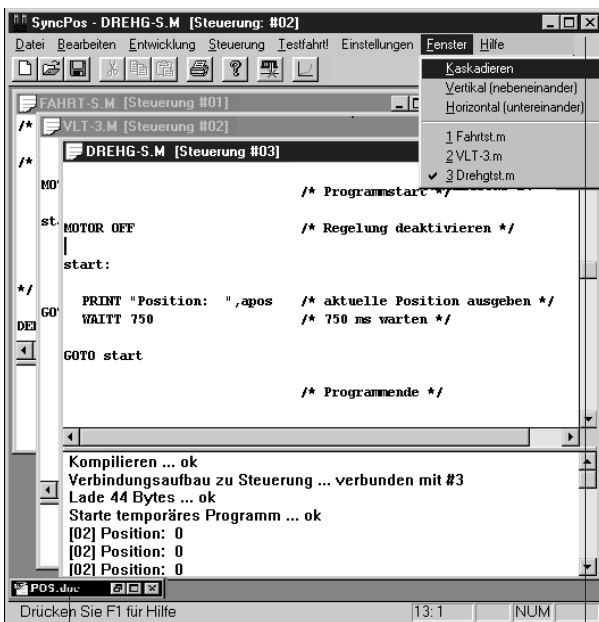
Für die Darstellung von mehreren Editierfenstern wählen Sie im Menü "FENSTER" zwischen → "KASKADIEREN", "VERTIKAL" (nebeneinander) oder "HORIZONTAL" (untereinander).

Klicken Sie auf das Vollbildsymbol, wenn Sie die Fenster weder kaskadieren noch nebeneinander darstellen wollen.

Wenn Sie Dateien als Symbole darstellen, werden sie im Allgemeinen links unten im SyncPos-Fenster angeordnet.

■ "KASKADIEREN"

Alle geöffneten Dateien sind im Menü "FENSTER" aufgelistet. Klicken Sie diejenige Datei an, die beim Kaskadieren als oberste dargestellt werden soll. Die Dateien werden leicht versetzt übereinander dargestellt:



Dateisymbol

Vollbildsymbol

■ "VERTIKAL" (nebeneinander)

Wollen Sie mehrere Programme gleichzeitig beobachten, können Sie verschiedene Fensterdarstellung wählen. Zum Beispiel das SyncPos-Fenster teilen und die Dateien nebeneinander zeigen.

■ "HORIZONTAL" (untereinander)

Das SyncPos-Fenster wird geteilt und die Dateien untereinander dargestellt.

■ Menü "HILFE"


Die Darstellung und Funktionalität der Online-Hilfe unterscheidet sich in Abhängigkeit vom eingesetzten Betriebssystem.

■ "INHALT"

Klicken Sie auf "HILFE" → "INHALT". Wenn Sie zum Beispiel Details zum Befehl "ACC" suchen, wählen Sie im Inhaltsverzeichnis das Thema „Alle SyncPos-Befehle von ACC bis #INCLUDE“ und dort den Befehl ACC aus.

Volltextsuche

Für die Volltextsuche klicken Sie zuerst auf das Register "SUCHEN", damit die Wortliste erstellt wird. Dann geben Sie den Suchbegriff ein und Sie erhalten alle Textstellen, in denen der Begriff vorkommt zur Auswahl. Markieren Sie eine Wortentsprechung und wählen Sie dann unter den angebotenen Themen das gewünschte aus.

Suchen über Index

Oder geben Sie im Register Index „acc“ ein und wählen den passenden Indexeintrag aus.

Kontextsensitive Hilfe

Die "BEFEHLSHILFE", alle Parameter-Dialogfelder im Menü "STEUERUNG" sowie alle Registerkarten im CAM-Editor bieten einen direkten Zugang zur Online-Hilfe. Markieren Sie den Befehl in der "BEFEHLSHILFE" bzw. stellen Sie den Mauscursor in das Eingabefeld eines Parameters und drücken Sie [F1]. Sie erhalten dann den entsprechenden Abschnitt der Hilfe direkt angezeigt.

Verweise

Verweise auf andere Textstellen sind im Handbuch unterstrichen und in der Online-Hilfe zusätzlich farbig markiert, zum Beispiel POSA. Klicken Sie darauf und das Thema wird angezeigt. Klicken Sie auf "ZURÜCK" in der Menüleiste der Hilfe, wenn Sie das vorhergehende Thema weiter lesen wollen. Markierungen mit punktiertem Unterstrich blenden Bilder, Grafiken und Begriffe aus dem Glossar ein, wenn Sie darauf klicken, zum Beispiel MLONG. Klicken Sie neben dieses Popup oder tasten Sie [ESC] um das Popup wieder auszublenden.

Hilfetexte ausdrucken oder kopieren

Sie können die Hilfetexte ausdrucken oder mit "BEARBEITEN" → "KOPIEREN" in Ihr Editierfenster "EINFÜGEN", zum Beispiel fertige Programmstrings aus den Programmbeispielen.

■ "PROGRAMMINFO"

Hier finden Sie die Versionsnummern des SyncPos-Programmes, der Programmbibliothek und des Compilers.

Kapitel 5 a

■ PC-Software Benutzeroberfläche	
im MTC10 Modus	89
■ Betriebsmodus: Offline oder Online	89
■ SyncPos im MCT10-Modus starten	89
■ SyncPos im MCT10-Modus beenden	
Beenden durch den Anwender	89
Automatisches Programmende	89
■ Dateien-Handling	89
■ Das SyncPos-Fenster im MCT10 Mode	89
MCT10 Betriebsmodus: Online	89
MCT10 Betriebsmodus: Offline	89
Menü "STEUERUNG"	89
Menü "TESTFAHRT"	89
Menü "CAM-EDITOR"	89
■ Menü "DATEI" im MCT10-Modus.....	90
"DATEI" → "NEU"	90
"DATEI" → "ÖFFNEN"	90
"DATEI" → "SPEICHERN ALS"	90
"DATEI" → "DRUCKEN" und "DRUCKER	
EINRICHTEN"	90
"DATEI" → "LETZTE DATEI"	90
■ Menü "ENTWICKLUNG" im MCT10-Modus	90
Online	90
Offline	90
"ENTWICKLUNG" → "SYNTAXPRÜFUNG"	90
"ENTWICKLUNG" → "BEFEHLSHILFE"	90
■ CAM-Editor im MCT10-Modus	91
■ CAM-Editor-Fenster ohne Kurvendaten	91
Menü "DATEI" (CAM-Editor) im MCT10-Modus	91
Menü "BEARBEITEN" (CAM-Editor) im	
MCT10-Modus.....	92
Menü "ANSICHT" (CAM-Editor) im MCT10-Modus...	92
■ Menü "EINSTELLUNGEN" im MCT10-Modus.....	92

SyncPos verhält sich anders, wenn es nicht direkt durch Klicken auf die SyncPos Programm-Schaltfläche, sondern indirekt durch Öffnen einer Programm- (*.m) oder einer Konfigurationdatei (*.cnf) über das **Motion Control Tool MCT10** gestartet wird. Der Hauptunterschied besteht darin, dass das gesamte Dateien-Handling ausschließlich durch das MCT10 stattfindet.

Das folgende Kapitel beschreibt die Unterschiede in der SyncPos Benutzeroberfläche im MCT10-Modus, zum Beispiel wie man Dateien öffnet und sichert oder wie man eine CAM-Kurve editiert. Darüber hinaus unterscheidet sich die Benutzeroberfläche auch in Abhängigkeit vom Betriebsmodus des MCT10: Offline oder Online Modus.

■ Betriebsmodus: Offline oder Online

SyncPos wird durch MCT10 im Offline- oder Online-Modus gestartet, je nachdem ob das MCT10 eine Verbindung zu einem Antrieb hat oder nicht. Die Auswahl des Betriebsmodus Offline oder Online sowie die Interface-Einstellungen erfolgen durch das MCT10 beim Starten von SyncPos automatisch und können nicht geändert werden, solange SyncPos läuft.

■ SyncPos im MCT10-Modus starten

Im MCT10-Modus können Sie SyncPos nicht wie gewohnt über "START" → "PROGRAMME" → "SYNCPOS" starten, sondern es wird automatisch geöffnet, wenn eine Programm- (*.m) oder Konfigurationsdatei (*.cnf) über das MCT10 ausgewählt wurde. MCT10 startet SyncPos, überträgt den ausgewählten Dateinamen und öffnet diese Datei. Es ist möglich, über die MCT10 Struktur und Dateiauswahl mehrere SyncPos-Fenster mit Programmen zu öffnen. MCT10 konfiguriert auch automatisch die Sprache und die Interface-Einstellungen von SyncPos.

■ SyncPos im MCT10-Modus beenden

Beenden durch den Anwender

Klicken Sie auf "DATEI" → "PROGRAMMENDE" oder auf die Schließen-Schaltfläche.

Automatisches Programmende

SyncPos wird automatisch geschlossen wenn sowohl der Programm-Editor als auch eine eventuell ebenfalls geöffnete Konfigurationsdatei im CAM-Editor durch den Anwender geschlossen wird. Der Grund hierfür ist, dass jedes Dateien-Handling durch das MCT10 ausgeführt werden muss und es keine Möglichkeit gibt, eine neue oder existierende Datei in SyncPos zu öffnen, wenn es im MCT10-Modus betrieben wird. Daher ist es auch nicht notwendig, SyncPos aktiviert zu lassen, wenn alle Dateien geschlossen sind.

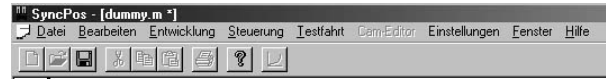
■ Dateien-Handling

MCT10 ist verantwortlich für die Handhabung der SyncPos Programmdateien (*.m) sowie der SyncPos Konfigurationsdateien (*.cnf). Das heißt, dass Dateien nur über MCT10 erzeugt, geöffnet, umbenannt oder gedruckt werden können. Es gibt keine Dateiauswahl in SyncPos, solange es im MCT10-Modus läuft. Es wird in SyncPos automatisch die Datei geöffnet, die der Anwender über das MCT10 ausgewählt hat. Aufgrund der Tatsache, dass das Dateien-Handling ausschließlich dem MCT10 vorbehalten ist, sind einige Menüs und Funktionen von SyncPos abgeblendet, das heißt nicht verfügbar.

■ Das SyncPos-Fenster im MCT10 Modus

MCT10 Betriebsmodus: Online

Das Menü "CAM-EDITOR" in der Titelleiste und die meisten Schaltflächen in der Symbolleiste sind abgeblendet, also nicht verfügbar:



Der "CAM-EDITOR" kann zwar nicht geöffnet werden, wenn SyncPos im MCT10-Modus betrieben wird, aber er wird immer dann automatisch geöffnet, wenn eine Konfigurationsdatei (*.cnf) über das MCT10 ausgewählt wurde oder das Programm eine Konfigurationsdatei enthält.

Die Schaltflächen "NEUE DATEI", "DATEI ÖFFNEN", "DRUCKEN" und der → "CAM-EDITOR" sind ebenfalls abgeblendet. Lesen Sie in der folgenden Beschreibung, wie Sie eine neue Datei erstellen, eine Datei öffnen oder Dateien speichern.

MCT10 Betriebsmodus: Offline

Im Betriebsmodus Offline können alle Funktionen, die einen Anschluss zum Antrieb erfordert, nicht genutzt werden. Hier sind weitere Menüs abgeblendet:



Menü "STEUERUNG"

Es gibt keine Verbindung zum Antrieb. Daher kann dieser auch durch SyncPos weder konfiguriert noch gesteuert werden.

Menü "TESTFAHRT"

Es gibt keine Verbindung zum Antrieb. Daher gibt es keine Möglichkeit, eine Testfahrt durchzuführen.

Menü "CAM-EDITOR"

Der CAM-Editor kann zwar nicht geöffnet werden, wenn SyncPos im MCT10-Modus betrieben wird, aber er wird immer dann automatisch geöffnet, wenn eine Konfigurationsdatei (*.cnf) über das MCT10 ausgewählt wurde oder das Programm eine Konfigurationsdatei enthält.

■ Menü "DATEI" im MCT10-Modus

Da das gesamte Dateien-Handling ausschließlich durch das MCT10 vorgenommen wird, sind folgende Menüpunkte abgeblendet:



"DATEI" → "NEU"

Erzeugen Sie neue Dateien über das MCT10.

"DATEI" → "ÖFFNEN"

Wählen Sie die Datei über das MCT10 aus. SyncPos und damit die Datei wird dann automatisch geöffnet.

"DATEI" → "SPEICHERN ALS"

Wenn Sie eine Programmdatei (*.m) umbenennen oder kopieren wollen, benutzen Sie die Möglichkeiten des MCT10. In SyncPos können Sie das aktuelle Programm bzw. Änderungen jedoch unter dem gleichen Dateinamen → "SPEICHERN".

"DATEI" → "DRUCKEN" und "DRUCKER EINRICHTEN"

Programmdateien (*.m) können Sie über das MCT10 ausdrucken. Der Drucker wird dann natürlich auch über das MCT10 eingerichtet.

"DATEI" → "LETZTE DATEI"

Die letzte(n) Dateien werden nicht zur Auswahl angeboten, da man diese nicht direkt öffnen kann, solange SyncPos im MCT10-Modus läuft.

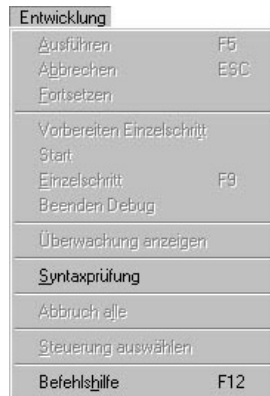
■ Menü "ENTWICKLUNG" im MCT10-Modus

Online

Im Online Modus gibt es keinen Unterschied zur SyncPos-Benutzeroberfläche; alle Funktionen können genutzt werden.

Offline

Im Offline Mode können keine Funktionen genutzt werden, die einen Zugang zum Antrieb erfordern. Daher sind die meisten Menüpunkte abgeblendet:



Folgende zwei Menüpunkte können Sie aber auch im MCT10 Offline-Modus nutzen:

"ENTWICKLUNG" → "SYNTAXPRÜFUNG"

Damit können Sie die Syntax eines Programmes im SyncPos-Editierfenster prüfen.

"ENTWICKLUNG" → "BEFEHLSHILFE"

Die Befehlshilfe bietet eine etwas begrenzte Funktionalität: Es kann nur genutzt werden, um Befehl im Editierfenster einzufügen oder die kontextsensitive Hilfe für die Befehle aufzurufen.

Es können aber keine Befehle direkt ausgeführt oder Positionen angefahren werden.

■ CAM-Editor im MCT10-Modus

Wie die SyncPos-Programmdateien können auch Konfigurationsdateien (*.cnf) nur über das MCT10 geöffnet werden. MCT10 gibt SyncPos beim Starten an, welche Dateien geöffnet werden sollen.

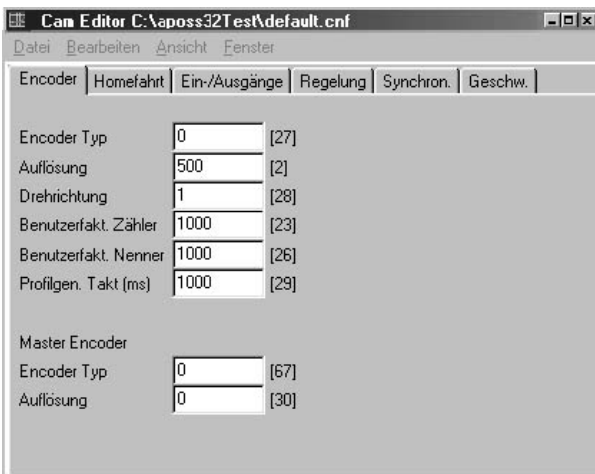
Falls eine Konfigurationsdatei (*.cnf) mit Kurvendaten dabei ist, wird automatisch der CAM-Editor geöffnet und Sie können die in der Datei enthaltenen Parameter und ein eventuell vorhandenes Kurvenprofil mit allen Funktionen interaktiv bearbeiten.

Enthält die Konfigurationsdatei keine Kurvendaten, wird ein stark reduzierter CAM-Editor mit den Registerkarten der Parameter geöffnet und es besteht nachträglich Bedarf, können Sie ein → "CAM-PROFIL EINFÜGEN".

Falls keine Konfigurationsdatei beim Start von SyncPos angegeben ist, wird der CAM-Editor nicht geöffnet und es besteht für den Anwender auch nachträglich keine Möglichkeit diesen doch noch zu starten oder eine Datei für den CAM-Editor zu öffnen.

■ CAM-Editor-Fenster ohne Kurvendaten

Auch wenn die Konfigurationsdatei keine Kurvendaten enthält, können Sie die in der Datei abgelegten Parameter mit dem reduzierten CAM-Editor ändern. Das modifizierte CAM-Editor-Fenster zeigt dann statt der vier Bereiche nur die Registerkarten der Parameter:



Menü "DATEI" (CAM-Editor) im MCT10-Modus

Da das gesamte Dateien-Handling ausschließlich durch das MCT10 durchgeführt wird, sind viele Menüpunkte permanent oder spezifisch abgeblendet:



Abgeblendete Menüpunkte

"DATEI" → "NEUE CNF"

Erzeugen Sie neue Konfigurationsdateien über das MCT10.

"DATEI" → "LADEN CNF"

Laden Sie die Datei über das MCT10; diese wird dann automatisch beim Öffnen von SyncPos geladen.

Falls die automatisch geöffnete Konfigurationsdatei keine Kurve beinhaltet, ist das Einfügen einer Kurve mit "BEARBEITEN" → "CAM-PROFIL EINFÜGEN" möglich.

"DATEI" → "SICHERN CNF ALS"

Wenn Sie eine Konfigurationsdatei (*.m) umbenennen oder kopieren wollen, benutzen Sie die Möglichkeiten des MCT10. In SyncPos können Sie die aktuelle Datei bzw. Änderungen jedoch unter dem gleichen Dateinamen → "SICHERN".

"DATEI" → "EXPORT" bzw. → "IMPORT"

Kurvendaten können Sie nur über das MCT10 exportieren bzw. importieren.

Spezifische Menüpunkte

"DATEI" → "DRUCKEN" und → "DRUCKER SW"

Die in einer Konfigurationsdatei hinterlegten Kurven können grafisch ausgedruckt werden, da hierfür von MCT10 aus keine Möglichkeit besteht. Falls die Konfigurationsdatei keine Kurven beinhaltet, sind diese Menüpunkte ebenfalls abgeblendet.

"DATEI" → "ENDE"

Wenn Sie eine Konfigurationsdatei manuell mit "DATEI" → "ENDE" oder durch Klicken auf die Schließen-Schaltfläche beenden, können Sie diese nicht wieder öffnen, da Konfigurationsdateien nur automatisch via Auswahl im MCT10 geöffnet werden.

Falls zudem keine Programmdatei mehr geöffnet ist, wird SyncPos ebenfalls geschlossen.

Menü "BEARBEITEN" (CAM-Editor) im MCT10-Modus

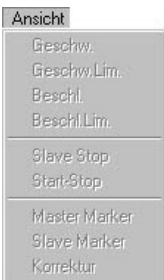
Bei Konfigurationsdateien, die keine Kurve beinhalten, ist das Einfügen einer Kurve mit "BEARBEITEN" → "CAM-PROFIL" einfügen möglich.



Es wird ein leeres Kurvenprofil erstellt und der "CAM-EDITOR" mit allen vier Bereichen geöffnet, um die Kurvenpunkte und Kurvendaten zu definieren und die Kurve interaktiv bearbeiten.

Menü "ANSICHT" (CAM-Editor) im MCT10-Modus

Alle Funktionen sind abgeblendet, wenn eine Konfigurationsdatei bearbeitet wird, die keine Kurve beinhaltet.



■ **Menü "EINSTELLUNGEN" im MCT10-Modus**

Das Menü ist reduziert auf die Funktionen "EINSTELLUNGEN" → "COMPILER" und → "FARBEN EDITOR".



Schnittstelle einstellen und Sprache auswählen

Alle Interface-Einstellungen und die Auswahl der Sprache werden beim automatischen Starten von SyncPos durch das MCT10 vorgenommen. Sie können in diesem Modus nicht in SyncPos verändert werden.

Kapitel 6**■ Programmieren mit SyncPos**

Grundsätzlicher Programmaufbau	95
Grundsätzlicher Befehlsaufbau	95
Interrupt	96
Sprachelemente	98
Arithmetik	99
Benutzereinheiten	101

■ Software-Referenz

Alle Befehle in der Übersicht.....	102
Alle Befehle von ACC bis #INCLUDE	105

■ Parameter-Referenz

VLT-Parameter und SyncPos-Parameter	172
VLT-Parameter	172
VLT Parameter Liste.....	176
Alle SyncPos-Parameter in alphabetischer Übersicht	177
Alle SyncPos-Parameter im Detail.....	179

Wie Sie mit SyncPos programmieren ist in folgenden Abschnitten beschrieben. Anfänger lesen bitte die grundsätzlichen Erläuterungen zur Programmiersprache SyncPos, also Programmaufbau, Befehlsaufbau, Interrupt, Sprachelemente, Arithmetik und Benutzereinheit. Erfahrene Programmierer informieren sich bitte über die SyncPos-spezifischen Grundlagen, zum Beispiel Benutzereinheiten oder Parameter.

Alle Befehle sind in der Software-Referenz zunächst in einer Übersicht und anschließend alphabetisch geordnet ausführlich beschrieben und mit kurzen Beispielen ergänzt. Knapp 50 Programmbeispiele finden Sie zum Nachvollziehen in der Online-Hilfe.

Und in der Parameter-Referenz sind nach der grundsätzlichen Beschreibung der Parameter, diese ebenfalls zuerst in einer Übersicht und im Anschluss daran im Detail beschrieben.

■ Programmieren mit SyncPos
■ Grundsätzlicher Programmaufbau

- Definitionen: Arrays, Interrupts, Benutzerparameter
- Initialisierung: Parameter, Flags und Variablen setzen
- Hauptprogrammschleife


```
main:
  -
  GOTO main
```
- Unterprogrammbereich


```
SUBMAINPROG
  SUBPROG name
  -
  RETURN
ENDPROG
```

Definitionen

Array	DIM send[12], receive[12]
Interrupt	ON ERROR GOSUB errhandle ON INT -1 GOSUB stopprog ON PERIOD 500 GOSUB calc ON TIME 10000 GOSUB break
Benutzerparameter	LINKXPAR SYNCACCURACY 710 "ACCURACY [qc]" 0 100000 0 LINKGPAR 133 716 "Offset [qc]" 0 100000 0

Initialisierung

Parameter	SET POSERR 100000000 SET 133 10000 SETVLT 205 50
Flags/Variablen	offset = 0 sync_flag = 0
Systemparameter	VEL 100 ACC 100 DEC 100

Hauptprogramm

```
main:
IF (IN 3 == 1) THEN
/* Synchronisationsmodus, wenn Eingang 3 = 1 */
  GOSUB syncprog
ELSE
  GOSUB speedprog
/* Drehzahlmodus, wenn Eingang 3 nicht 1 */
GOTO main
```

Unterprogramm

```
SUBMAINPROG:
  SUBPROG syncprog
    IF (sync_flag == 0) THEN
      /* synchronisieren, falls nicht schon aktiv */
      SYNC
      sync_flag = 1
    ENDIF
  RETURN
  SUBPROG errhandle
    WAITI 18 on
    /* warte auf Eingang 18, Fehler löschen */
    sync_flag = 0
    ERRCLR
  RETURN
ENDPROG
```

Sequentielle Befehlsabarbeitung

Generell wird ein Befehl vollständig abgearbeitet, bevor ein neuer begonnen wird. Das führt dazu, dass bei Positionierbefehlen gewartet wird, bis die Zielposition erreicht ist.
Ausnahme: Wenn NOWAIT ON gesetzt ist.

■ Grundsätzlicher Befehlsaufbau

Alle Anweisungen bestehen aus: BEFEHLSWORT + eventuell **Parameter**.
Statt einer absoluten Zahl kann für einen Parameter auch eine Variable stehen:

Beispiel
POSA 10000
oder
pos = 10000
POSA pos

Befehlsausführungszeiten

Falls die Befehlsausführungszeit in einer Anwendung kritisch sein sollte, können Sie die Ausführungszeiten einer Befehlssequenz mit Hilfe des Befehls TIME unter den verschiedenen Betriebsbedingungen messen.

Tips zur Erhöhung der Programmlesbarkeit

- Verwenden Sie Groß- und Kleinschreibung, z.B. alle Befehle groß, Variablen klein.
- Fügen Sie Leerzeichen zwischen den Befehls-teilen ein.
- Kommentieren Sie Ihr Programm. Die Kom-mentare stehen zwischen
/* Kommentar */ oder nach //Kommentar
Unzulässig ist die aber Schachtelung von Kom-mentaren (/* ... /*...*/ ... */).
/* Beginn KOMMENTAR Ende */
// Beginn KOMMENTAR Ende
- Verwenden Sie Zeileneinzüge innerhalb von Schleifen.

Werteeingaben

Wie in anderen Programmiersprachen auch werden Werteeingaben nicht geprüft. Es liegt also in der Ver-antwortung des Programmierers, dass zu extreme Werte zu Problemen führen. Bei der Suche nach solchen Problemen können Sie den Debug-Modus nutzen.

■ Interrupt

Generell gibt es drei Sorten von Interrupts

ON INT Interrupt bei Flanken eines Eingangs
ON PERIOD / ON TIME

 Interrupt nach Ablauf einer Zeit-
spanne

ON COMBIT / ON STATBIT

 Interrupt wenn Bit n gesetzt wird

ON PARAM Interrupt wenn sich ein Benutzer-
Parameter n ändert

Generelle Abarbeitung von Interrupt Prozeduren

Nach jedem internen SyncPos-Befehl wird abgefragt, ob ein Interrupt-Ereignis vorliegt. Dabei ist zu be-achten, dass mit jedem internen SyncPos-Befehl der Compiler eine Anweisung im SyncPos-Maschinen-code erzeugt.

So wird zum Beispiel eine einfache Anweisung wie:
POSA (*ziel* + 1000)

in folgenden SyncPos-Maschinencode zerlegt:

```
MOVE ziel nach Register 101
MOVE Immediate 1000 nach Register 102
ADDREG Register 102 plus Register 101 nach
Register 101
POSA Achse nach Register 101
```

Außerdem wird bei länger dauernden Befehlen (wie DELAY oder WAITAX) ständig geprüft, ob ein Inter-rupt-Ereignis aufgetreten ist. In diesem Fall wird der Befehl unterbrochen und nach Abarbeitung des Interrupts wieder fortgesetzt.


Achtung!

Verwenden Sie nicht WAITT in Verbindung mit Interrupts, da dabei der Wartevorgang nach der Unterbrechung erneut von vorn beginnt.

Benutzung von Variablen innerhalb von Interrupt Pro-zeduren

Das obige Beispiel mit dem "SyncPos-Maschinen-code" zeigt auch deutlich, dass bei der Zuweisung von Variablen innerhalb von Interrupt Prozeduren mit größter Sorgfalt vorgegangen werden muss.

Wird zum Beispiel im Hauptprogramm eine Zuwei-sung der Art

```
ziel = ziel + wert - 1000
```

vorgenommen, wird diese in eine Folge von SyncPos-Maschinencode-Befehlen zerlegt, wobei die Zwischenergebnisse in temporären Registern gespeichert werden. Erst am Ende der Folge wird das Ergebnis nach *ziel* zurückgespeichert.

Wird nun während der Ausführung dieser Befehle ein Interrupt ausgelöst und in der entsprechenden Prozedur ein Befehl

```
ziel = 0
```

ausgeführt, wird es in diesem Fall Probleme geben. Denn nach der Abarbeitung der Interruptprozedur wird in das Hauptprogramm zurückgesprungen und dann das immer noch vorhandene Zwischenergeb-nis nach *ziel* gespeichert: Somit wird die 0 in *ziel* wieder überschrieben.

ON PERIOD

Bei ON PERIOD Funktionen wird dagegen beim Start einer solchen Funktion die Zeit berechnet, wann der nächste Aufruf erfolgen soll, also

```
START_TIME = TIME + PERIOD.
```

Sobald diese Zeit erreicht ist, wird die Funktion aus-geführt und anschließend die nächste Startzeit berechnet mit der Formel

```
START_TIME = START_TIME + PERIOD.
```

Dies sorgt dafür, dass die Abstände des Aufrufens wirklich gleich sind, da die Ausführungszeit die Berechnung nicht beeinflusst. Dies bedeutet aber auch, dass der Anwender darauf achten muss, dass die Periode wirklich länger als die Ausführungszeit ist, da sonst ein "Stau" entsteht, das heißt es würde eigentlich nur noch die ON PERIOD Funktion aus-geführt.

Reaktionszeiten

Das Vorhandensein eines Interrupts wird in einer speziellen Funktion geprüft, die auch zur Watch-Dog-Überwachung verwendet wird. Deshalb wird diese generell in jeder Prozedur die etwas länger dauern könnte und in allen Schleifen etc. aufgerufen. In dieser Prozedur wird immer nach 1 ms geprüft, ob ein solches Ereignis vorliegt und gegebenenfalls ein entsprechendes Flag gesetzt. Dieses Flag wird dann spätestens nach Abarbeitung des gerade aktuellen APOS-Maschinencodes erkannt und ausgewertet. Die Reaktionszeit ist die maximale Ausführungszeit eines Maschinencodes oder 1 ms, je nachdem was größer ist: Beim VLT ist dies ca. 1 ms. Eine Ausnahme bildet der Zeit-Interrupt (ON TIME / ON PERIOD). Hier wird nur alle 20 ms geprüft ob die Zeit abgelaufen ist. Daher macht es auch keinen Sinn, eine ON PERIOD mit weniger als 20 ms zu definieren.



ACHTUNG!

Außerdem ist generell darauf zu achten, dass Interrupt-Funktionen nicht zu lange dauern.

Vor allem bei ON PERIOD Funktionen ist dringend darauf zu achten, dass die Funktion nicht länger dauert als die Periode, weil sonst ein Stau von Funktionsaufrufen entsteht.

Prioritäten

Falls zwei Interrupt Ereignisse gleichzeitig auftreten sollten, wird die Abarbeitung wie folgt priorisiert:

- ON INT geht vor
- ON APOS, ON MAPOS, ON MCPOS vor
- ON COMBIT vor
- ON STATBIT vor
- ON PARAM vor
- ON TIME / ON PERIOD,

wobei die anderen Ereignisse aber nicht verloren gehen.

Innerhalb der einzelnen Typen gilt wiederum Folgendes:

ON INT / ON COMBIT / ON STATBIT

Sollten zwei (Eingangs)-Interrupts gleichzeitig kommen, wird der mit der niedrigeren Nummer zuerst ausgeführt, wobei der andere aber nicht verloren geht. Die anderen werden dann nach Beendigung der Interrupt Prozedur entsprechend aufgerufen. Sollte derselbe Eingang bzw. Interrupt während der Ausführung der Prozedur noch einmal kommen, wird auch dieser wieder vermerkt und anschließend ausgeführt.

Ein Interrupt kann also nur verlorengehen, wenn er während der Ausführung einer Interrupt Prozedur zweimal kommt.

ON TIME / PERIOD

Wie bereits oben beschrieben, wird in einer internen Struktur für jede Zeitfunktion die nächste Ausführungszeit vermerkt. Bei gleichzeitiger Ausführungszeit wird die Prozedur zuerst ausgeführt, die zuerst in der Liste steht. Die Priorität ergibt sich also aus der Reihenfolge der ON PERIOD Befehle.

ON PARAM

Wenn mehrere dieser Interrupts gleichzeitig auftreten, werden sie in der Reihenfolge der ON PARAM Befehle im Programm abgearbeitet.

Interrupt Schachtelung

Es ist nicht möglich, dass ein Interrupt von einem anderen Interrupt unterbrochen wird. Während ein Interrupt behandelt wird, kann demnach kein zweiter behandelt werden. Einzige Ausnahme ist die ON ERROR Funktion, die auch während der Abarbeitung von Interrupts möglich ist.

Eine ON ERROR Funktion kann aber von keinem Interrupt unterbrochen werden.

NOWAIT in Interrupts

Generell ist während eines Interrupts NOWAIT auf ON gesetzt, das heißt dass nicht auf die Beendigung von POSA Befehlen gewartet wird.

Dies ist nötig, da sonst ein POSA Befehl von einer Interrupt-Prozedur nicht unterbrochen werden kann, da sofort auf das Ankommen der Achsen gewartet werden würde. Will man also innerhalb einer Interrupt-Prozedur auf das Ankommen einer Achse warten, muss man dies explizit mit WAITAX tun.

■ Sprachelemente
Konstanten

Überall dort, wo Parameter oder Werte erwartet werden, können Konstanten stehen, die typischerweise in Ganzzahlenwerte eingegeben werden, zum Beispiel: wert = 5000

Der Vorteil von Konstanten ist, dass sie keinen eigenen Speicherplatz belegen.

Konstanten ...

- ... sind Ganzzahlenwerte im Bereich von -2 bis +2 Mrd.,
- ... gelten innerhalb des gesamten Programms (sie sind global),
- ... können dezimal, hexadezimal (0x + Hexadezimalzahl), oktäl (0 + Oktälzahl) oder in ASCII (zwischen Apostroph) eingegeben werden, zum Beispiel
 wert = 5000 = Dezimal 5000
 wert = 0x7F = Dezimal 127
 wert = 0100 = Dezimal 64
 wert = 'A' = Dezimal 65
 Besonders die Hexadezimal- und ASCII-Eingaben vermeiden manche Umrechnung und das Programm wird lesbarer, z.B. key = 'A'

Variablen

- ... können zur Zwischenspeicherung von Abfrage- und Rechenergebnissen verwendet werden,
- ... entstehen durch die Zuweisung eines Wertes,
- ... müssen nicht separat definiert werden,
- ... gelten innerhalb des gesamten Programms (sie sind global),
- ... enthalten Ganzzahlenwerte im Bereich von -2 bis +2 Mrd.,
- ... können innerhalb von Befehlen statt fester Werte verwendet werden,
- ... müssen vor der Verwendung in einem Befehl einen Wert zugewiesen bekommen.

Variablennamen

- ... können beliebig lang sein,
- ... können aus Buchstaben, Ziffern und dem Unterstrich bestehen,
- ... dürfen keine länderspezifischen Zeichen, wie Umlaute enthalten,
- ... müssen mit einem Buchstaben beginnen,
- ... können groß oder klein geschrieben werden (keine Unterscheidung!),
- ... dürfen nicht mit einem Befehlsnamen identisch sein.

Spezielle Variablen

ERRNO = Systemvariable, die die aktuelle Fehlernummer enthält.

Arrays

Die Programmierung von Programmen mit Dialog erfordert die Speicherung von Benutzereingaben oder Positionen über längere Zeit, also auch nach dem Ausschalten des VLTs. Meistens sind dies mehrere Werte, die am Besten in Feldern bzw. Arrays abgelegt werden.

Die Arrays werden im Speicherbereich der Benutzerprogramme abgelegt und sind global definiert, das heißt unabhängig vom aktuellen Programm. Der Benutzer kann selbst festlegen, wie viele Arrays er definiert und wie groß die einzelnen Arrays sein sollen. Die Festlegung erfolgt durch die Anweisung DIM und ist danach fest und kann nicht mehr geändert werden (außer durch Speicher löschen). In jedem Programm, das Arrays benutzen soll, muss eine entsprechende DIM Anweisung stehen, die mit der ursprünglichen Definition übereinstimmt. Andernfalls wird ein Fehler gemeldet.

DIM Anweisung

DIM muss die erste Anweisung in einem Programm sein und noch vor dem Unterprogramm-bereich erscheinen.

Die DIM Anweisung vereinbart die später verwendbaren Arrays. Sollten bis dahin noch keine Arrays angelegt gewesen sein, werden sie neu angelegt. Waren bereits Arrays definiert, müssen die Angaben mit der ursprünglichen Definition übereinstimmen.

Beispiel

```
DIM ziel1[20], ziel2[20], ziel3[20], werkoffset[50]
DIM parameter[10]
```

Mit diesen Befehlen werden insgesamt 5 Arrays mit den entsprechenden Größen definiert. Wenn dieses Programm einmal ausgeführt wurde, sind die obigen Arrays in der SyncPos-Option angelegt. Wird bei einem erneuten Start eines Programmes festgestellt, dass die Array-Definition von den Arrays in der SyncPos-Option abweicht, wird dies als Fehler angezeigt. Allerdings ist es korrekt, wenn ein zweites Programm nur folgende Zeile enthält:

Beispiel

```
DIM ziel1[20], ziel2[20], ziel3[20]
```

Die Reihenfolge der Definition muss aber immer gleich sein, da die SyncPos-Option nicht die Namen der Arrays speichert, sondern nur deren Position in der DIM Anweisung. So ist auch die folgende Programmzeile korrekt und das Array xpos ist dann identisch mit dem Array ziel1.

Beispiel

```
DIM xpos[20], ypos[20], zpos[20], offs[50]
```

Indizes

Die Elemente eines Arrays werden über einen entsprechenden Index in eckigen Klammern bezeichnet: xpos[5]. Dabei sind Indizes von 1 bis zur Größe des definierten Arrays erlaubt. Also im obigen Fall bei xpos von 1 bis 20. Der Versuch auf Elemente davor oder dahinter zuzugreifen, führt zu einer Fehlermeldung, da dies zu Datenverlust und Zerstörung der Arrays führen könnte.

Arrays schreiben und lesen

Der Zugriff auf die so definierten Arrays erfolgt analog zu der Benutzung von Variablen. So sind alle folgenden Anweisungen korrekt:

Beispiel

```
xpos[1] = 10000
xpos[2] = 20000
xpos[3] = 30000
i = 1
WHILE(i<20) DO
ypos[i] = i*1000
i = i+1
ENDWHILE
zpos[1] = APOS
POSA xpos[1]
offs[1] = (xpos[2]) % 20
```

Arrays versus Variablen

Grundsätzlich können Arrays überall dort verwendet werden, wo auch Variablen zulässig sind. Auch belegt ein Array nur den Platz einer internen Variablen und vermindert somit die Zahl der maximal zulässigen Variablen lediglich um eins. Die maximale Variablenzahl können Sie im Menü "EINSTELLUNGEN" → "COMPILER" einstellen.

■ **Arithmetik**

Der Compiler bietet folgende Befehle und Parameter:

Operatoren: plus, minus, mal, geteilt, XOR, Modulo, Division, Absolutbetrag

Bitoperatoren: und, oder, invert, linksschieben, rechtsschieben, Bit, Byte, Word, Long

Vergleichsoperationen: größer als, kleiner als, größer, gleich als, kleiner, gleich als, gleich wie, ungleich

logische Verknüpfungen: und, oder, nicht

Informieren Sie sich im Anschluss an die Art der Zuweisung (Assignment Operation), die entsprechend den Bit-/Byte-Befehlen aufgebaut ist und über die Prioritäten der Operatoren und Operationen.



ACHTUNG!

Alle Arithmetikoperationen sind Ganzzahloperationen!

Operatoren

Symbol	Bedeutung	Syntax / Beispiel	Beschreibung
+	plus	$3 + 3 = 6$	Addition
-	minus	$9 - 3 = 6$	Subtraktion
*	mal	$2 * 3 = 6$	Multiplikation
%	geteilt	$19 \% 3 = 6$	Division (Ergebnis abgeschnitten)
^	XOR	$\text{expr1} \wedge \text{expr2}$ $127 \wedge 255 = 128$	Exklusiv Oder (binäre Operation)
mod	Modulo	$\text{expr1} \text{ mod } \text{expr2}$ $250 \text{ mod } 16 = 10$	Mathematisches modulo (Rest einer Integerdivision)
rnd	Division	$\text{expr1} \text{ rnd } \text{expr2}$ $250 \text{ rnd } 16 = 16$	Division mit Runden, im Gegensatz zur Division (%) mit Abschneiden
abs	Absolutbetr.	$\text{Abs}(\text{expr})$ $\text{abs}(-5) = 5$	Absolutbetrag des Ausdrucks

Bitoperatoren

Symbol	Bedeutung	Syntax / Beispiel	Beschreibung
&	und	$7 \& 6 = 6$	bitweise Verknüpfung
	oder	$2 4 = 6$	bitweise Verknüpfung
~	invert	$\sim(-7) = 6$	bitweises Invertieren
<<	linksschieben	$3 \ll 1 = 6$	bitweises Linksschieben
>>	rechtsschieben	$12 \gg 1 = 6$	bitweises Rechtsschieben
.	Bit	$\text{expr1}.\text{expr2}$ $7.1 = 1$ $7.3 = 1$ $7.4 = 0$	Liefert das Bit expr2 von expr1 zurück
.b	Byte	$\text{expr1}.\text{b} \text{ expr2}$ $0x027F.\text{b}1 = 127$ $0x027F.\text{b}2 = 2$	Liefert das Byte expr2 von expr1 zurück
.w	Word	$\text{expr1}.\text{w} \text{ expr2}$ $0x0010FFFF.\text{w}2 = 16$	Liefert das Wort expr2 von expr1 zurück
.l	Long	$\text{expr1}.\text{l} \text{ expr2}$	Liefert das Long expr2 von expr1 zurück (standard)

Vergleichsoperationen und logische Verknüpfungen

Vergleichsoperationen	>	größer als
	<	kleiner als
	>=	größer, gleich als
	<=	kleiner, gleich als
	==	gleich wie
	!=	ungleich
Logische Verknüpfungen	AND	und
	OR	oder
	NOT	nicht

Zuweisung (Assignment Operation)

Wert	= 0	Standard Zuweisung zu einer Variablen
Feld[1]	= 0	Standard Zuweisung zu einem Array Wert
Wert.3	= 1	Bit 3 wird auf 1 gesetzt, Wert = 4
Feld[1].8	= 1	Bit 8 wird auf 1 gesetzt, Feld[1] = 128
Wert.b1	= 72	Unterstes Byte von Wert wird auf 72 gesetzt Wert = 72
Wert.b2	= 128	Zweites Byte von Wert wird auf 128 gesetzt Wert = 0x00008048
Wert.w2	= 15	Zweites Wort von Wert wird auf den Wert 15 gesetzt. Wert = 0x000F8048

Priorität der Operatoren und Operationen

Operatoren in derselben Zeile haben die gleiche Priorität, werden also nacheinander abgearbeitet. Die Prioritäten sind in absteigender Folge erläutert:

*	%	multiplikativ
+	-	additiv
>>	<<	bitweises Schieben/shiften
>=	<=	Relation
==	!=	Gleichheit/Equality
&		bitweises und
		bitweises inklusive oder
AND		logisches und
OR		logisches oder

■ Benutzereinheiten

Damit der Anwender mit sinnvollen Maßangaben arbeiten kann, können für den Antrieb, bzw. für den Slave und den Master die Einheiten vom Benutzer beliebig definiert werden.

Benutzereinheit [BE]

Wegangaben in Fahrbefehlen erfolgen immer in Benutzereinheiten und werden intern in Quadcounts umgerechnet. Diese wirken sich auf alle Befehle für das Positionieren aus: z.B. APOS, POS.

Auch für die Kurvenscheibensteuerung kann der Anwender sinnvolle Einheiten wählen, um die Kurve für den Master und den Slave zu beschreiben. Zum Beispiel 1/100 mm oder bei Anwendungen, bei denen eine Umdrehung betrachtet wird 1/10 Grad.

Bei der Kurvenscheibensteuerung wird der maximale Fahrabstand des Slaves bzw. die Slave-Zykluslänge in Benutzereinheiten BE (qc) angegeben.

Sie normieren die Einheit mit einem Faktor. Dieser ist ein Bruch, der sich aus Zähler und Nenner zusammensetzt:

$$1 \text{ BE} = \text{POSFAC T_Z (23)} / \text{POSFAC T_N (26)}$$

Die Normierung bestimmt, wie viele Quadcounts eine Benutzereinheit ergeben: wenn der Faktor zum Beispiel 50375/1000 beträgt, entspricht eine BE genau 50,375 qc.


NB!

Wenn die Benutzereinheiten in qc umgerechnet werden, wird der Integer-Wert benutzt.

Wenn qc in Benutzereinheiten umgerechnet werden, wird gerundet.

Masterunit [MU]

Die Kurvenlänge bzw. Master-Zykluslänge und andere Angaben (zum Beispiel der Markerabstand) für die Kurvenscheibensteuerung werden in Masterunits MU angegeben.

Für die Umrechnung in qc wird wie bei der Benutzereinheit ein Faktor (Bruch) benutzt:

$$1 \text{ MU} = \text{SYNCFAC T_M (49)} / \text{SYNCFAC T_S (50)}$$

■ Software-Referenz
■ Alle Befehle in der Übersicht
Initialisierung der SyncPos-Option (INI)

Befehle zum Initialisieren der Achse und der SyncPos-Option, zum Anfahren und Definieren der/ des Nullpunkte(s).

DEFMORIGIN	Aktuelle Master-Position als Nullpunkt für den Master setzen
DEF ORIGIN	Aktuelle Position als Nullpunkt setzen
DELETE ARRAYS	Alle Arrays im RAM löschen
ERRCLR	Fehlermeldung löschen
HOME	Maschinennullpunkt anfahren
INDEX	Indexposition des Drehgebers anfahren
MOTOR OFF	Motorregelung ausschalten
MOTOR ON	Motorregelung einschalten
ORIGIN	Reallnullpunkt anfahren
RST ORIGIN	Temporärnullpunkt löschen
SETMORIGIN	Beliebige Position als Nullpunkt für den Master setzen
SET ORIGIN	Temporärnullpunkt setzen
SAVE ARRAYS	Arrays im EEPROM sichern
SAVE AXPARS	Aktuelle Achsenparameter im EEPROM sichern
SAVE GLBPARS	Aktuelle globale Parameter im EEPROM sichern
SAVEPROM	Speicher in EEPROM sichern
SWAPMENC	Master- und Slave-Drehgeber intern tauschen

Steuerungsbefehle (CON)

Befehle zur Steuerung des Programmablaufs und zur Strukturierung von Programmen.

DELAY	Zeitverzögerung
DIM	Definition eines Arrays
EXIT	Vorzeitiger Programmabbruch
GOSUB	Aufruf eines Unterprogrammes
GOTO	Sprung zu einem Programmlabel
IF THEN	Bedingte einfache Programmverzweigung
..ELSE ..IF..THEN	Bedingte mehrfache Programmverzweigung
... ELSE	Alternative Programmverzweigung
... ENDIF	Ende der Programmverzweigung
LOOP	Definierte Schleifenwiederholung
CONTINUE	Abgebrochene Positionier- und Drehzahlbefehle fortsetzen, z.B. nach einem MOTOR STOP
MOTOR STOP	Stoppen des Antriebs
NOWAIT ON/OFF	Wartemodus ein-/ausschalten

REPEAT	Bedingte Schleife Anfang
REPEA...UNTIL	Bedingte Schleife Ende
SUBMAINPROG	Beginn der Unterprogrammdefinition
... ENDPROG	Ende der Unterprogrammdefinition
SUBPROG	Beginn eines Unterprogrammes
... RETURN	Ende eines Unterprogrammes
SYSVAR	Systemvariable (Pseudo-Array) liest Systemwerte
WAITAX	Warten bis die Zielposition erreicht ist
WAITI	Warten auf bestimmten Eingangszustand
WAITNDX	Warten auf Index
WAITP	Warten bis Position erreicht
WAITT	Zeitverzögerung
WHILE ... DO	While-Schleife Anfang
... ENDWHILE	While-Schleife Ende
#INCLUDE	Einfügen des Inhalts einer Datei

Absolutbewegungen (ABS)

Befehle zum absoluten Positionieren der Achse.

ACC	Beschleunigung setzen
DEC	Beschleunigung für den Bremsweg setzen
POSA	Positioniere Achsen absolut
VEL	Geschwindigkeit für relative und absolute Bewegungen und max. zulässige Geschwindigkeit für Synchronisationsvorgänge setzen

Drehzahlregelung (DRE)

Befehle zum permanenten Verfahren der Achse.

CSTART	Starten des Drehzahlmodus
CSTOP	Stoppen des Drehzahlmodus
CVEL	Geschwindigkeit Drehzahlregelung setzen

Ein-/Ausgabe-Befehle (I/O)

Befehle zum Setzen und Rücksetzen der Ausgänge, Abfragen der Eingänge, Abfragen von Bewegungsinfos, Abfragen von Systemdaten und zum Ein- und Ausgeben von Benutzerinfos.

APOS	Aktuelle Position lesen
AXEND	Status der Programmausführung abfragen
CPOS	Sollposition lesen
MAPOS	Aktuelle Istposition des Masters abfragen
MIPOS	Letzte Index- bzw. Markerposition des Masters abfragen

IPOS	Letzte Index- bzw. Markerposition des Slaves abfragen	ON ERROR GOSUB	Unterprogramm bei Fehler aufrufen
MAVEL	Aktuelle Geschwindigkeit des Masters abfragen	ON INT .. GOSUB	Unterprogramm bei Flanke eines Einganges aufrufen
AVEL	Aktuelle Geschwindigkeit der Achse abfragen	ON MAPOS .. GOSUB	Unterprogramm aufrufen, wenn die Master-Position xxx (qc) passiert wurde
TRACKERR	Aktuellen Schleppabstand des Slaves abfragen	ON MCPOS .. GOSUB	Unterprogramm aufrufen, wenn die Master-Position xxx (MU) passiert wurde
SYNCERR	Aktuellen Synchronisationsfehler einer Achse abfragen	ON PARAM .. GOSUB	Unterprogramm aufrufen, wenn sich ein Benutzer-Parameter ändert
ERRNO	Fehlernummer lesen	ON PERIOD GOSUB	Unterprogramm in regelmäßigen Zeitabständen aufrufen
IN	Eingänge bitweise lesen (einzeln)	ON STATBIT	Unterprogramm aufrufen, wenn Bit n des VLT-Status gesetzt ist
INB	Eingänge byteweise lesen (8 Stück)	ON TIME	Unterprogramm nach einmaligem Zeitablauf aufrufen
INAD	Analogeingänge lesen		
INKEY	Einlesen eines Tastensignals vom VLT		
OUT	Ausgänge bitweise setzen (einzeln)		
OUTAN	VLT-Sollwert setzen		
OUTB	Ausgänge byteweise setzen (8 Stück)		
OUTDA	(frei verfügbaren) Analogausgang setzen		
PID	PID-Berechnung durchführen		
PRINT	Ausgabe von Informationen	Parameter (PAR)	
STAT	Status der Achse lesen		
TESTSETP	Aufzeichnungsdaten für Testfahrt festlegen		
TESTSTART	Aufzeichnung der Testfahrt starten		
TIME	Systemzeit auslesen	GET	SyncPos-Parameterwert lesen (Achsparemeter, globale Parameter und Benutzerparameter)
_GETVEL	Abtastzeit für AVEL und MAVEL ändern	GETVLT	Liest einen VLT-Parameter
		GETVLTSUB	Liest einen VLT-Parameter mit Indexnummer
<u>Interrupt-Funktionen (INT)</u>			
DISABLE ...	Interrupt ... ausschalten	SET	SyncPos-Parameterwert setzen (Achsparemeter, globale Parameter und Benutzerparameter)
DISABLE ALL	Alle Interrupts (außer ON ERROR) ausschalten.	SETVLT	Setzt bzw. verändert einen VLT-Parameter
ENABLE ...	Interrupt ... einschalten	SETVLTSUB	Setzt bzw. verändert einen VLT-Parameter mit Indexnummer
ENABLE ALL	Alle Interrupts einschalten	LINKAXPAR	Achsparemeter mit LCP-Display verknüpfen
ON APOS .. GOSUB	Unterprogramm aufrufen, wenn die Slave-Position xxx passiert wurde	LINKGPAR	Globalen Parameter mit LCP-Display verknüpfen
ON COMBIT .. GOSUB	Unterprogramm aufrufen, wenn Bit n des Kommunikationspuffers gesetzt ist	LINKSYSVAR	Systemvariable mit LCP-Display verknüpfen
ON DELETE .. GOSUB	Löscht einen Positions-Interrupt: ON APOS, ON MCPOS oder ON MAPOS		

<u>Kommunikationsoption</u>		<u>CAM-Mode (CAM)</u>	
COMOPTSEND	Schreibt in den Puffer der Kommunikationsoption	Befehle für die Synchronisation im CAM-Mode (Kurvenscheibensteuerung)	
COMOPTGET	Liest ein Telegramm der Kommunikationsoption	CURVEPOS	Slave-Position, die der aktuellen Master-Position der Kurve entspricht, abfragen.
PCD	Pseudo-Array für den direkten Zugriff auf den Feldbus-Datenbereich	DEFMCPOS	Anfangsposition des Masters definieren
<u>Relativbewegungen (REL)</u>		POSA CURVEPOS	Slave auf die, der Master-Position entsprechenden Kurvenposition fahren.
Befehle zum relativen Positionieren der Achse.		SETCURVE	CAM-Kurve setzen
ACC	Beschleunigung setzen	SYNCC	Synchronisation im CAM-Mode
DEC	Beschleunigung für den Bremsweg setzen	SYNCCMM	Synchronisation im CAM-Mode mit Markerkorrektur des Masters
POSR	Positioniere Achsen relativ	SYNCCMS	Synchronisation im CAM-Mode mit Markerkorrektur des Slaves
VEL	Geschwindigkeit setzen	SYNCCSTART	Slave zur Synchronisation im CAM-Mode starten
<u>Synchronisation (SYN)</u>		SYNCCSTOP	Slave nach der CAM-Synchronisation anhalten
Befehle zum (Auf-)Synchronisieren der Slaves auf den Master, bzw. auf die Mastersimulation.			
DEF SYNCORIGIN	Definiert Master-Slave-Verhältnis für den nächsten SYNCP oder SYNCM Befehl		
MOVESYNCORIGIN	Synchronisationsursprung relativ verschieben		
SYNCV	Geschwindigkeitssynchronisation		
SYNCP	Winkel-/Positionssynchronisation		
SYNCM	Winkel-/Positionssynchronisation mit Markerkorrektur		
SYNCSTAT	Flag für Synchronisationsstatus abfragen		
SYNCSTATCLR	Zurücksetzen der Flags MERR und MHIT		
PULSACC	Beschleunigung für virtuellen Master setzen		
PULSVEL	Geschwindigkeit für virtuellen Master setzen		

■ Alle Befehle von ACC bis #INCLUDE

Alle Befehle sind im folgenden Abschnitt alphabetisch geordnet und ausführlich mit Syntax-Beispielen beschrieben.

■ ACC

Mit dem Befehl ACC bestimmen Sie die Beschleunigung für die nächsten Fahrbefehle im Drehzahl-, Positionier- oder Synchronisationsmodus. Der Wert bleibt so lange gültig, bis mit einem weiteren Befehl ACC eine neue Beschleunigung gesetzt wird. Der Wert bezieht sich auf die Parameter **Kürzeste Rampe** RAMPMIN (31) und **Maximalgeschwindigkeit** VELMAX (1) sowie **Geschwindigkeitsteiler** VELRES (22).



ACHTUNG!

Wenn man mit der SyncPos-Optionskarte arbeitet, sollte man die Rampen immer über die Optionskarte einstellen und nicht im VLT. Die VLT-Rampen müssen dabei immer auf Minimum stehen.

Kurzinfo

Beschleunigung für Bewegungsvorgänge setzen.

Syntax

ACC a

Parameter

a = Beschleunigung



ACHTUNG!

Wurde vor einem Fahrbefehl noch keine Beschleunigung definiert, wird mit dem Defaultwert aus DFLTACC (34) beschleunigt.

Befehlsgruppe

REL, ABS

Querverweise

DEC VEL, POSA, POSR

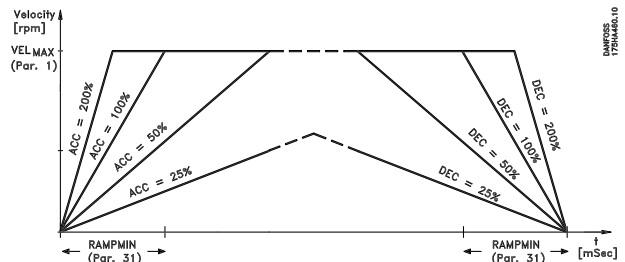
Parameter: RAMPMIN (31), VELMAX (1), VELRES (22)

Syntax-Beispiel

ACC 10 /* Beschleunigung 10 */

Beispiel

kürzeste Rampe:	1000 ms
Maximalgeschwindigkeit:	1500 U/min (25 U/s)
Geschwindigkeitsteiler:	100



Programmbeispiel

ACC_01.M

■ APOS

Der Befehl APOS kann die Position der Achse absolut zum aktuellen Nullpunkt abfragen.

Kurzinfo

Aktuelle Istposition einer Achse abfragen

Syntax

erg = APOS

Rückgabewert

erg = Istposition in Benutzereinheiten (BE) absolut zum aktuellen Nullpunkt

Wegangaben in Fahrbefehlen erfolgen immer in Benutzereinheiten und werden intern in Quadcounts umgerechnet. (Siehe auch **Benutzerfaktor Zähler** und **Nenner**, Parameter 23 und 26.)



ACHTUNG!

Wenn ein mit SET ORIGIN gesetzter und aktiver Temporärnullpunkt existiert, ist der Positionswert auf diesen Nullpunkt bezogen.

Befehlsgruppe

I/O

Querverweise

CPOS, DEF ORIGIN, SET ORIGIN, POSA, POSR

Parameter: POSFACT_Z (23), POSFACT_N (26)

Syntax-Beispiel

PRINT APOS

/* aktuelle Position der Achse am PC ausgeben */

Programmbeispiele

APOS_01.M
GOSUB_01.M, MOTOR_01.M

■ AVEL

Diese Funktion liefert die aktuelle Geschwindigkeit der Achse in Benutzereinheiten pro Sekunde (BE/sec) zurück. Die Genauigkeit der Werte hängt von der Messdauer (Mittelung) ab. Diese ist standardgemäß auf 20 ms eingestellt, kann aber vom Anwender mit dem _GETVEL Befehl verändert werden. Es genügt den Befehl einmal aufzurufen, um von da an mit einer anderen Messzeit zu arbeiten. So stellt der Befehl:

```
var = _GETVEL 100
```

die Messdauer auf 100 ms ein, so dass man bei AVEL und MAVEL eine wesentlich bessere Auflösung der Geschwindigkeit erhält, schnelle Änderungen dagegen erst mit einer Verzögerung von maximal 100 ms.

Kurzinfo

Aktuelle Geschwindigkeit der Achse abfragen

Syntax

erg = AVEL

Rückgabewert

erg = aktuelle Geschwindigkeit der Achse n
in BE/sec; Wert mit Vorzeichen

Befehlsgruppe

I/O

Querverweise

MAVEL, APOS, _GETVEL

Syntax-Beispiel

```
PRINT AVEL
/* aktuelle Geschwindigkeit der Achse am PC
ausgeben */
```

■ AXEND

Der Befehl AXEND liefert den aktuellen Status der Achse bzw. den Stand der Programmausführung. Damit können Sie zum Beispiel abfragen, wann die „Position erreicht“ ist und ein Positionierbefehl (POSA, POSR) wirklich abgeschlossen ist. Wenn Bit 1 auf „0“ gesetzt ist, ist der Positioniervorgang abgeschlossen und die Position erreicht. Wenn aber der Positionierbefehl mit MOTOR STOP unterbrochen wurde und später mit CONTINUE fortgesetzt wird, dann würden folgende Bits auf „1“ gesetzt sein:

- das Bit 0 für „Motor befindet sich im Stillstand“
- das Bit 1 für „Positioniervorgang aktiv“
- das Bit 3 für „Motor ist im STOP-Zustand“
- das Bit 6 für „Lageregelung abgeschaltet“

Der Befehl AXEND eignet sich besonders um im NOWAIT ON Zustand festzustellen, ob eine Bewegung abgeschlossen ist.

Kurzinfo

Status der Programmausführung abfragen

Syntax

erg = AXEND

Rückgabewert

erg = Achsstatus mit folgender Bedeutung:

Wert	Bit	Bedeutung
128	7	1 = Motor ist zurückgesetzt (reset), d.h. er ist startbereit und regelt wieder, zum Beispiel nach ERRCLR, MOTOR STOP, MOTOR ON.
64	6	1 = Lageregelung ist abgeschaltet, Motor ist aus
	4–5	nicht verwendet
8	3	1 = Motor ist im STOP-Zustand nach einem MOTOR STOP
4	2	1 = Drehzahlmodus ist aktiv
2	1	1 = Positioniervorgang ist aktiv
1	0	1 = Motor befindet sich im Stillstand

Befehlsgruppe

I/O

Querverweise

WAITAX, STAT

Syntax-Beispiel

```

NOWAIT ON // nicht warten bis Position erreicht ist
POSA 100000
WHILE (AXEND&2) DO
// Solange Positioniervorgang aktiv, Schleife
//wiederholen
  IF IN1 THEN // Wenn Eingang 01 gesetzt wird
    VEL 100 // Geschwindigkeit erhöhen
    POSA 100000
    WAIT IN1 OFF
    // Warten, bis Taste losgelassen
  ENDIF
ENDWHILE
  
```

Syntax-Beispiel

```

IF (AXEND&64) THEN
/* Ausgang 01 setzen, wenn Lageregelung
abgeschaltet */
  OUT 1 1
ELSE
  OUT 1 0
ENDIF
  
```

Programmbeispiele

AXEND_01.M

COMOPTGET

COMOPTGET liest aus dem Puffer der Kommunikationsoption *anz* Datenworte aus und schreibt sie in das Array *array* beim ersten Element beginnend.

Kurzinfo

Liest ein Telegramm der Kommunikationsoption

Syntax

COMOPTGET *anz* *array*

Parameter

array = der Name eines Arrays, das mindestens die Größe *anz* haben muss
anz = Anzahl Datenworte, die gelesen werden sollen

Portabilität

mit eingebauter Kommunikationsoption

Funktion der Kommunikationsoption

Die Lese- und Schreibparameter werden von der SyncPos-Option nicht verändert.

Kontrolldaten:

Die Funktionen des Steuerwortes (STW) und des Hauptsollwertes (HSW) hängen davon ab, wie der Parameter 700 gesetzt ist; Zustandswort und (ZSW) und Hauptistwert (HIW) sind immer aktiv.

	Parameter 700 "SyncPos EIN"	Parameter 700 "SyncPos AUS"
STW/HSW	Inaktiv	Aktiv
ZSW/HIW	Aktiv	Aktiv

Prozessdaten:

PCD's 1...4 von PPO Typ 2/ 4 und PCD's 1...8 von PPO Typ 5 sind nicht mit einer Parameternummer 915 und 916 festgelegt, sondern können frei in einem SyncPos Programm benutzt werden.

Der Befehl COMOPTGET kopiert die empfangenen Daten der Kommunikationsoption in ein Array, in dem jedes Array-Element ein Datenwort (16 bit) enthält.

Der Befehl COMOPTSEND kopiert die Daten von einem Array, in dem jedes Array Element ein Datenwort (16 bit) enthält, in einen Sendepuffer der Kommunikationsoption, von dem es via Netzwerk zum Master gesendet wird.

Befehlsgruppe

Kommunikationsoption

Querverweise
COMOPTSEND

Programmbeispiel
COM_OPT

■ COMOPTSEND

COMOPTSEND schreibt in den Puffer der Kommunikationsoption. Dabei werden aus array die ersten *anz* Datenworte gesendet.

Kurzinfo
Schreibt in den Puffer der Kommunikationsoption

Syntax
COMOPTSEND *anz* array

Parameter

array	= der Name eines Arrays, das mindestens die Größe <i>anz</i> haben muss
anz	= Anzahl der Datenworte, die gesendet werden sollen

Portabilität
mit eingebauter Kommunikationsoption

Funktion der Kommunikationsoption
siehe COMOPTGET

Befehlsgruppe
Kommunikationsoption

Querverweis
COMOPTGET

Programmbeispiele
COM_OPT

■ CONTINUE

Mit CONTINUE können Positionier- und Drehzahlbefehle, die durch den Befehl MOTOR STOP oder einen Fehlerzustand abgebrochen oder mit MOTOR OFF angehalten wurden, fortgesetzt werden. CONTINUE kann besonders in einem Fehlerunterprogramm in Verbindung mit dem Befehl ERRCLR eingesetzt werden, um nach einem Fehlerabbruch den Bewegungsablauf korrekt weiterzuführen.



ACHTUNG!

CONTINUE setzt aber nicht abgebrochene Synchronisationsbefehle fort.

Kurzinfo
Abgebrochene Positionier- und Drehzahlbefehle fortsetzen

Syntax
CONTINUE

Befehlsgruppe
CON

Querverweise
MOTOR STOP, ERRCLR, ON ERROR GOSUB

Syntax-Beispiel
CONTINUE
/* Unterbrochene Bewegungsvorgänge fortsetzen */

Programmbeispiele
MSTOP_01.M

■ CPOS

Mit dem Befehl CPOS kann die aktuelle Kommando-
position einer Achse absolut zum aktuellen Null-
punkt abgefragt werden. Unter der Kommando-
position versteht man die temporäre Sollposi-
tion, die durch die Lageregelung jede Millisekunde neu be-
rechnet wird.

Die Kommando-
position kann unabhängig vom Be-
triebszustand (Lageregelung im Stillstand, Positio-
niervorgang, Drehzahlregelung oder Synchronisa-
tion) abgefragt werden.

Kurzinfo

Aktuelle Kommando-
position der Achse abfragen

Syntax

erg = CPOS

Rückgabewert

erg = Kommando-
position in Benutzereinheiten (BE)
absolut zum aktuellen Nullpunkt.


ACHTUNG!

Wenn ein mit SET ORIGIN gesetzter und
aktiver Temporärnullpunkt existiert, bezieht
sich der Positionswert auf diesen Nullpunkt.

Befehlsgruppe

I/O

Querverweise

APOS, DEF ORIGIN, SET ORIGIN, POSA, POSR
Parameter: POSFACT_Z (23), POSFACT_N (26)

Syntax-Beispiel

PRINT CPOS

/* aktuelle Kommando-
position der Achse am PC
ausgeben */

Programmbeispiele

CPOS_01.M
GOSUB_01.M

■ CSTART

Mit dem Befehl CSTART wird ein drehzahl-
geregelter Fahrbefehl gestartet.

Die Beschleunigungsrampe sowie die Drehzahl
sollte vor dem Starten des Drehzahlmodus mit den
Befehlen ACC, DEC und CVEL festgelegt werden.

CSTART enthält nicht den Befehl MOTOR ON, der die
Motorregelung einschaltet. Nach vorangegangenem
MOTOR OFF ist bei Verwendung von CSTART also
ein explizites Aufrufen von MOTOR ON notwendig.

Kurzinfo

Starten des Drehzahlmodus

Syntax

CSTART


ACHTUNG!

Wenn zum Zeitpunkt des CSTART noch kein
Drehzahlwert mit CVEL definiert wurde, wird
die Default-Geschwindigkeit 0 verwendet und der
Motor dreht sich nicht, die Lageregelung ist aber
aktiv.

Alle nach dem Start des Drehzahlmodus folgenden
CVEL Befehle werden sofort ausgeführt: Es wird
sofort eine entsprechende Drehzahlanpassung mit
der durch ACC bzw. DEC definierten Beschleuni-
gungs- bzw. Bremsrampe vorgenommen.

Befehlsgruppe

DRE

Querverweise

ACC, DEC, CVEL, CSTOP

Syntax-Beispiel

CSTART /* Drehzahlmodus starten */

Programmbeispiel

CMODE_01.M

■ CSTOP

Mit dem CSTOP Befehl wird der Drehzahl-Regelmodus verlassen und in den Positioniermodus geschaltet. Dabei wird eine noch drehende Achse mit der durch DEC definierten Verzögerung abgebremst und der Motor in der Stop-Position angehalten.

Kurzinfo

Stoppen des Antriebs im Drehzahlmodus

Syntax

CSTOP



ACHTUNG!

Ein im Positioniermodus ausgeführter CSTOP Befehl führt ebenfalls zu einem abrupten Abbrechen des Positioniervorganges.

Befehlsgruppe

DRE

Querverweise

ACC, DEC, CVEL, CSTART

Syntax-Beispiel

CSTOP /* Drehzahlmodus stoppen */

Programmbeispiel

CMODE_01.M

■ CURVEPOS

Mit dem Befehl CURVEPOS kann die Slave-Kurvenposition, die der aktuellen Master-Position entspricht, abgefragt werden.

Die Position kann unabhängig vom Betriebszustand (Lageregelung im Stillstand, Positioniervorgang, Drehzahlregelung oder Synchronisation) abgefragt werden.

CMASTERCPOS (SYSVAR) und CURVEPOS werden nun auch aktualisiert, sogar wenn SYNCC nicht mehr aktiv ist. Diese Werte werden aktualisiert nach einem Befehl SETCURVE (falls SYNCMSTART < 2000 ist) oder nach SYNCC und dem erster Master-Marker (falls SYNCMSTART = 2000).

Nachdem der SYNCC Befehl angehalten wurde, wird fortgefahren, diese Werte zu aktualisieren, wenn SYNCMSTART < 2000.

Kurzinfo

Slave-Kurvenposition, die der aktuellen Master-Position der Kurve entspricht, abfragen.

Syntax

erg = CURVEPOS

Rückgabewert

erg = Slave-Position in CAM-Einheiten (BE) absolut zum aktuellen Nullpunkt



ACHTUNG!

Die Position ist nur definiert, wenn zuvor ein SETCURVE gesetzt wurde.



ACHTUNG!

Wenn ein mit SET ORIGIN gesetzter und aktiver Temporärnullpunkt existiert, bezieht sich der Positionswert auf diesen Nullpunkt.



ACHTUNG!

DEFMCPOS und DEFMORIGIN können diese Position noch verändern.

Befehlsgruppe

CAM

Querverweise

APOS, DEF ORIGIN, SET ORIGIN, POSA, POSR, DEFMCPOS

Parameter: SYNCFAC TM, SYNCTACTS

Syntax-Beispiel

PRINT CURVEPOS

// aktuelle Slave-Position der Kurve ausgeben

■ CVEL

Mit dem CVEL Befehl wird die Geschwindigkeit für die nächsten drehzahlgeregelten Motorbewegungen gesetzt. Der Wert bleibt so lange gültig bis mit einem weiteren CVEL Befehl eine neue Geschwindigkeit gesetzt wird.

Der zu übergebende Geschwindigkeitswert bezieht sich auf die Parameter **Maximalgeschwindigkeit** VELMAX (1) und **Geschwindigkeitsteiler** VELRES (22).

Kurzinfo

Geschwindigkeit für drehzahlgeregelte Motorbewegungen setzen.

Syntax

CVEL v

Parameter

v = Geschwindigkeitswert (negativer Wert für andere Drehrichtung)

$$\text{Sollgeschwindigkeit (U/min)} = v \cdot \frac{\text{VELMAX (1)}}{\text{VELRES (22)}}$$


ACHTUNG!

CVEL Befehle, die nach einem CSTART folgen, werden sofort ausgeführt, das heißt die Geschwindigkeit wird mit der durch ACC/DEC vorgegebenen Beschleunigung bzw. Verzögerung auf den mit CVEL übergebenen neuen Wert angepasst.

Wurde vor dem Starten des Drehzahlmodus (CSTART) noch keine Geschwindigkeit definiert, ist die Standardgeschwindigkeit 0. Der Motor dreht sich also nicht und erst eine Geschwindigkeitsvorgabe mit CVEL startet die Bewegung im Drehzahlmodus.

Befehlsgruppe

DRE

Querverweise

ACC, DEC; CSTART, CSTOP

Parameter: VELMAX (1), VELRES (22)

Syntax-Beispiel

CVEL 100 /* Geschwindigkeit 100 */

Programmbeispiele

CMODE_01.M

■ DEC

Mit dem Befehl DEC bestimmen Sie Verzögerung (negative Beschleunigung) für die nächsten Fahrbefehle im Drehzahl-, Positionier- oder Synchronisationsmodus.

Der Wert bleibt so lange gültig, bis mit einem weiteren Befehl DEC eine neue Verzögerung gesetzt wird. Der Wert bezieht sich auf die Parameter **Kürzeste Rampe** RAMPMIN (31), **Maximalgeschwindigkeit** VELMAX (1) und **Geschwindigkeitsteiler** VELRES (22).


ACHTUNG!

Wenn man mit der SyncPos-Optionskarte arbeitet, sollte man die Rampen immer über die Optionskarte einstellen und nicht im VLT. Die VLT-Rampen müssen dabei immer auf Minimum stehen.

Kurzinfo

Verzögerung (negative Beschleunigung) setzen

Syntax

DEC a

Parameter

a = Verzögerung


ACHTUNG!

Wurde vor einem Positionierbefehl noch keine Verzögerung definiert, wird mit dem im Parameter DFLTACC (34) vorgegebenen Defaultwert (negativ) abgebremst.

Befehlsgruppe

REL, ABS

Querverweise

ACC

Parameter: RAMPMIN (31), VELMAX (1), VELRES (22)

Syntax-Beispiel

ACC 50 // Beschleunigung 50, beim Bremsen 10

DEC 10

Beispiel

kürzeste Rampe: 1000 ms

Maximalgeschwindigkeit: 1500 U/min

Geschwindigkeitsteiler: 100

■ DEFMCPOS

DEFMCPOS definiert die Anfangsposition des Masters (in MU) im CAM-Mode und somit, wo die Kurve startet, sobald die Masterpulse gezählt werden.

Kurzinfo

Anfangsposition des Masters definieren.

Syntax

DEFMCPOS p

Parameter

p = Position in Benutzereinheiten (MU)

Befehlsgruppe

CAM

Querverweise

DEFMORIGIN, SETMORIGIN, SYNCC

Parameter: SYNCMSTART (62)

Syntax-Beispiel

```
DEFMCPOS 1000
    // internen MU-Zähler auf 1000 setzen
```

■ DEFMORIGIN

DEFMORIGIN definiert die aktuelle Master-Position als Nullpunkt für den Master. Die Master-Position (MAPOS) bezieht sich bis zu einer erneuten Definition mit DEFMORIGIN oder SETMORIGIN auf diesen Nullpunkt.

Kurzinfo

Aktuelle Master-Position als Nullpunkt für den Master setzen

Syntax

DEFMORIGIN

Befehlsgruppe

INI

Querverweise

MAPOS, SETMORIGIN

Syntax-Beispiel

```
DEFMORIGIN // Nullpunkt für Master definieren
```

■ DEF ORIGIN

Mit DEF ORIGIN wird die aktuelle Position als Nullpunkt gesetzt. Alle absoluten Positionierbefehle (POSA) beziehen sich fortan auf diesen Nullpunkt.

Kurzinfo

Aktuelle Position als Nullpunkt setzen

Syntax

DEF ORIGIN

Befehlsgruppe

INI

Querverweise

POSA

Syntax-Beispiel

```
DEF ORIGIN /* Realnullpunkt definieren */
```

Programmbeispiele

DORIG_01.M, ORIG_01.M

■ DEF SYNCORIGIN

Mit diesem Befehl kann Master-Slave-Verhältnis für den nächsten SYNCP oder SYNCM Befehl definiert werden. Er setzt die interne Slave-Befehlsposition auf den Wert des Slaves.

Der Wert des Masters wird für einen internen MOVE-SYNCORIGIN benutzt. Dafür wird ein MOVESYNCORIGIN durch diesen Befehl überschrieben. Beide Aktionen werden in dem Moment ausgeführt, wenn der SYNC Befehl aktiviert wird. Das garantiert, dass Master und Slave auf die o.g. Master-Slave-Position synchronisiert werden

Kurzinfo

Definiert Master-Slave-Verhältnis für den nächsten SYNCP oder SYNCM Befehl

Syntax

DEFSYNCORIGIN master slave

Parameter

master = Sollposition in qc

slave = Sollposition

Portabilität

Standardbefehl mit Optionskarte Version 5.00 aufwärts

Befehlsgruppe

SYN

Querverweis

MOVESYNCORIGIN

■ DELAY

Der DELAY Befehl führt zu einer definierten Programmverzögerung. Der Übergabeparameter gibt dabei die Verzögerungszeit in Millisekunden an. Wenn während der Verzögerungszeit ein Interrupt auftritt, wird nach der Abarbeitung der Interrupt-Prozedur der Wartevorgang unter korrekter Einbeziehung der innerhalb der Interrupt-Prozedur abgelaufenen Zeit fortgesetzt. Der DELAY Befehl führt somit zu einer konstanten Wartezeit unabhängig davon, ob verschiedene Interrupts während der Verweilzeit behandelt werden mussten.

Nimmt der Interrupt mehr Verarbeitungszeit in Anspruch als restliche Verweilzeit zur Verfügung steht, wird die Interrupt-Prozedur zu Ende abgearbeitet, bevor mit dem auf die DELAY Anweisung folgenden Befehl das Programm fortgesetzt wird.

Kurzinfo

Zeitverzögerung

Syntax

DELAY t

Parameter

t = Verzögerungszeit in Millisekunden (maximal MLONG)

Befehlsgruppe

CON

Querverweise

WAITT, WAITI, WAITAX

Syntax-Beispiel

DELAY 1000 /* 1 Sekunde Verzögerung */

Programmbeispiele

DELAY_01.M

■ DELETE ARRAYS

Mit DELETE ARRAYS können Sie alle Arrays im RAM löschen, ohne – wie bisher – auch die Parameter etc. zu löschen. Dieser neue Befehl bewirkt das Gleiche, wie der Menübefehl "STEUERUNG" → "RESET" → "ARRAYS".

Kurzinfo

Alle Arrays im RAM löschen.

Syntax

DELETE ARRAYS



ACHTUNG!

Wenn Sie anschließend ein SAVE ARRAYS durchführen, werden auch die Arrays im EEPROM überschrieben!



ACHTUNG!

Falls DELETE ARRAYS nach einer DIM Anweisung im Programm durchgeführt wird, darf danach nicht mehr auf die Array-Elemente zugegriffen werden.



ACHTUNG!

Wenn ein Programm einen DELETE ARRAYS Befehl enthält, gibt es nach Verlassen des Programmes im RAM keine Arrays mehr.

Befehlsgruppe

INI

■ DIM

Mit einer DIM Anweisung am Programmumfang vereinbaren Sie die Verwendung von ein oder mehreren Arrays (= Variablenfeldern).

Arrays besitzen Gültigkeit für alle im VLT abgelegten Programme. Sollten noch keine Arrays im Speicher der SyncPos-Option vorhanden sein, werden durch die DIM Anweisung die Arrays angelegt. Bei bereits im Speicher vorhandenen Arrays wird überprüft, ob deren Größe mit der aktuellen DIM Anweisung übereinstimmt. Sollten dabei Unterschiede auftreten, wird eine Fehlermeldung ausgegeben. Wenn zusätzlich zu den übereinstimmenden Arrays noch weitere neue Arrays erklärt sind, müssen diese an das Ende der DIM Anweisung angefügt werden.

Auf jedes Array-Element kann später ähnlich wie auf eine Variable zugegriffen und es können Rechenergebnisse, Zeichen oder andere Informationen abgelegt werden.

Ein Array-Element wird über den Array-Namen und einen Index angesprochen. Die Indizes sind dabei von 1 bis zu der in der DIM Anweisung definierten Größe zulässig.

Ein wesentlicher Unterschied zwischen Variablen und Array-Elementen besteht jedoch darin, dass Arrays im nicht flüchtigen Speicherbereich abgelegt sind und ihr Inhalt – sofern mit SAVEPROM oder SAVE ARRAYS gesichert – auch beim Abschalten der Versorgungsspannung erhalten bleibt.

Im Gegensatz zu Variablen besitzen Arrays nicht nur für ein Programm, sondern für alle im VLT abgelegten Programme Gültigkeit. Einzige Voraussetzung dafür ist, dass die Arrays mit einer DIM Anweisung in den gewünschten Programmen zugänglich gemacht werden, wodurch ein Datenaustausch zwischen mehreren Programmen möglich wird. Es spielt hierbei keine Rolle, ob das Array in allen Programmen durch den gleichen Namen gekennzeichnet ist. Entscheidend ist lediglich die Reihenfolge der Array-Definitionen. Wodurch zum Beispiel das erste definierte Array in allen Programmen immer auf das erste im Speicher abgelegte Array zugreift, unabhängig vom Array-Namen.

Kurzinfo

Definition eines Arrays

Syntax

DIM array [n]

Parameter

array = Name des Arrays

n = Anzahl der Array-Elemente


ACHTUNG!

Die DIM Anweisung muss die erste Anweisung in einem Programm sein und noch vor dem Unterprogrammbereich erscheinen!

Indizes sind von 1 bis zur Größe des definierten Arrays erlaubt.

Eine einmal definierte Array-Größe gilt für alle Programme und kann nicht geändert werden. Einzig die Reihenfolge der Array-Definition (und nicht der Namen) bestimmt, auf welche Datenfelder zugegriffen wird. Array-Definitionen können nur durch das Löschen des gesamten Speichers rückgängig gemacht werden!

Befehlsgruppe

CON

Syntax-Beispiel

```
DIM xpos[100], ypos[100]
/* Array XPOS und YPOS mit je 100 Elementen
definieren */
```

Programmbeispiele

DIM_01.M

■ DISABLE ... Interrupts

DISABLE schaltet alle oder explizit genannte Interrupts – außer ON ERROR – ab. Wenn die Funktion DISABLE ... im Hauptprogramm verwendet wird, kann sie Interrupts der entsprechenden Art verhindern.

Dies ist insbesondere nützlich, wenn eine Variable, die in einer Interruptprozedur gesetzt ist, im Hauptprogramm verwendet wird.

Dazu sollten Sie im Hauptprogramm zunächst die entsprechenden (oder alle) Interrupts mit DISABLE... abschalten, die Variable ändern und anschließend die entsprechenden (oder alle) Interrupts mit ENABLE... wieder einschalten.

Kurzinfo

Schaltet Interrupts ab

Syntax

DISABLE inttyp


ACHTUNG!

DISABLE kann nicht innerhalb von Interruptprozeduren aufgerufen werden. (Nach einem Interrupt wird automatisch wieder enabled.)

Parameter

```
inttyp =  ALL
          INT
          COMBIT
          STATBIT
          PARAM
          PERIOD
          TIME
          POSINT (ON APOS, ON MAPOS, ON
                 MCPOS)
```


ACHTUNG!

DISABLE ALL = alle außer ON ERROR; außerdem werden die Interrupts nach der Freigabe mit ENABLE ALL noch ausgeführt.


ACHTUNG!

Wird ein Interrupt disabled (d.h. gesperrt) existiert er weiterhin, wird aber nicht mehr ausgeführt. (Ausnahme: DISABLE ALL)

Die Erkennung läuft weiter im Hintergrund und die Interrupt-Anforderung wird im Fall eines nicht (!) flanken-getriggerten oder nachrichten-orientierten Interrupts (ON PERIOD, ON APOS, ON PARAM, etc.) gespeichert. Wenn der Interrupt dann wieder enabled (d.h. freigegeben) wird und es zuvor einen noch nicht ausgeführten, gespeicherten (nicht flanken-getriggerten) Interrupt gab, wird dieser Interrupt sofort ausgeführt.

Im Fall eines flanken-getriggerten Interrupts (z.B. ON INT, ON COMBIT, ON STATBIT), werden alle Interrupts, die während der DISABLE-Phase stattgefunden haben, nicht ausgeführt, auch dann nicht, wenn wieder auf ENABLE umgeschaltet wird. Diese Interrupts werden im DISABLE-Status nicht gespeichert. Flanken-getriggerte Interrupts, die nach dem erneuten ENABLE stattfinden, werden weiterhin wieder ausgeführt.


ACHTUNG!

Ausnahme: DISABLE ALL

Während bei dem selektiven Sperren flanken-getriggerten Interrupts (z.B. DISABLE INT) diese Interrupts, wie beschrieben, ignoriert und auch nach der Freigabe nicht mehr ausgeführt werden, wird bei DISABLE ALL die Anforderung (auch von flanken-getriggerten Interrupts) gespeichert und der Interrupt nach der Freigabe (ENABLE ALL) noch ausgeführt!

Befehlsgruppe

INT

Querverweise

ON INT, ON COMBIT, ON STATBIT, ON PARAM, ON PERIOD, ON TIME, ENABLE .. Interrupts

Syntax-Beispiel

```
DISABLE ALL /* Alle Interrupts abschalten */
DISABLE STATBIT
           /* Interrupt für Statusbit abschalten */
```

■ ENABLE ... Interrupts

Kurzinfo

Schaltet alle oder explizit genannte Interrupts wieder ein.

Syntax

ENABLE inttyp



ACHTUNG!

ENABLE kann nicht innerhalb von Interrupt-prozeduren aufgerufen werden. (Nach einem Interrupt wird automatisch wieder enabled.)

Parameter

inttyp = ALL
 INT
 COMBIT
 STATBIT
 PARAM
 PERIOD
 POSINT
 TIME



ACHTUNG!

Siehe Unterschiede zwischen den nicht-flanken-getriggerten und nachrichten-orientierten Interrupts (ON PERIOD, ON APOS, ON PARAM, etc.) und den flanken-getriggerten Interrupts (z.B. ON INT, ON COMBIT, ON STATBIT) bei DISABLE...

Befehlsgruppe

INT

Querverweise

ON INT, ON COMBIT, ON STATBIT, ON PARAM, ON PERIOD, ON TIME, DISABLE .. Interrupts

Syntax-Beispiel

ENABLE ALL /* Alle Interrupts einschalten */
 ENABLE COMBIT
 /* Interrupt für Kommunikationsbit einschalten */

■ ERRCLR

Fehlermeldungen der SyncPos-Option werden nicht automatisch nach dem Beseitigen der Fehlerursache gelöscht, sondern müssen durch ein ERRCLR gelöscht werden. Voraussetzung ist jedoch, dass die Fehlerursache auch tatsächlich beseitigt wurde, da sonst die gleiche Fehlermeldung noch mal auftritt. Wenn zwischenzeitlich ein weiterer, noch nicht behobener Fehler aufgetreten ist, wird nur die erste Fehlermeldung gelöscht.

Kurzinfo

Löschen einer Fehlermeldung

Syntax

ERRCLR

Der ERRCLR Befehl sollte nur in einem Unterprogramm zur Fehlerbehandlung eingesetzt werden (siehe ON ERROR GOSUB).



ACHTUNG!

ERRCLR beinhaltet den Befehl MOTOR ON, der die Regelung automatisch wieder einschaltet. (Der Motor wird auf aktueller Position geregelt.)

BIS VERSION 3.00

ERRCLR setzt nur die SyncPos-Option Fehler zurück; VLT-Meldungen müssen über einen digitalen Eingang oder das LCP-Display gelöscht werden.

AB VERSION 3.1x

ERRCLR setzt ab Version 3.1x auch Bit 7 des Steuerwortes vom VLT zurück. Die VLT-Meldungen müssen dann nicht mehr gelöscht werden.

Befehlsgruppe

INI, CON

Querverweise

ON ERROR GOSUB, ERRNO, CONTINUE, MOTOR ON
 Meldungen und Fehlerreferenz

Syntax-Beispiel

ERRCLR /* aktuelle Fehlermeldung löschen */

Programmbeispiele

ERROR_01.M
 IF_01.M, INDEX_01.M

■ ERRNO

ERRNO ist eine Systemvariable, die in allen Programmen verfügbar ist und die aktuelle Fehlernummer enthält. Alle Fehlernummern sind in der Meldungen und Fehler-Referenz erläutert. Für den Fall, dass zum Abfragezeitpunkt kein Fehler aufgetreten ist, enthält ERRNO eine 0.

Kurzinfo

Systemvariable mit der aktuellen Fehlernummer

Syntax

erg = ERRNO

Befehlsgruppe

I/O

Querverweise

ON ERROR GOSUB, ERRCLR
Meldungen und Fehlerreferenz

Syntax-Beispiel

PRINT ERRNO
/* aktuelle Fehlernummer ausgeben */

Programmbeispiele

ERROR_01.M
IF_01.M, INDEX_01.M

■ EXIT

Der EXIT Befehl beendet ein Programm, wobei aktive Positionierprozesse noch zu Ende ausgeführt werden.

Der EXIT Befehl ist besonders für den Einsatz in einer Fehlerbehandlungsroutine vorgesehen und ermöglicht zum Beispiel bei nicht behebbaren Fehlern einen gezielten Programmabbruch.

Ein mit "AUTOSTART" gekennzeichnetes Programm wird nach einem Abbruch mit EXIT automatisch wieder anlaufen, wenn SET PRGPAR = -1.

Kurzinfo

Vorzeitiger Programmabbruch

Syntax

EXIT


ACHTUNG!

Normalerweise sollte ein Programm nur bei schwerwiegenden Fehlern, wie zum Beispiel beim Ansprechen eines Endschalters, abgebrochen werden.

Befehlsgruppe

CON

Querverweise

ON ERROR GOSUB
SET PRGPAR
"AUTOSTART"

Syntax-Beispiel

EXIT /* Programmabbruch */

Programmbeispiele

EXIT_01.M
ERROR_01.M

■ GET

GET liest einen Achsparameter, einen globalen Parameter oder einen Benutzerparameter der SyncPos-Option.

Die Parameterkennungen finden Sie in der Übersicht der Parameter-Referenz in der linken Spalte, zum Beispiel KPROP (11) für den **Proportionalfaktor** oder POSERR (15) für den **Tolerierten Positionsfehler** und I_BREAK (105) für den **Eingang für Abbruch**.

Benutzerparameter sind mit einer Nummer von 130 bis 229 gekennzeichnet, die Sie auch in der Parameter-Referenz finden.

Kurzinfo

Liest einen Achsparameter, einen globalen Parameter oder einen Benutzerparameter

Syntax

erg = GET par

Parameter

par = Parameterkennung

Rückgabewert

erg = aktueller Parameterwert

Befehlsgruppe

PAR

Querverweise

SET, GETVLT, SETVLT, LINKGPARG, LINKAXPAR
 Parameter-Referenz

Syntax-Beispiel

```
PRINT GET POSLIMIT
/* Positives Fahrbereichslimit ausgeben */
schlepp = GET POSERR
/* Lese aktuelle Einstellung des Tolerierten Positionsfehlers */
PRINT GET I_BREAK
/* Eingang für Abbruch lesen */
```

Programmbeispiele

GETP_01.M

■ GETVLT

GETVLT liest einen VLT-Parameter und liefert den entsprechenden Wert zurück. Mit GETVLT haben Sie somit Zugriff auf Betriebsdaten (z.B. Motorstrom P520) oder auf Konfigurationen (z.B. max. Sollwert P205) des VLT.

Da ausschließlich Ganzzahlwerte übertragen werden, muss bei der Auswertung des Rückgabewertes der Konvertierungsindex beachtet werden.

So ist ein LCP Wert von 50,0 Hz (Konvertierungsindex = -1) gleichbedeutend mit einem Rückgabewert von 500.

Die Liste der VLT-Parameter mit dem zugehörigen Konvertierungsindex finden Sie in der VLT5000/VLT5000Flux Betriebsanleitung.

Kurzinfo

Liest einen VLT-Parameter

Syntax

erg = GETVLT par


ACHTUNG!

Benutzen Sie GETVLTSUB um Parameter mit Indexnummern zu lesen, z.B. die VLT Parameter 606...617.

Parameter

par = Parameternummer

Rückgabewert

erg = aktueller Parameterwert

Befehlsgruppe

PAR

Querverweise

SETVLT

Syntax-Beispiel

```
PRINT GETVLT 202
/* Lese Parameter 202 Ausgang hohe Frequenz */
```

■ GETVLTSUB

GETVLTSUB liest einen VLT Parameter inklusive der Indexnummer, z.B. die VLT Parameter 606...617, und gibt den entsprechenden Wert zurück.
Da ausschließlich Ganzzahlenwerte übertragen werden, muss bei der Auswertung des Rückgabewertes der Konvertierungsindex beachtet werden. So ist ein LCP Wert von 50,0 Hz (Konvertierungsindex = -1) gleichbedeutend mit einem Rückgabewert von 500. Die Liste der VLT-Parameter mit dem zugehörigen Konvertierungsindex finden Sie in der VLT5000/ VLT5000Flux Betriebsanleitung.

Kurzinfo

Liest einen VLT Parameter mit Indexnummer

Syntax

erg = GETVLTSUB par indxn

Parameter

par = Parameter Nummer
indxn = Indexnummer

Rückgabewert

erg = Parameterwert

Portabilität

Mit Optionskarte Version 5.00 aufwärts

Befehlsgruppe

PAR

Querverweise

SETVLTSUB

Syntax-Beispiel

```
PRINT GETVLTSUB 25 1
  // Index 1 des Parameters Quick Menü
  // (nur VLT5000Flux) ausgeben
```

■ GOSUB

Der GOSUB Befehl ruft ein Unterprogramm auf und der zugehörige Programmbereich wird abgearbeitet. Nach dem letzten Unterprogrammbehl (RETURN) wird im Hauptprogramm mit dem auf die GOSUB Anweisung folgenden Befehl fortgefahren.

Kurzinfo

Aufruf eines Unterprogrammes

Syntax

GOSUB name

Parameter

name = Name des Unterprogrammes


ACHTUNG!

Unterprogramme müssen am Anfang oder Ende des Programms innerhalb des SUBMAINPROG Bereichs definiert sein.

Befehlsgruppe

CON

Querverweise

SUBMAINPROG...ENDPROG,
SUBPROG...RETURN
ON ERROR GOSUB, ON INT n GOSUB

Syntax-Beispiel

```
GOSUB testup
/* Aufruf des Unterprogrammes testup */
  Befehlszeile 1
  Befehlszeile n
SUBMAINPROG
/* Unterprogramm testup muss definiert sein */
SUBPROG testup
  Befehlszeile 1
  Befehlszeile n
RETURN
ENDPROG
```

Programmbeispiele

GOSUB_01.M
AXEND_01.M, INCL_01.M, STAT_01.M

■ GOTO

Mit dem GOTO Befehl wird unbedingt zu der angegebenen Programmposition gesprungen und die Programmabarbeitung an dieser Position fortgesetzt. Die Programmposition, zu der gesprungen werden soll, ist durch ein Label gekennzeichnet. Ein Label kann aus einem oder mehreren Zeichen bestehen und darf nicht mit einem Variablennamen oder einem Befehlswort identisch sein. Ein Label muss zudem eindeutig sein, es darf nicht mehrfach an unterschiedlichen Programmpositionen verwendet werden.

Mit dem GOTO Befehl ist es zum Beispiel möglich, eine Endlosschleife zu programmieren.

Kurzinfo

Sprung zu einem Programmlabel

Syntax

GOTO label

Parameter

label = Kennung der Programmzielposition


ACHTUNG!

Das Label muss an der Programmzielposition mit einem Doppelpunkt beendet werden, z.B. endlos:

Befehlsgruppe

CON

Querverweise

LOOP

Syntax-Beispiel

endlos:

/* Label zu dem gesprungen wird */

Befehlszeile 1

Befehlszeile n

GOTO endlos

/* Sprungbefehl zu Label endlos */

Programmbeispiele

GOTO_01.M

EXIT_01.M, IF_01.M

■ HOME

Der HOME Befehl fährt den Antrieb zum Referenzschalter und setzt den Maschinennullpunkt als Referenzposition.

Die Geschwindigkeit und Beschleunigung/Verzögerung für die Homefahrt wird mit den Parametern HOME_VEL (7) und HOME_RAMP (41) festgelegt. Um eine exakte Positionierung zu erreichen, sollte die **Home-Geschwindigkeit** HOME_VEL (7) nicht höher sein als 10% der Maximaldrehzahl.

Das Vorzeichen von HOME_VEL (7) bestimmt, in welcher Drehrichtung nach dem Referenzschalter gesucht wird.

Wenn die Home-Position erreicht ist, wird diese als Nullpunkt definiert.

Der Referenzschalter kann in vier verschiedenen Arten anfahren werden. Welche Art Homefahrt durchgeführt wird, wird im Parameter HOME_TYPE (40) festgelegt.

0 = Fahren bis zum Referenzschalter, Reversieren und den Referenzschalter verlassen und beim nächsten Indeximpuls (Encoder Nullimpulse oder externes Markersignal) halten.

1 = Wie 0, aber ohne Suchen des Indeximpulses.

2 = Wie 0, aber ohne Reversieren, sondern in gleicher Richtung weiter aus dem Schalter heraus.

3 = Wie 2, aber ohne Suchen des Indeximpulses.

Wird die Homefahrt durch einen Interrupt abgebrochen, wird HOME nicht automatisch weitergeführt wenn die Interrupt-Routine wieder verlassen wird. Statt dessen wird mit dem nächsten Befehl fortgefahren. Dies dient dazu, dass nach einem Störfall HOME auch abgebrochen werden kann.

Kurzinfo

Maschinennullpunkt (Referenzschalter) anfahren und als Realnullpunkt setzen

Syntax

HOME

Besonderheiten

Die Anlage muss mit einem Referenzschalter sowie nach Möglichkeit mit Drehgebern mit Indeximpuls ausgestattet sein!


ACHTUNG!

Der HOME Befehl wird auch bei NOWAIT ON zu Ende ausgeführt, bevor mit der weiteren Programmabarbeitung begonnen wird.


ACHTUNG!

Bitte beachten Sie, dass ON PERIOD xx GOSUB xx während der Homefahrt deaktiviert sein muss. Zum Beispiel ON PERIOD n GOSUB x und dann Reset, nachdem die Homefahrt beendet ist.

Befehlsgruppe

INI

Querverweise

INDEX

Parameter: HOME_VEL (7), HOME_RAMP (41), HOME_TYPE (40), HOME_FORCE (3)

Syntax-Beispiel

HOME

/* Referenzschalter und Index anfahren */

Programmbeispiele

HOME_01.M

■ IF ..THEN .., ELSEIF .. THEN .. ELSE .. ENDIF

Mit der IF..ENDIF Konstruktion können bedingte Programmverzweigungen realisiert werden.

Ist die hinter IF bzw. ELSEIF stehende Bedingung erfüllt, werden die Befehle bis zur nächsten ELSEIF, ELSE oder ENDIF Anweisung ausgeführt und dann mit den nach der ENDIF Anweisung stehenden Befehlen das Programm fortgesetzt.

Ist die Bedingung nicht erfüllt, werden die nachfolgenden ELSEIF Verzweigungen überprüft und sofern die Bedingung erfüllt ist, wird der entsprechende Programmteil ausgeführt und das Programm nach ENDIF fortgesetzt.

Die Verzweigungsbedingung, die nach einer IF oder ELSEIF Anweisung überprüft wird, kann sich aus einer oder mehreren Vergleichsoperationen zusammensetzen.

Innerhalb der IF..ENDIF Konstruktion können beliebig viele ELSEIF Verzweigungen auftreten, es darf jedoch nur eine ELSE Anweisung vorhanden sein. Hinter der ELSE Anweisung steht der Programmteil, der abgearbeitet wird, sofern keine der Bedingungen erfüllt wurde.

Die Anweisungen ELSEIF und ELSE können, aber müssen nicht innerhalb einer IF ENDIF Konstruktion enthalten sein.

Kurzinfo

Bedingte ein- oder mehrfache Programmverzweigung (Wenn Bedingung erfüllt, dann führe aus ..., sonst ...)

Syntax

IF (Bedingung) THEN Befehl

ELSEIF (Bedingung) THEN Befehl

ELSE Befehl

ENDIF

Parameter

Bedingung = Verzweigungskriterium

Befehl = ein oder mehrere Programmbefehle


ACHTUNG!

Nachdem eine Bedingung erfüllt wurde, wird der zugehörige Programmteil ausgeführt und das Programm nach der ENDIF Anweisung fortgesetzt. Weitere Bedingungen werden nicht mehr überprüft.

Befehlsgruppe

CON

Querverweise

REPEAT...UNTIL, WHILE..ENDWHILE

Syntax-Beispiel

/* Einfachverzweigung */

IF (a == 1) THEN /* Variable a = 1, dann */

Befehlszeile 1

Befehlszeile n

ENDIF

/* Mehrfachverzweigung */

IF (a == 1 AND b != 1) THEN

Befehlszeilen

ELSEIF (a == 2 AND b != 1) THEN

Befehlszeilen

ELSEIF (a == 3) THEN

Befehlszeilen

ELSE

Befehlszeilen

ENDIF

Programmbeispiele

IF_01.M

ERROR_01.M, EXIT_01.M, HOME_01.M, IN_01.M

■ IN

Mit dem IN Befehl können Sie den Zustand eines digitalen Eingangs abfragen: In Abhängigkeit von dem anliegenden Signalpegel wird eine 0 oder 1 zurückgeliefert.

Kurzinfo

Zustand eines digitalen Eingangs abfragen

Syntax

erg = IN n

Parameter

n = Nummer des Eingangs
 SyncPos-Option: 1 – 8
 VLT-Steuerkarte: 16, 17, 18, 19, 29, 32, 33

Rückgabewert

erg = Zustand des Eingangs
 0 = Low-Pegel oder undefiniert
 1 = High-Pegel


ACHTUNG!

Die Definition des High- und Low-Pegels sowie die Eingangsbeschaltung ist im Kapitel Ein- und Ausgangsklemmen und im VLT5000 Handbuch beschrieben.

Befehlsgruppe

I/O

Querverweise

INB, OUT, OUTB

Syntax-Beispiel

```
ein4 = IN 4
/* Zustand Eingang 4 in Variable ein4 speichern */
IF (IN 2) THEN
/* Bei High Pegel an Klemme 2 Ausgang 01 setzen */
  OUT 1 1
ELSE
  OUT 1 0
ENDIF
```

Programmbeispiel

IN_01.M

■ INAD

Der INAD Befehl liest den Wert des Analogeinganges.

Kurzinfo

Analogeingang lesen

Syntax

erg = INAD n

Parameter

n = Nummer des Analogeinganges: 53, 54 und 60

Rückgabewert

erg = Analogwert


ACHTUNG!

Mit Einführung der Softwareversion SyncPos VLT 3.12 wurde die Auflösung der Analogeingänge erhöht.

Für diese Versionen gelten die folgenden Rückgabewerte:

Klemme 53/54	-1000 ... 1000	= -10 V ... 10 V
Klemme 60:	0 ... 2000	= 0 – 20 mA

Für die Softwareversionen SyncPos VLT2.03 gelten folgende Rückgabewerte:

Klemme 53/54	0 ... 10 V	erg = 0 ... 100
Klemme 60:	0 ... 20 mA	erg = 0 ... 200

Die Nummer der Softwareversion können Sie im Parameter 624 am VLT-Display ablesen.

Befehlsgruppe

I/O

Syntax-Beispiel

```
an1 = INAD 53
PRINT "Analogeingang 53 " ,an1
```

■ INB

Mit dem INB Befehl können Sie den Zustand der digitalen Eingänge byteweise abfragen. Der zurückgelieferte Wert spiegelt den Zustand der einzelnen Eingänge wieder.

Kurzinfo

Zustand der digitalen Eingänge byteweise abfragen

Syntax

erg = INB n

Parameter

n = Eingangsbyte
 0 = Eingang 1 (LSB) ... 8 (MSB)
 1 = Eingang 33 (LSB) ... 16 (MSB)

Rückgabewert

erg = Wert des Eingangsbytes (0 ... 255)

Das niederwertigste Bit (LSB) entspricht dabei dem Zustand des Einganges 1/33.


ACHTUNG!

Die Definition des High- und Low-Pegels, sowie die Eingangsbeschaltung sind im Kapitel Ein- und Ausgangsklemmen und im VLT5000 Handbuch beschrieben.

Befehlsgruppe

I/O

Querverweise

IN, OUT, OUTB

Syntax-Beispiel

ein = INB 0
 /* Zustand der ersten 8 Eingänge speichern */

Beispiel

IN1 = low, IN2 = high, IN3 = high,
 alle anderen Eingänge low
 erg= 2¹ + 2² = 6

Programmbeispiele

INB_01.M, INB_02.M
 OUTB_01.M

■ INDEX

Der INDEX Befehl startet eine Fahrt zur Indexposition des Drehgebers. Die Indexsuche erfolgt mit der **Home-Geschwindigkeit**, die im Parameter HOME_VEL (7) festgelegt ist. Das Vorzeichen von HOME_VEL (7) bestimmt, in welcher Drehrichtung nach dem Indexsignal gesucht wird.

Kurzinfo

Indexposition des Drehgebers anfahren

Syntax

INDEX

Besonderheiten

Die verwendeten Drehgeber müssen über einen Indexkanal verfügen.


ACHTUNG!

Es müssen Drehgeber mit low-aktivem Indexpuls verwendet werden.

Wird innerhalb einer kompletten Umdrehung kein Indexpuls gefunden, erfolgt eine Fehlermeldung. Der INDEX Befehl wird auch bei NOWAIT ON zu Ende ausgeführt, bevor mit der weiteren Programmabarbeitung begonnen wird.

Befehlsgruppe

INI

Querverweise

HOME, POSA, DEF ORIGIN

Syntax-Beispiel

INDEX /* Index anfahren */

Programmbeispiele

INDEX_01.M

■ INKEY

Mit dem INKEY Befehl kann ein Tastensignal vom VLT-Tastenfeld eingelesen werden. Der mit INKEY übergebene Parameter bestimmt dabei, ob auf ein Tastensignal ohne Bedingung, eine gewisse Zeitspanne oder gar nicht gewartet wird. Pro erfolgreichem INKEY Befehl wird jeweils ein Tastensignal eingelesen. Für die Eingabe von Zeichenketten muss der INKEY Befehl ($p \diamond 0$) in einer Schleife so oft wiederholt werden, bis keine weiteren Tastensignale mehr vorliegen.

Kurzinfo

Einlesen eines Tastensignals

Syntax

INKEY (p)

Parameter

p = maximale Wartezeit, definiert in ...
 p = 0
 es wird gewartet bis ein Tastensignal kommt
 p > 0
 es wird maximal p Millisekunden gewartet
 p < 0
 es wird nicht auf ein Tastensignal gewartet

Rückgabewert

Tastencode

bzw. -1 falls kein Zeichen vorhanden ist

Die folgenden Tastencodes werden zurückgeliefert, solange die Tasten gedrückt werden. Werden mehrere Tasten gleichzeitig gedrückt, wird die entsprechende Summe der Werte zurückgeliefert.

<u>Taste:</u>	<u>Wert:</u>
MENU	1
QUICK MENU	2
DISPLAY / STATUS	8
OK	16
CANCEL	32
CHANGE DATA	128
→-Taste	256
↑-Taste	512
↓-Taste	1024
←-Taste	2048
START	4096
FWD. REV.	8192
JOG	16384
STOP / RESET	32768
Kombinationen senden entsprechende Werte:	
OK und CANCEL	48
START und ↑-key	4608


ACHTUNG!

Die Tasten behalten ihre ursprüngliche VLT-Funktion, wenn sie nicht in Parameter 014 bis 017 deaktiviert werden. Ein negativer Parameter muss in Klammern angegeben werden.

Befehlsgruppe

I/O

Querverweise

PRINT

Kapitel Grundlagen des SyncPos Programmes

Syntax-Beispiel

```
input = INKEY 0
/* Warten bis Tastensignal gelesen wird */
input = INKEY 5000
/* Max. 5 Sekunden warten, bis zur Eingabe */
input = INKEY (-1)
/* nicht auf Eingabe warten */
```

Programmbeispiele

INKEY_01.M

EXIT_01.M, WHILE_01.M

■ IPOS

Kurzinfo

Letzte Index- bzw. Markerposition des Slaves abfragen

Syntax

erg = IPOS

Rückgabewert

erg = letzte Slaveposition (Index- bzw. Marker) absolut zum aktuellen Nullpunkt.

Beschreibung

Der Befehl IPOS liefert die letzte Index- bzw. Markerposition des Slaves absolut zum aktuellen Nullpunkt zurück.
Die Positionsangabe wird in Benutzereinheiten (BE) zurückgeliefert und entspricht in der Standardeinstellung der Anzahl Quadcounts. Parameter POSFACT_Z (23) und POSFACT_N (26) = 1



ACHTUNG!

Wenn ein mit SET ORIGIN gesetzter und aktiver Temporärnullpunkt existiert, ist der Positionswert auf diesen Nullpunkt bezogen.

Die Konfiguration von IPOS, d.h. ob die Index- oder Markerposition des Slave (= geregelter Antrieb) zurückgeliefert wird, erfolgt über den Parameter SYNCMTYPS.



ACHTUNG!

Das Triggersignal für die Markerposition muss dabei zwingend an den Eingang 2 angeschlossen werden.

Der Positionswert in IPOS ist auf +/- 1qc genau. Im Gegensatz zu der Positionsinformation in APOS, welche nur im Reglerzyklus von typisch 1 ms aktualisiert wird, wird der aktuelle Positionswert hardwaremäßig beim Auftreten des konfigurierten Signals (in einem internen Prozessorregister) in Echtzeit zwischengespeichert und dann in die Systemvariable IPOS kopiert.

Falls gleichzeitig zur Markerposition ein Interrupt ausgelöst wird (ON INT 2 GOSUB ...) und in diesem Interrupt mit IPOS gearbeitet wird, sollte im Interrupt-Unterprogramm eine Verzögerung von 2 Millisekunden (DELAY 2) vor dem Lesen von IPOS verwendet werden. So kann sichergestellt werden, dass der gelatchte Positionswert bereits sicher in die Systemvariable IPOS kopiert ist und nicht noch auf einen veralteten Wert zurückgegriffen wird.

Siehe Beispiel.

Befehlsgruppe

I/O

Querverweise

CPOS, DEF ORIGIN, SET ORIGIN, POSA, POSR, MIPOS, SYNCMTYPS
Parameter: POSFACT_Z (23), POSFCT_N (26)

Syntax-Beispiel

```
PRINT IPOS
/* letzte Indexposition am PC ausgeben */
```

Beispiel

```
// Definition Interrupt-Handler
ON INT 2 GOSUB slave_int
// Definition von IPOS-Latching auf positive
// Flanke an Eingang 2
SET SYNCMTYPS 2
// Bewegung starten
CVEL 10
CSTART x(1)
// Endlos-Loop
mainloop:
// ...
GOTO mainloop

SUBMAINPROG
SUBPROG slave_int
// Testweise APOS zwischenspeichern
// um zu sehen, wie genau dies wäre ...
int_pos = APOS
// 2 Millisekunden warten, damit IPOS
// sicher aktualisiert ist
DELAY 2
// IPOS zwischenspeichern für spätere
// Bearbeitung etc.
triggered_pos = IPOS
// ...
// ...
PRINT „Interrupt position: „,int_pos
PRINT „Triggered position: „,triggered_pos
RETURN
ENDPROG
```

■ LINKAXPAR

Mit LINKAXPAR können Sie einen der Achsparameter mit dem LCP-Display verknüpfen. Anschließend können Sie den Parameter über das LCP verändern oder den gesetzten Wert lesen.

Wird ein verknüpfter Parameter mit einem SET-Befehl verändert, wird der Wert automatisch auch an das LCP übergeben, wirkt aber nur temporär. Ändert der Anwender einen verknüpften Parameter am LCP, wird der neue Wert übernommen und ausgeführt. Nach dem Aus- und Wiedereinschalten des VLT wird die Werkseinstellung, bzw. der zuletzt gespeicherte Benutzerparameter aktiviert. Erst wenn Sie den Parameter mit OK bestätigen, wird der neue Wert als Benutzerparameter übernommen und permanent gespeichert. Dies entspricht der Funktion OK im Menü "STEUERUNG" → "PARAMETER" → "ACHSEN".

Kurzinfo

Achsparameter mit LCP-Display verknüpfen

Syntax

LINKAXPAR param optpar "text" min max typ

Parameter

param = Achsenparametername
 optpar = LCP-Parameter Nummer (710–779 und 795–799)
 text = beschreibender Text für Display
 min = minimaler Wert, den der Parameter annehmen darf
 max = maximaler Wert, den der Parameter annehmen darf
 typ = Parameter Art
 0 = offline, d.h. Änderungen werden erst durch die Bestätigung mit OK aktiv.
 1 = online, d.h. Änderungen über das LCP-Display sind sofort aktiv.


ACHTUNG!

Wenn der Parameterwert außerhalb des durch min- and max-Wert definierten Bereiches ist, wird der Befehl weder dargestellt noch korrekt ausgeführt!

Befehlsgruppe

PAR

Querverweise

LINKGPAR, SET, GET
 Benutzerparameter, Parameter-Referenz

Syntax-Beispiel

LINKAXPAR POSERR 712 "Schleppfehler" 300 50000 0

■ LINKGPAR

Mit LINKGPAR können Sie einen vordefinierten, globalen Parameter oder einen freien internen Benutzerparameter mit dem LCP-Menü verknüpfen.

Danach können Sie über das LCP den Parameter verändern oder lesen.

Wird ein verknüpfter Parameter mit einem SET Befehl verändert, wird er automatisch auch an das LCP übergeben, wirkt jedoch nur temporär.

Wenn der Anwender einen verknüpften Parameter am LCP ändert, wird der neue Wert ausgeführt. Aber erst, wenn dieser Wert mit OK bestätigt wird, wird er permanent als Benutzerparameter im EEPROM gespeichert.

NEU: Der Befehl LINKGPAR prüft, ob der Wert des Benutzerparameters innerhalb des vorgegebenen Bereiches liegt. Falls nicht, wird das entsprechende Limit verwendet und dieser Wert gespeichert. Auf diese Weise wird sichergestellt, dass im Display eine Anzeige erscheint.

Kurzinfo

Globalen Parameter mit LCP-Display verknüpfen

Syntax

LINKGPAR param optpar "text" min max typ

Parameter

param = Globaler Parametername oder Benutzerparameternummer (130–229)
 optpar = LCP-Parameter Nummer (710–779 und 795–799)
 text = beschreibender Text für Display
 min = minimaler Wert, den der Parameter annehmen darf
 max = maximaler Wert, den der Parameter annehmen darf
 typ = Parameter Art
 0 = offline, d.h. Änderungen werden erst durch die Bestätigung mit OK aktiv.
 1 = online, d.h. Änderungen über das LCP-Display sind sofort aktiv.


ACHTUNG!

Wenn der Parameterwert außerhalb des durch min- and max-Wert definierten Bereiches ist, wird der Befehl weder dargestellt noch korrekt ausgeführt!

Querverweise

LINKAXPAR, SET, GET
 Benutzerparameter, Parameter-Referenz

Syntax-Beispiel

```
LINKGPAR I_ERRCLR 710 " " 0 33 0
/* Vordefinierte globale Parameter */
LINKGPAR 132 711 "name" 0 100000 0
/* freie interne Benutzerparameter */
LINKGPAR PRGPAR 701 "Autoprogramm" -1 10 0
/* Programm für den Autostart festlegen */
```

■ LINKSYSVAR

Der Befehl LINKSYSVAR verknüpft die Systemvariable SYSVAR[indx] mit dem VLT-Parameter *vltnr* (795-799) und dem Anzeigetext *text*. Auf diese Weise können Sie interne Werte ohne Umweg über LINKGPAR auf dem Display darstellen.

Wenn Sie zum Beispiel mit #DEBUG NOSTOP compilieren, verknüpfen Sie die interne Zeilennummer mit dem VLT-Parameter 795. Dann können Sie die Programmausführung ganz gezielt beobachten.

Kurzinfo

Systemvariable mit LCP-Display verknüpfen

Syntax

```
LINKSYSVAR indx vltnr "text"
```

Parameter

indx = Index der Systemvariable SYSVAR
 vltnr = LCP-Parameter Nummer (795...799)
 text = beschreibender Text für Display



ACHTUNG!

Wenn der Parameterwert außerhalb des durch min- und max-Wert definierten Bereiches ist, wird der Befehl weder dargestellt noch korrekt ausgeführt!



ACHTUNG!

Alle 40 ms findet ein Update des Parameters *vltnr* statt. Wenn also auf diese Weise fünf Parameter verknüpft werden, dauert es mindestens 200 ms, bis derselbe Parameter wieder erneuert wird.

Befehlsgruppe

PAR

Querverweise

LINKGPAR, SYSVAR
 Benutzerparameter, Parameter-Referenz

Syntax-Beispiele

```
LINKSYSVAR 33 795 "interne Zeilennr."
LINKSYSVAR 30 796 "Motorspannung"
/* Mit Benutzerparameter 11 und 12 festlegen:
Zeilennummer in erster Displayzeile und
Motorspannung in zweiter Displayzeile anzeigen. */
```

■ LOOP

Mit dem LOOP Befehl können Sie einen bestimmten Programmbereich ein- oder mehrmals wiederholen. Die Anzahl der Schleifenwiederholungen kann dabei als absoluter Wert oder auch in Form einer Variablen angegeben werden.

Die Programmposition, zu der gesprungen werden soll, ist durch ein Label gekennzeichnet. Ein Label kann aus einem oder mehreren Zeichen bestehen und darf nicht mit einem Variablennamen oder einem Befehlswort identisch sein. Ein Label muss zudem eindeutig sein, das heißt das gleiche Label darf nicht mehrfach für unterschiedliche Programmpositionen verwendet werden.

Kurzinfo

Definierte Schleifenwiederholung

Syntax

```
LOOP n label
```

Parameter

n = Anzahl der Schleifenwiederholungen
 label = Kennung der Programmzielposition



ACHTUNG!

Das Label muss an der Programmzielposition mit einem Doppelpunkt abgeschlossen werden, z.B. schleife:

Da der interne Schleifenzähler erst am Schleifenende überprüft und danach verringert wird, werden die Befehle innerhalb der Schleife insgesamt einmal mehr als in dem entsprechenden Übergabewert angegeben ausgeführt.

Befehlsgruppe

CON

Querverweise

GOTO, WHILE...ENDWHILE, REPEAT...UNTIL

Syntax-Beispiel

```
schleife: /* Label zu dem gesprungen wird */
Befehlszeile 1
Befehlszeile n
LOOP 9 schleife:
/* Schleifeninhalt 10-mal wiederholen */
```

Programmbeispiele

```
LOOP_01.M
APOS_01.M, IN_01.M, MOTOR_01.M, NOWAI_01.M
```

■ MAPOS

Mit MAPOS können Sie die aktuelle Masterposition (absolut zum aktuellen Nullpunkt) abfragen.

Kurzinfo

Aktuelle Istposition des Masters abfragen

Syntax

erg = MAPOS

Rückgabewert

erg = Masterposition absolut zum aktuellen Nullpunkt in qc

Befehlsgruppe

I/O

Querverweise

CPOS, DEF ORIGIN, SET ORIGIN, POSA, POSR
Parameter: POSFACT_Z (23), POSFACT_N (26)

Syntax-Beispiel

PRINT MAPOS

/* aktuelle Masterposition am PC ausgeben */

■ MAVEL

Diese Funktion liefert die aktuelle Geschwindigkeit des Master-Antriebes in qc/sec zurück, wobei sich qc auf den Master-Encoder bezieht.

Die Genauigkeit der Werte hängt von der Messdauer (Mittelung) ab. Diese ist standardgemäß auf 20 ms eingestellt, kann aber vom Anwender mit dem _GETVEL Befehl verändert werden. Es genügt den Befehl einmal aufzurufen, um von da an mit einer anderen Messzeit zu arbeiten. So stellt der Befehl:

```
var = _GETVEL 100
```

die Messdauer auf 100 ms ein, so dass man bei MAVEL eine wesentlich bessere Auflösung der Geschwindigkeit erhält, schnelle Änderungen dagegen erst mit einer Verzögerung von maximal 100 ms.

Kurzinfo

Aktuelle Geschwindigkeit des Masters abfragen

Syntax

erg = MAVEL

Rückgabewert

erg = aktuelle Geschwindigkeit des Master-Antriebes in qc/sec; Wert mit Vorzeichen

Befehlsgruppe

I/O

Querverweise

AVEL

Syntax-Beispiel

PRINT MAVEL

/* aktuelle Master-Geschwindigkeit am PC ausgeben */

■ MIPOS
Kurzinfo

Letzte Index- bzw. Markerposition des Masters abfragen

Syntax

erg = MIPOS

Rückgabewert

erg = letzte Index- bzw. Markerposition des Masters absolut zum aktuellen Nullpunkt in qc

Beschreibung

Diese Funktion liefert die letzte Index- bzw. Markerposition des Masters absolut zum aktuellen Nullpunkt in qc zurück.

Die Konfiguration von MIPOS, d.h. ob die Index- oder Markerposition des Master-Encoders zurückgeliefert wird, erfolgt über den Parameter SYNCMTYPM.


ACHTUNG!

Das Triggersignal für die Markerposition muss dabei zwingend an den Eingang 1 angeschlossen werden.

Der Positionswert in MIPOS ist auf +/- 1qc genau. Im Gegensatz zu der Positionsinformation in MAPOS, welche nur im Reglerzyklus von typisch 1ms aktualisiert wird, wird der aktuelle Positionswert hardware-mäßig beim Auftreten des konfigurierten Signals (in einem internen Prozessorregister) in Echtzeit zwischengespeichert und dann in die Systemvariable MIPOS kopiert.

Falls gleichzeitig zur Markerposition ein Interrupt ausgelöst wird (ON INT 1 GOSUB ...) und in diesem Interrupt mit MIPOS gearbeitet wird, sollte im Interrupt-Unterprogramm eine Verzögerung von 2 Millisekunden (DELAY 2) vor dem Lesen von MIPOS verwendet werden. So kann sichergestellt werden, dass der gelatchte Positionswert bereits sicher in die Systemvariable MIPOS kopiert ist und nicht noch auf einen veralteten Wert zurückgegriffen wird. Siehe Beispiel.

Befehlsgruppe

I/O

Querverweise

CPOS, DEF ORIGIN, SET ORIGIN, POSA, POSR,, SYNCMTYPM

Parameter: POSFACT_Z (23), POSFACT_N (26)

Syntax-Beispiel

```
PRINT MIPOS
/* letzte Indexposition des Masters am PC
ausgeben */
```

Beispiel

```
// Definition Interrupt-Handler
ON INT 1 GOSUB master_int
// Definition von IPOS-Latching auf positive Flanke
// an Eingang 1
SET SYNCMTYPM 2
```

```
// Bewegung starten
```

```
CVEL 10
CSTART x(1)
// Endlos-Loop
mainloop:
// ...
GOTO mainloop
```

SUBMAINPROG

```
SUBPROG master_int
// Testweise mapos zwischenspeichern
// um zu sehen, wie genau dies wäre ...
int_mpos = MAPOS
// 2 Millisekunden warten, damit MIPOS
// sicher aktualisiert ist
DELAY 2
// IPOS zwischenspeichern für spätere
// Bearbeitung, etc.
triggered_mpos = MIPOS
// ...
// ...
PRINT „Interrupt master position: „,int_mpos
PRINT „Triggered master position:
„,triggered_mpos
RETURN
ENDPROG
```

■ MOTOR OFF

Der Befehl MOTOR OFF schaltet die Motorregelung ab. Nach einem MOTOR OFF kann, sofern keine Motorbremse vorhanden ist, die Antriebsachse frei bewegt werden. Eine Überwachung der aktuellen Position findet weiterhin statt, das heißt es kann auch nach einem MOTOR OFF die aktuelle Position (APOS) abgefragt werden.



ACHTUNG!

Um einen Bewegungsvorgang nach MOTOR OFF wieder zu starten, muss der Befehl MOTOR ON verwendet werden. Nur der Befehl ERRCLR aktiviert MOTOR ON automatisch.

Kurzinfo

Ausschalten der Motorregelung

Syntax

MOTOR OFF

Befehlsgruppe

INI

Querverweise

MOTOR ON

Syntax-Beispiel

MOTOR OFF

/* Lageregelung der Achse abschalten */

Programmbeispiele

MOTOR_01.M

POS_01.M

■ MOTOR ON

Der MOTOR ON Befehl schaltet die Motorregelung nach einem vorausgegangenen MOTOR OFF wieder ein. Beim Ausführen des MOTOR ON wird die Sollposition als aktuelle Position gesetzt, das heißt der Motor verharnt lagegeregelt auf der aktuellen Position.

Dabei wird der Schleppfehler automatisch bei der Ausführung von MOTOR ON zurückgesetzt.

Kurzinfo

Einschalten der Motorregelung

Syntax

MOTOR ON



ACHTUNG!

Der MOTOR ON Befehl ist nicht geeignet, die nach einem Fehler abgeschaltete Lageregelung wieder zu aktivieren. Dazu muss der ERRCLR Befehl verwendet werden.

Befehlsgruppe

INI

Querverweise

MOTOR OFF

Syntax-Beispiel

MOTOR ON

/* Lageregelung der Achse einschalten */

Programmbeispiele

MOTOR_01.M

POS_01.M

■ MOTOR STOP

Mit dem MOTOR STOP Befehl wird ein im Positionier-Drehzahl- oder Synchronisationsmodus fahrender Antrieb mit der zuletzt programmierten Beschleunigung abgebremst und auf der aktuellen Position lagegeregelt.

Ein mit MOTOR STOP abgebremster Antrieb kann zu einem späteren Zeitpunkt mit dem Befehl CONTINUE seinen ursprünglichen Bewegungsablauf wieder aufnehmen. (Ausnahme: CONTINUE setzt keine abgebrochenen Synchronisationsbefehle fort.)


ACHTUNG!

Wenn MOTOR STOP in einem Unterprogramm ausgeführt wird oder wenn NOWAIT ON gesetzt ist, werden während der Abarbeitung von MOTOR STOP schon die nächsten Programmzeilen abgearbeitet; der Bremsvorgang läuft im Hintergrund.

Um den Antrieb auf Drehzahl Null abzubremsen ist daher sicherzustellen, dass während des Bremsens kein neuer Fahrbefehl gesetzt wird.

Kurzinfo

Stoppen des Antriebs

Syntax

MOTOR STOP

Befehlsgruppe

CON

Querverweise

POSA, CSTART, CONTINUE, CSTOP

Syntax-Beispiel

MOTOR STOP

/* Bewegungsvorgang der Achse unterbrechen */

Programmbeispiele

MSTOP_01.M

■ MOVESYNCORIGIN

Der Befehl verschiebt den Synchronisationsursprung bezüglich des Masters. Während SET SYNCPOSOFFS den Positionsoffset absolut setzt, bezieht sich MOVESYNCORIGIN immer auf den letzten und verschiebt den Positionsoffset relativ. Wenn Sie den Positionsoffset ständig verschieben müssen, können Sie so zu große Zahlen bzw. einen Überlauf verhindern.

Kurzinfo

Synchronisationsursprung relativ verschieben

Syntax

MOVESYNCORIGIN mwert

Parameter

mwert = Relativer Offset in Bezug zum Master in qc

Value range = -MLONG / SYNCFACTS (50) ... MLONG / SYNCFACTS (50)


ACHTUNG!

Gültig für Positionssynchronisation SYNCPOS und Positionssynchronisation mit Markerkorrektur SYNCM.

Befehlsgruppe

SYN

Querverweise

SET, SYNCPOSOFFS (54)

Syntax-Beispiel

MOVESYNCORIGIN x(1) 1000

■ NOWAIT

Mit dem NOWAIT Befehl wird das Verhalten nach Positionierbefehlen definiert.
 Wenn NOWAIT OFF eingestellt ist, werden Positionierbefehle vollständig, das heißt bis zur Zielposition ausgeführt, bevor mit der Abarbeitung der folgenden Befehle begonnen wird.
 Bei NOWAIT ON wird nach dem Starten eines Positionierbefehls sofort die weitere Befehlsabarbeitung fortgesetzt und der Positioniervorgang läuft gleichsam im Hintergrund ab. Im Zustand NOWAIT ON ist es somit auch möglich, während eines Positioniervorganges die aktuelle Position abzufragen, die Geschwindigkeit oder die Zielposition zu ändern.

Kurzinfo

Nach Positionierbefehlen warten / nicht warten

Syntax

NOWAIT s

Parameter

s = Zustand
 ON = nicht Warten bis Zielposition erreicht
 OFF = Warten bis Position erreicht



ACHTUNG!

Der Default-Zustand ist NOWAIT OFF. Wenn also innerhalb eines Programmes keine NOWAIT ON Anweisung enthalten ist, werden Positioniervorgänge vollständig ausgeführt, bevor mit der Abarbeitung des nächsten Befehls begonnen wird. Wenn im Zustand NOWAIT ON während eines noch aktiven Positioniervorganges ein weiterer Positionierbefehl erfolgt, wird ohne Unterbrechung und ohne voriges Anfahren der ersten Zielposition sofort auf die neue Endposition verfahren.
 Sowohl der HOME, wie auch der INDEX Befehl werden im Zustand NOWAIT ON zu Ende abgearbeitet, bevor der nächste Befehl ausgeführt wird.

Befehlsgruppe

CON

Querverweise

WAITAX, AXEND, POSA, POSR, HOME, INDEX

Syntax-Beispiel

NOWAIT ON
 /* Nach Positionierbefehlen nicht warten */
 NOWAIT OFF
 /* Warten nach Pos.-Befehlen bis Ziel erreicht */

Programmbeispiele

NOWAI_01.M, MSTOP_01.M, OUT_01.M, VEL_01.M

■ ON APOS .. GOSUB

Mit der Anweisung ON APOS kann man ein Unterprogramm aufrufen, wenn eine bestimmte Slave-Position (BE) in positiver bzw. negativer Richtung passiert wurde. Die Anweisung kann für Positionier- und Synchronisiersteuerungen wie auch für Kurvenscheibensteuerungen und Nockenschaltwerke nützlich sein. Zum Beispiel um bei offenen Kurven die anwachsende Slave-Position nach jedem Zyklus durch einen wiederkehrenden Bezugspunkt zu ersetzen.

Kurzinfo

Unterprogramm aufrufen, wenn die Slave-Position xxx passiert wurde

Syntax

ON sign APOS xxx GOSUB name

Parameter

sign = + = wenn die Slave-Position in positiver Richtung passiert wurde
 - = wenn die Slave-Position in negativer Richtung passiert wurde
 xxx = Slave-Position (BE)
 name = Name des Unterprogrammes

100 xxx 0
 <- - - - - positive Richtung
 - - - - - > negative Richtung

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb eines Programmbereiches SUBMAINPROG und ENDPROG stehen.



ACHTUNG!

Ein ON APOS Interrupt kann mit dem Befehl ON DELETE .. GOSUB .. gelöscht werden.

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN, DISABLE ENABLE, Prioritäten der Interrupts, ON DELETE .. GOSUB

Syntax-Beispiel

ON -APOS 800 GOSUB name
 // Unterprogramm *name* aufrufen, wenn die
 // Slave-Position 800 in negativer Richtung
 // passiert wurde

■ ON COMBIT .. GOSUB

Mit der Anweisung ON COMBIT wird ein Unterprogramm aufgerufen, wenn Bit n des Kommunikationspuffers gesetzt wird.

Kurzinfo

Unterprogramm aufrufen, wenn Bit n des Kommunikationspuffers gesetzt wird.

Syntax

ON COMBIT n GOSUB name

Parameter

n = Bit n des Kommunikationspuffers
 $-32 \leq n \leq 32, n \neq 0$
 name = Name des Unterprogrammes



ACHTUNG!

ON COMBIT bezieht sich auf die ersten 32 Bit des Prozessdatenspeichers.

Priorität

Sollten mehrere Interrupts gleichzeitig auftreten, wird zuerst das dem niedrigsten Bit zugeordnete Unterprogramm abgearbeitet. Die anderen Interrupts werden anschließend abgearbeitet. Tritt während eines Interrupt-Unterprogrammes der gleiche Interrupt auf (Ausnahme: Fehler-Interrupt), wird dieser ignoriert und geht verloren.

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPROG gekennzeichneten Programmbereichs definiert sein.



ACHTUNG!

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Portabilität

Bei COMOPTGET und COMOPTSEND wird der Offset von 2 Word aus Kompatibilitätsgründen beibehalten.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN, COMOPTGET, COMOPTSEND

Syntax-Beispiel

```
ON COMBIT 5 GOSUB test
// Interrupt auf bit 5 des Feldbusses setzen
```

■ ON DELETE .. GOSUB

Kurzinfo

Löscht einen Positions-Interrupt: ON APOS, ON MCPOS oder ON MAPOS

Syntax

ON DELETE pos GOSUB name

Parameter

pos = Wert
 name = Name des Unterprogramms

Beschreibung

Der Befehl kann genutzt werden, um einen ON APOS Interrupt zu löschen, der zuvor wie folgt definiert wurde:

```
ON sign APOS xxx GOSUB name
```

Der Parameter 'pos' kann jeden Wert annehmen, z.B. 0. Der Wert wird nicht geprüft und hat keine Bedeutung für das Löschen des Interrupts.

Die Hauptbedeutung kommt dem Parameter 'name' zu, der den Namen des Unterprogramms enthalten muss, das vorher im ON APOS Befehl definiert worden ist. Daher löscht das 'ON DELETE pos GOSUB name' jeden (!) Positions-Interrupt, der zum Unterprogramm gehört und durch den Namen erkannt wird. Siehe Beispiel 1.



ACHTUNG!

Es werden nur Positions-Interrupts gelöscht, keine anderen Interrupttypen.

Umleiten eines ON ... APOS ... GOSUB Befehls

Ein Positions-Interrupt kann in ein anderes Unterprogramm umgeleitet werden. Dies definiert keinen neuen Interrupt, sondern modifiziert nur das Unterprogramm, das im Fall einer Interrupt-Erkennung ausgeführt werden muss.

Die Befehlssyntax ist die gleiche wie für den ON APOS Befehl:

```
ON sign APOS xxx GOSUB newname
```

Die Parameter 'sign' und 'xxx' müssen genau die gleichen sein, wie in der ursprünglichen Definition. Die Position, die es betrifft wird durch diese zwei Parameter identifiziert. Der Parameter 'newname' muss den aktualisierten Namen des Unterprogramms enthalten, das aufgerufen werden soll, wenn der Interrupt eintritt. Siehe Beispiel 2.



ACHTUNG!

Es können nur Positions-Interrupts umgeleitet werden, keine anderen Interrupttypen.

Befehlsgruppe

INT

Querverweise

ON APOS .. GOSUB, ON MAPOS .. GOSUB, ON MCPOS .. GOSUB

Programmbeispiel 1

```
ON - APOS 20000 GOSUB hitinfo // Interrupt #1
ON - APOS 10000 GOSUB hitinfo // Interrupt #2
ON + APOS 10000 GOSUB hitinfo // Interrupt #3
ON + APOS 0 GOSUB hitzero // Interrupt #4
ON - APOS 0 GOSUB hitzero // Interrupt #5
ON INT 3 GOSUB hitinfo // Interrupt #6
```

...

ON DELETE 0 GOSUB hitinfo

...

```
ON + APOS 99999 GOSUB hitinfo
// Neuer definierter Positions-Interrupt
```

Ergebnis:

Alle Positions-Interrupts (#1, #2, #3), die zu dem Unterprogramm hitinfo gehören, werden gelöscht sobald 'ON DELETE 0 GOSUB hitinfo' ausgeführt wird. Diese Interrupts zählen nicht mehr für die maximale Anzahl der verfügbaren Interrupts und können daher auch nicht mehr freigegeben oder gesperrt werden. Alle anderen nicht-Positions-Interrupts, sogar diejenigen, die zum gleichen Unterprogramm gehören (z.B. ON INT 3) sind weiterhin gültig!

Sobald die Befehlszeile 'ON + APOS 99999 GOSUB hitinfo' ausgeführt wird, definiert dies einen neuen Positions-Interrupt, der zum dem Unterprogramm verweist das bereits früher in Gebrauch war.

Programmbeispiel 2

```
ON - APOS 10000 GOSUB hitinfo // Interrupt #1
ON + APOS 10000 GOSUB hitinfo // Interrupt #2
```

...

```
ON + APOS 10000 GOSUB hitposdir
// Interrupt #2 umleiten
```

Ergebnis:

Sobald die zweite Definition des 'ON + APOS 10000 ...' Befehls ausgeführt ist, wird der Interrupt #2 zu einem neu definierten Unterprogramm 'hitposdir' umgeleitet. Es ist nach wie vor der gleiche Interrupt (d.h. kein weiterer), der nun ein anderes Unterprogramm aufruft. Die „alte“ Definition des Interrupts #1 'ON - APOS 10000 GOSUB hitinfo' ist weiterhin ohne jede Veränderung gültig.

■ ON ERROR GOSUB

Mit der ON ERROR GOSUB Anweisung wird ein Unterprogramm definiert, das bei einem Fehler aufgerufen wird. Tritt nach dieser Definition zu einem beliebigen Zeitpunkt ein Fehler auf, wird das Programm nicht automatisch abgebrochen, sondern das definierte Unterprogramm aufgerufen.

Innerhalb dieses Unterprogrammes ist es dann möglich, gezielt auf den Fehler zu reagieren, auf Benutzereingriffe zu warten, die Fehlermeldung mit ERRCLR zu löschen oder bei nicht behebbaren Fehlern mit der EXIT Anweisung das Programm abzubrechen.

Wird das Programm nicht abgebrochen, dann wird nach dem RETURN Befehl an der vor dem Aufruf des Fehlerunterprogrammes bearbeiteten Programmposition fortgefahren.

Mit dem CONTINUE Befehl kann man den durch den Fehler unterbrochenen Bewegungsablauf fortsetzen (Ausnahme: Synchronisationsbefehle).

Kurzinfo

Definition eines Unterprogrammes für den Fehlerfall

Syntax

ON ERROR GOSUB name

Parameter

name = Name des Unterprogrammes

Besonderheiten

Die ON ERROR GOSUB Anweisung sollte am Programmanfang stehen, damit sie für das ganze Programm gültig ist.

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPORG gekennzeichneten Programmbereichs definiert sein.

Das Erkennen eines Interrupts und der Aufruf des entsprechenden Unterprogrammes benötigt maximal 2 Millisekunden.



ACHTUNG!

Fehlerunterprogramme können nicht durch andere Interrupts unterbrochen werden.



ACHTUNG!

Während der Ausführung eines Fehlerunterprogrammes ist automatisch NOWAIT ON gesetzt.

Wird das Fehlerunterprogramm mit einem Fehler verlassen, weil zum Beispiel kein ERRCLR durchgeführt wurde oder bereits ein neuer Fehler aufgetreten ist, erfolgt ein erneuter Aufruf.


ACHTUNG!

Die ON ERROR GOSUB Anweisung beendet nicht HOME- und INDEX-Befehle. Das heißt, diese werden zu Ende ausgeführt, sobald der Fehler gelöscht ist.

Um dies zu verhindern, kann man ein ON TIME 1 in die ERROR-Anweisung einfügen.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN, ERRCLR, ERRNO, CONTINUE, EXIT

Syntax-Beispiel

```
ON ERROR GOSUB errhandle
/* Definition eines Fehlerunterprogrammes */
  Befehlszeile 1
  Befehlszeile n
SUBMAINPROG
/* Unterprogramm errhandle muss definiert sein */
SUBPROG errhandle
  Befehlszeile 1
  Befehlszeile n
RETURN
ENDPROG
```

Programmbeispiele

ERROR_01.M
IF_01.M, INDEX_01.M

■ ON INT .. GOSUB

Mit der Anweisung ON INT GOSUB wird ein Unterprogramm definiert, das aufgerufen wird, wenn an dem überwachten Eingang eine Pegeländerung auftritt.

Pro Eingang kann maximal ein Unterprogramm definiert werden. (Es ist nicht möglich, für die fallende und für die steigende Flanke des gleichen Eingangs Interrupts zu definieren.) Tritt nach dieser Definition zu einem beliebigen Zeitpunkt der entsprechende Interrupt auf, wird das zugehörige Unterprogramm aufgerufen und abgearbeitet. Nach dem letzten Unterprogrammbefehl (RETURN) wird das Programm an der vor dem Interrupt bearbeiteten Programmposition fortgesetzt.

Kurzinfo

Definition eines Interrupt-Einganges

Syntax

ON INT n GOSUB name

Parameter

n = Nummer des zu überwachenden Eingangs (Bereich -8 ... 8 und VLT Eingänge 16 ... 33 und -33 ... -16)
 positive Eingangsnummer (1 ... 8) = Reaktion auf steigende Flanke
 negative Eingangsnummer (-8 ... 8) = Reaktion auf fallende Flanke
 name = Name des Unterprogrammes

Priorität

Sollten mehrere Interrupts gleichzeitig auftreten, wird zuerst das dem kleinsten Eingang der aufgetretenen Interrupts zugeordnete Unterprogramm abgearbeitet. Die anderen Interrupts werden anschließend abgearbeitet. Tritt während eines Interrupt-Unterprogrammes der gleiche Interrupt auf (Ausnahme: Fehler-Interrupt), wird dieser ignoriert und geht verloren.

Besonderheiten

Die ON INT .. GOSUB Anweisung sollte am Programmfang stehen, damit sie für das ganze Programm gültig ist.

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPROG gekennzeichneten Programmbereichs definiert sein.

Das Erkennen eines Interrupts und der Aufruf des entsprechenden Unterprogrammes benötigt maximal 2 Millisekunden. Ein Interrupt vom VLT-Eingang addiert im schlimmsten Fall zusätzlich 2 ms.

Es ist eine minimale Signallänge von 1 ms notwendig, um eine Pegeländerung sicher zu erkennen. Informieren Sie sich bitte im Kapitel Ein- und Ausgangsklemmen über die Beschaltung und technischen Daten der Eingänge.



ACHTUNG!

Die Anweisung ON INT GOSUB ist flanken- und nicht pegelgesteuert.



ACHTUNG!

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN, ON ERROR GOSUB

Syntax-Beispiel

```
ON INT 4 GOSUB posin
/* Definition von Eingang 4 (positive Flanke) */
ON INT -5 GOSUB negin
/* Definition von Eingang 5 (negative Flanke) */
  Befehlszeile 1
  Befehlszeile n
SUBMAINPROG
/* Unterprogramme müssen definiert sein */
SUBPROG posin
  Befehlszeile 1
  Befehlszeile n
RETURN
SUBPROG negin
  Befehlszeile 1
  Befehlszeile n
RETURN
ENDPROG
```

Programmbeispiele

ONINT_01.M, DELAY_01.M

■ ON MAPOS .. GOSUB

Mit der Anweisung ON MAPOS kann man ein Unterprogramm aufrufen, wenn eine bestimmte Master-Position (MU) in positiver bzw. negativer Richtung passiert wurde. Zum Beispiel um bei einem Linearantrieb (Slave) mit einem Verfahrbereich von 0 bis 10000 BE an einer beliebigen Position einen Ausgang zu setzen.

Kurzinfo

Unterprogramm aufrufen, wenn die Master-Position xxx (MU) passiert wurde

Syntax

ON sign MAPOS xxx GOSUB name

Parameter

sign = + = wenn die Master-Position in positiver Richtung passiert wurde
 - = wenn die Master-Position in negativer Richtung passiert wurde
 xxx = Master-Position in MU
 name = Name des Unterprogrammes

100	xxx	0
<- - - - -	positive Richtung	
- - - - -	>	negative Richtung

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb eines Programmbereiches SUBMAINPROG und ENDPROG stehen.



ACHTUNG!

Ein ON MAPOS Interrupt kann mit dem Befehl ON DELETE .. GOSUB .. gelöscht werden.



ACHTUNG!

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN, DISABLE ENABLE.., Priorität der Interrupts, ON DELETE .. GOSUB

Syntax-Beispiel

```
ON +MAPOS 1200 GOSUB name
// Unterprogramm name aufrufen, wenn die
// Slave-Position 1200 in positiver Richtung
// passiert wurde
```

■ ON MCPOS .. GOSUB

Mit dieser für Kurvenscheibensteuerungen typischen Anweisung ON MCPOS kann man ein Unterprogramm aufrufen, wenn eine bestimmte Master-Position (MU) in positiver bzw. negativer Richtung passiert wurde. Auf diese Weise lassen sich nicht nur Nockenschaltwerke realisieren, sondern auch viel komplexere Aufgaben durchführen. Zum Beispiel könnte man abhängig von der Position online Parameter ändern.

Kurzinfo

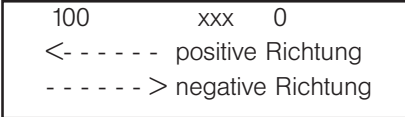
Unterprogramm aufrufen, wenn die Master-Position xxx (MU) passiert wurde

Syntax

ON sign MCPOS xxx GOSUB name

Parameter

sign = + = wenn die Master-Position in positiver Richtung passiert wurde
 - = wenn die Master-Position in negativer Richtung passiert wurde
 xxx = Master-Position in MU
 name = Name des Unterprogramms



ACHTUNG!

Vor dem Befehl ON MCPOS .. GOSUB muss immer ein DEFMCPOS oder ein SETCURVE stehen, weil sonst die Kurvenposition nicht bekannt ist.

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb eines Programmbereiches SUBMAINPROG und ENDPROG stehen.



ACHTUNG!

Ein ON MCPOS Interrupt kann mit dem Befehl ON DELETE .. GOSUB .. gelöscht werden.



ACHTUNG!

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN, DISABLE ENABLE.., Priorität der Interrupts, ON DELETE .. GOSUB

Syntax-Beispiel

```
ON +MCPOS 1200 GOSUB parameter
    // Unterprogramm parameter immer an
    // Position 1200 aufrufen
```

Beispiel

Auf einem Band werden Kartons unregelmäßig transportiert. Durch Setzen eines Ausgangs wird der Slave immer dann gestartet, wenn die Position xxx erreicht wird.

```
ON +MCPOS 4500 GOSUB ausgang
    // Unterprogramm ausgang immer an
    // Position 4500 aufrufen
SUBMAINPROG
    // Unterprogramm ausgang setzen
SUBPROG ausgang
    OUT 3 1// 03 an
RETURN
ENDPROG
```

■ ON PARAM .. GOSUB

Mit der Anweisung ON PARAM kann man reagieren, wenn Parameter über das VLT-Display geändert werden und ein Unterprogramm aufrufen.

Kurzinfo

Unterprogramm aufrufen, wenn sich ein Benutzer-Parameter ändert.

Syntax

ON PARAM n GOSUB name

Parameter

n = 701 ... 779
name =Name des Unterprogrammes

Priorität

Tritt während eines Interrupt-Unterprogrammes der gleiche Interrupt auf (Ausnahme: Fehler-Interrupt), wird dieser ignoriert und geht verloren.

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPROG gekennzeichneten Programmbereichs definiert sein.

Es sind maximal 10 ON PARAM Funktionen möglich.



ACHTUNG!

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN

Syntax-Beispiel

```
LINKXPAR POSERR 712 "Schleppfehler" 300 5000 0
ON PARAM 712 GOSUB schlepp
// Wenn max. Schleppfehler geändert wird
SUBMAINPROG
SUBPROG schlepp
PRINT "Neuer Schleppfehler: ", GET POSERR
RETURN
```

■ ON PERIOD

Mit ON PERIOD kann man ein Unterprogramm in regelmäßigen Abständen (zeitgeführt) aufrufen. ON PERIOD wirkt wie ein Interrupt und wird alle 20 ms überprüft.

Kurzinfo

Ruft ein Unterprogramm in regelmäßigen Abständen auf

Syntax

ON PERIOD n GOSUB name

Parameter

n > 20 ms = Zeit in ms, nach der das Programm immer wieder aufgerufen wird (maximal MLONG)
n = 0 = Funktion abschalten



ACHTUNG!

Die Genauigkeit mit der die Zeit eingehalten wird, hängt vom restlichen Programm ab. Typischerweise ist die Genauigkeit ± 1 ms.

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPROG gekennzeichneten Programmbereichs definiert sein.



ACHTUNG!

Während der Ausführung eines ON PERIOD Unterprogrammes ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

ON TIME, GOSUB

■ ON STATBIT .. GOSUB

Mit der Anweisung ON STATBIT wird ein Unterprogramm aufgerufen, wenn Bit n des VLT-Status gesetzt ist. Diese 32 Bit des VLT-Status setzen sich aus dem VLT-Statuswort, dem Byte 2 des internen Status (zum Beispiel „Achse fährt“) und dem Bit n von SYNCSTAT zusammen.

Kurzinfo

Unterprogramm aufrufen, wenn Bit n des VLT-Status gesetzt wird.

Syntax

ON STATBIT n GOSUB name

Parameter

n = Bit n des VLT-Status
 Bit 1...16 VLT Statuswort
 Bit 17...24 Byte 2 des internen Status:
 Bit 17 = Achse fährt
 Bit 18 = Überlauf Slave Drehgeber
 Bit 19 = Überlauf Master Drehgeber
 Bit 20 = Lageregelung ist temporär
 abgeschaltet *)
 Bit 25...32 SYNCSTAT
 Bit 25 = SYNCREADY
 Bit 26 = SYNCFAULT
 Bit 27 = SYNCACCURACY
 Bit 28 = SYNCMMHIT
 Bit 29 = SYNCSTMHIT
 Bit 30 = SYNCMMERR
 Bit 31 = SYNCSTMERR

name = Name des Unterprogrammes

*) Erläuterung: d.h. die Achse befindet sich innerhalb des Toleranzbereiches des Regelfensters REGWINMAX / REGWINMIN. Sobald das Regelfenster verlassen wird, wird die Lageregelung wieder eingeschaltet.

Priorität

Sollten mehrere Interrupts gleichzeitig auftreten, wird zuerst das dem niedrigsten Bit zugeordnete Unterprogramm abgearbeitet. Die anderen Interrupts werden anschließend abgearbeitet. Tritt während eines Interrupt-Unterprogrammes der gleiche Interrupt auf (Ausnahme: Fehler-Interrupt), wird dieser ignoriert und geht verloren.

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPROG gekennzeichneten Programmbereichs definiert sein.


ACHTUNG!

Während der Ausführung von Unterprogrammen, die durch einen Interrupt ausgelöst wurden, ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

SUBPROG...RETURN

Syntax-Beispiel

```
ON STATBIT 30 GOSUB markererror
/* Interrupt, wenn Fehlerflag Master */
SUBMAINPROG
    SUBPROG markererror
    SYNCSTATCLR 32
    /* Fehlerflag SYNCMMERR löschen */
    /* wert 32 von Parameter SYNCSTATCLR */
    /* nicht Bit-Nr.! */
RETURN
ENDPROG
```

■ ON TIME

Nach Ablauf der gesetzten Zeit wird das entsprechende Unterprogramm einmal aufgerufen. Das Programm läuft in der Zwischenzeit normal weiter.

Kurzinfo

Einmalaufruf eines Unterprogrammes

Syntax

ON TIME n GOSUB name

Parameter

n = Zeit in ms, nach der das Unterprogramm aufgerufen wird (maximal MLONG)
name = Name des Unterprogrammes



ACHTUNG!

Die Genauigkeit mit der die Zeit eingehalten wird, hängt vom restlichen Programm ab. Ein typischer Wert ist eine Genauigkeit von ± 1 ms.

Besonderheiten

Das aufzurufende Unterprogramm muss innerhalb des durch SUBMAINPROG und ENDPORG gekennzeichneten Programmbereichs definiert sein.



ACHTUNG!

Während der Ausführung eines ON TIME Unterprogrammes ist automatisch NOWAIT ON gesetzt.

Befehlsgruppe

INT

Querverweise

ON PERIOD, GOSUB

Syntax-Beispiel

```
OUT 1 1      /* Lampe an */
ON TIME 200 GOSUB aus1
/* Lampe nach 200 ms wieder ausschalten */
SUBMAINPROG
SUBPROG aus1
OUT 1 0
RETURN
ENDPROG
```

■ OUT

Es können sowohl die 8 digitalen Ausgänge der SyncPos-Option als auch die 2 Relaisausgänge des VLT5000 mit dem OUT Befehl gesetzt und zurückgesetzt werden.

Kurzinfo

Digitalen Ausgang setzen oder rücksetzen

Syntax

OUT n s

Parameter

n = Nummer des Ausgangs
SyncPos-Option: O1 ... O8
Relais O1 = 11
Relais O4 = 14
s = Zustand (0 = OFF, 1 = ON)



ACHTUNG!

Nach dem Einschalten der Anlage sind alle Ausgänge auf OFF. Der Schaltzustand von Ausgängen, die gemäß den I/O-Parametern vordefinierte Funktionen besitzen, wird mit dem OUT Befehl ebenfalls beeinflusst! Der aktuelle Schaltzustand bleibt auch nach Beendigung oder Abbruch eines Programmes erhalten. Die Schaltlogik sowie die maximale Strombelastbarkeit sind im Kapitel Ein- und Ausgangsklemmen und im VLT5000 Handbuch beschrieben.

Befehlsgruppe

I/O

Querverweise

OUTB, IN, INB

Syntax-Beispiel

```
OUT 3 1      /* 03 an */
OUT 6 0      /* 06 aus */
```

Programmbeispiele

OUT_01.M

■ OUTAN

Mit dem OUTAN Befehl können Sie den Sollwert für den VLT Bus vorgeben. (Der Drehzahl- oder Drehmoment-Sollwert hängt davon ab, wie der VLT-Parameter 100 gesetzt ist.)



ACHTUNG!

Vorher muss der Befehl MOTOR OFF ausgeführt werden. Daher ist die Schleppfehlerüberwachung nicht mehr aktiv.

Mit OUTAN kann man auch in OPEN LOOP mit MOTOR OFF den Regler abschalten und den VLT ohne Rückführung als reinen Frequenzumrichter betreiben. So können Sie SyncPos benutzen um direkt Sollwerte auszugeben, Eingänge zu lesen usw.

Kurzinfo

(Drehzahl-) Sollwert für VLT setzen

Syntax

OUTAN w

Parameter

w = Bus Sollwert

Bereich: -0X4000 ... 0X4000 = -100 % ... 100 %



ACHTUNG!

Bitte beachten Sie, dass bei älteren Versionen eine andere Skalierung vorliegen kann.

Befehlsgruppe

I/O

Querverweise

MOTOR OFF

Kapitel Ein- und Ausgangsklemmen; VLT5000 Betriebsanleitung

Syntax-Beispiel

MOTOR OFF

OUTAN 0X2000

/* Drehzahlsollwert auf 50 % setzen */

■ OUTB

Mit dem OUTB Befehl kann der Zustand der digitalen Ausgänge byteweise verändert werden. Der übergebene Bytewert bestimmt den Zustand der einzelnen Ausgänge. Das niederwertigste Bit des Bytewertes entspricht dabei dem Sollzustand des Ausgangs 1.

Kurzinfo

Zustand der digitalen Ausgänge byteweise verändern

Syntax

OUTB n v

Parameter

n = Ausgangsbyte (0 = O1 ... O8)

v = Wert (0 ... 255)



ACHTUNG!

Nach dem Einschalten der Anlage sind alle Ausgänge auf OFF. Der Schaltzustand von Ausgängen, die gemäß den I/O-Parametern vordefinierte Funktionen besitzen, wird mit dem OUTB Befehl ebenfalls beeinflusst! Der aktuelle Schaltzustand bleibt auch nach Beendigung oder Abbruch eines Programmes erhalten.

Schaltlogik, sowie maximale Strombelastbarkeit: siehe Kapitel Eingangs- und Ausgangsklemmen.

Befehlsgruppe

I/O

Querverweise

OUT, IN, INB

Syntax-Beispiel

OUTB 0 10

/* Ausgang 2 und 4 durchschalten, sonstige Ausgänge sperren */

OUTB 0 245

/* Ausgang 2 und 4 sperren, alle anderen Ausgänge durchschalten */

OUTB 0 128

/* nur Ausgang 8 durchschalten, andere sperren */

Programmbeispiele

OUTB_01.M

■ OUTDA

Mit dem OUTDA Befehl kann man die analogen und digitalen Ausgänge der VLT Steuerkarte steuern.

Der **VLT5000** hat zwei konfigurierbare Ausgänge – das können analoge oder digitale sein –, sie werden mit den Parametern 319 (Ausgang 42) und 321 (Ausgang 45) eingestellt.

Der **VLT5000Flux** hat zwei analoge Ausgänge (Ausgang 42 wird in Parameter 319, Ausgang 45 in Parameter 321 eingestellt) und zwei digitale/puls Ausgänge (Ausgang 26 wird in Parameter 341, Ausgang 46 in Parameter 355 eingestellt).

Ein VLT-Steuerkarten-Ausgang kann nur vom Anwendungsprogramm eingestellt werden, wenn er als Options-Ausgang im geeigneten Parameter eingerichtet ist.

Kurzinfo

VLT Ausgang setzen

Syntax

OUTDA nr wert

Parameter VLT5000

nr = Ausgangsnummer (42 oder 45)

wert = wenn Parameter 319/321 =

„Option digital“: Wert = 0 oder 1

0 = Ausgang low (0 V)

1 = Ausgang high (24 V)

wenn Parameter 319/321 =

„Option 0...20 mA“: Wert = 0...100000 entspricht dann 0...20 mA

„Option 4...20 mA“: Wert = 0...100000 entspricht dann 4...20 mA

„Option 0...32 kHz“: Wert = 0...100000 entspricht dann 0...32 kHz

Parameter VLT5000Flux

nr = Ausgangsnummer (26, 42, 45 und 46)

wert = wenn Parameter 319/321 (output 42/45) =

„Option 0...20 mA“: Wert = 0...100000 entspricht dann 0...20 mA

oder

„Option 4...20 mA“: Wert = 0...100000 entspricht dann 4...20 mA

oder

wenn Parameter 341/355 (output 26/46) =

„Option digital“: Wert = 0 oder 1

0 = Ausgang low (0 V)

1 = Ausgang high (24 V)

oder

„Option 0...50 kHz“: Wert = 0...100000 entspricht dann 0...50 kHz

Befehlsgruppe

I/O

Querverweise

VLT-Parameter 319, 321, 341 (nur Flux) und 355 (nur Flux)

Syntax-Beispiel

OUTDA 42 5000

/* Analogausgang auf 10 mA setzen */

/* Voraussetzung: Parameter 319 ist auf den */

/* Bereich „Option 0...20 mA“ eingestellt */

■ PCD

Ohne einen zusätzlichen Befehl COMOPTGET oder COMOPTSEND können Sie mit dem Befehl PCD direkt auf den Feldbus-Datenbereich zugreifen. Es wird wortweise (16-Bit) der Kommunikationsspeicher beschrieben oder gelesen.

Kurzinfo

Pseudo-Array für den direkten Zugriff auf den Feldbus-Datenbereich

Syntax

PCD[n]

n = index

Befehlsgruppe

Kommunikationsoption

Querverweise

SYSVAR

Syntax-Beispiel

Variable = PCD[1] // Wort 1

Variable = PCD[1].2 // Bit 2 von Wort 1

Variable = PCD[2].b1 // Byte 1 von Wort 2

PCD[1] = Variable

PCD[1].3 = Variable

Programmbeispiel

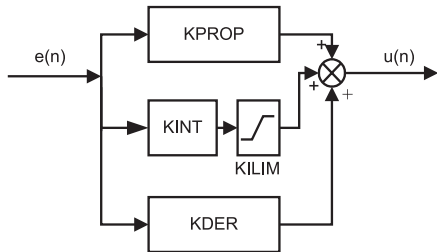
_IF (PCD [2] ==256) THEN // Wert vergleichen

_IF (PCD [3].2) THEN // ist Bit 2 von PCD3 high?

■ PID

Mit dieser Funktion kann ein PID-Filter berechnet werden, der nach folgender Formel arbeitet:

$$u(n) = (k_p * e(n) + k_d * (e(n) - e(n-1)) + k_i * \sum e(n)) / \text{Timer}$$



wobei gilt:

- e(n) Fehler zum Zeitpunkt n
- k_p Proportionalfaktor des PID-Reglers
- k_d Differentialfaktor
- k_i Integralfaktor (begrenzt durch Integrationslimit)
- Timer Abtastzeit

Die entsprechenden Faktoren können gesetzt werden durch die Befehle:

```
SET PID KPROP 1 /* setze kp 1 */
SET PID KDER 1 /* setze kd 1 */
SET PID KINT 0 /* setze ki 0 */
SET PID KILIM 0 /* Integrationslimit 0 */
SET PID TIMER 1 /* Abtastzeit = 1 */
```

wobei das folgende Syntax-Beispiel auch gleich die Default-Belegung der Faktoren zeigt.

Kurzinfo

PID-Filter berechnen

Syntax

`u(n) = PID e(n)`

Parameter

e(n) = aktuelle Abweichung (Fehler) auf die der PID-Filter angewendet werden soll

Rückgabewert

u(n) = Ergebnis der PID-Berechnung

Befehlsgruppe

I/O

Syntax-Beispiel

```
e = INAD 53
u = PID e
PRINT "input = ",e, "output = ",u
```

■ POSA

Mit dem POSA Befehl wird die Achse auf eine Position absolut zum aktuellen Nullpunkt bewegt.

Wenn durch den Befehl POSA die Positionsgrenzen NEGLIMIT (4) oder POSLIMIT (5) überschritten werden, wird nach dem Fehler mit dem nächsten Befehl fortgefahren.

Kurzinfo

Absolut zum aktuellen Nullpunkt positionieren

Syntax

`POSA p`

Parameter

p = Position in Benutzereinheiten (BE) absolut zum aktuellen Nullpunkt; in der Standardeinstellung entspricht die BE der Anzahl der Quadcounts.


ACHTUNG!

Wenn ein mit SET ORIGIN gesetzter und aktiver Temporärnullpunkt existiert, wird die Positionsangabe auf diesen Nullpunkt bezogen.


ACHTUNG!

Sollte beim Aufruf des POSA Befehls noch keine Beschleunigung und/oder Geschwindigkeit definiert sein, wird mit den Werten der Parameter DFLTVEL (33) und DFLTACC (34) gefahren.

Befehlsgruppe

ABS

Querverweise

VEL, ACC, POSR, HOME, DEF ORIGIN, SET ORIGIN
Parameter: POSFACT_Z (23), POSFACT_N (26)

Syntax-Beispiel

```
POSA 50000
/* Achse auf Position 50000 fahren */
```

Programmbeispiele

POS_01.M

■ POSA CURVEPOS

Dieser Befehl wirkt wie ein POSA Befehl und bewegt den Slave zur korrespondierenden Position der Kurve, die durch die aktuelle Master-Position vorgegeben ist.

Kurzinfo

Slave auf die, der Master-Position entsprechenden Kurvenposition fahren

Syntax

POSA CURVEPOS


ACHTUNG!

Wenn ein mit SET ORIGIN gesetzter und aktiver Temporärnullpunkt existiert, wird die Positionsangabe auf diesen Nullpunkt bezogen. Sollte beim Aufruf des POSA Befehls noch keine Beschleunigung und/oder Geschwindigkeit definiert sein, wird mit den Defaultwerten der Parameter DFLTVEL (33) und DFLTACC (34) gefahren.

Befehlsgruppe

ABS, CAM

Querverweise

CURVEPOS, SET ORIGIN

Syntax-Beispiel

```
POSA CURVEPOS
  // Slave auf die, der Master-Position
  // entsprechenden Kurvenposition fahren
```

■ POSR

Mit dem POSR Befehl kann die Achse auf eine Position relativ zur aktuellen Position bewegt werden.

Kurzinfo

Relativ zur aktuellen Position positionieren

Syntax

POSR d

Parameter

d = Distanz zur aktuellen Position in Benutzereinheiten (BE); dies entspricht in der Standardeinstellung der Anzahl Quadcounts.


ACHTUNG!

Sollte beim Aufruf des POSR Befehls noch keine Beschleunigung und/oder Geschwindigkeit definiert sein, wird mit den Werten der Parameter DFLTVEL (33) und DFLTACC (34) gefahren.

Befehlsgruppe

REL

Querverweise

VEL, ACC, POSA

Parameter: POSFACT_Z (23), POSFACT_N (26)

Syntax-Beispiel

```
POSR 50000
/* Achse relativ um 50000 BE verfahren */
```

Programmbeispiele

POS_01.M

■ PRINT

Mit dem PRINT Befehl können Rechenergebnisse, Variableninhalte und Textinformationen über RS485 Kommunikationsschnittstelle am angeschlossenen PC ausgegeben werden, wenn die SyncPos-Software geöffnet und die Kommunikation hergestellt ist.

Um mehrere Informationen mit einem einzigen PRINT Befehl auszugeben, müssen die einzelnen Elemente (Variablen, Texte etc.) durch ein Komma getrennt werden. Textinformationen müssen in Anführungszeichen gesetzt werden.

Nach jeder PRINT Anweisung wird normalerweise ein Zeilenvorschub erzeugt. Dieser automatische Zeilenvorschub lässt sich durch einen Strichpunkt nach dem letzten Ausgabeelement unterdrücken.

Kurzinfo

Ausgabe von Informationen

Syntax

PRINT i oder PRINT i;

Parameter

i = Information, zum Beispiel Variable, Text, CHR (n) getrennt durch Kommata. Der Befehl CHR liefert zu einer Zahl das entsprechende ASCII-Zeichen.


ACHTUNG!

Wenn kein Rechner angeschlossen ist, gehen die PRINT Befehle gleichsam „ins Leere“. Wenn aber ein Rechner ein XOFF schickt, da er gerade keine Zeit hat oder weil das Terminalprogramm beendet wurde, werden die PRINT Befehle in den Puffer geschrieben. Wenn dieser voll ist, hält die Steuerung das SyncPos-Programm an und wartet bis wieder Platz ist.

Befehlsgruppe

I/O

Querverweise

INKEY

Syntax-Beispiel

```
PRINT "Information ist wichtig!"
/* Textinformation ausgeben */
PRINT "Information ist wichtig!";
/* Information ohne Zeilenumbruch ausgeben */
variable = 10
PRINT variable /* Variableninhalt ausgeben */
PRINT APOS
/* Funktionsrückgabewerte ausgeben */
PRINT "Variable", variable,"Pos:.",APOS
/* Gemischte Informationen ausgeben */
```

Programmbeispiele

Verwendung [siehe](#) in allen Programmbeispielen

■ PULSACC

Mit PULSACC setzen Sie die Beschleunigung/Verzögerung für den virtuellen Master (Encoder-Ausgang).

Das virtuelle Mastersignal bildet ein Encodersignal nach. Zur Berechnung der Pulsbeschleunigung PULSACC sind daher die Parameter **Drehgeberauflösung, Mastergeschwindigkeit** und die Rampenzeiten zu berücksichtigen.

Die erzeugten Signale werden auch gleichzeitig als Mastereingang ausgewertet, so dass MAPOS, MIPOS etc. wie in einem externen Master funktionieren.

Kurzinfo

Beschleunigung für den virtuellen Master setzen

Syntax

PULSACC a


ACHTUNG!

Der Änderung der Beschleunigung in PULSACC wird erst nach dem nächsten PULSVEL-Befehl wirklich gesetzt.

Parameter

a = Beschleunigung in Hz/s

PULSACC = 0 ist Bedingung für das Abschalten des Modus Virtueller Master, vorausgesetzt es folgt ein PULSVEL Befehl.

Befehlsgruppe

SYN

Querverweise

PULSVEL

Beispiel

Das virtuelle Mastersignal soll einem Encodersignal von 1024 Strichen/Umdr. entsprechen. Die max. Geschwindigkeit von 25 Encoderumdrehungen/s soll in 1 s erreicht werden.

$$\begin{aligned}
 \text{PULSACC} &= \frac{\Delta \text{ Pulsgeschwindigkeit (PULSVEL) [Hz]}}{\Delta t[\text{s}]} \\
 &= \frac{25 \frac{\text{Umdr.}}{\text{s}} \cdot 1024 \text{ Striche/Umdr.}}{1 \text{ s}} \\
 &= 25600 \frac{\text{Striche}}{\text{s}^2} = 25600 \frac{\text{Hz}}{\text{s}}
 \end{aligned}$$

■ PULSVEL

Mit PULSVEL setzen Sie die Geschwindigkeit für den virtuellen Master (Encoder Ausgang).

Das virtuelle Mastersignal bildet ein Encodersignal nach. Zur Berechnung der Pulsgeschwindigkeit sind daher die Parameter **Drehgeberauflösung** und **Mastergeschwindigkeit** zu berücksichtigen.

Kurzinfo

Geschwindigkeit für den virtuellen Master setzen

Syntax

PULSVEL v

Parameter

v = Geschwindigkeit in Pulsen pro Sekunde (Hz)

Befehlsgruppe

SYN

Querverweise

PULSACC

Beispiel

Das virtuelle Mastersignal soll einem Encodersignal von 2048 Strichen/Umdr. bei einer Encoderdrehzahl von 50 Umdr./s entsprechen.

$$\begin{aligned} \text{PULSVEL} &= \text{Encoderstriche/Umdr.} \cdot \frac{\text{Umdrehungen}}{\text{s}} \\ &= 2048 \cdot 50 \text{ Hz} = 102400 \text{ Hz} \end{aligned}$$

■ REPEAT .. UNTIL ..

Mit der REPEAT .. UNTIL Konstruktion ist es möglich, den eingeschlossenen Programmbereich in Abhängigkeit von einem beliebigen Abbruchkriterium ein- oder mehrfach zu wiederholen. Das Abbruchkriterium setzt sich aus einer oder mehreren Vergleichsoperationen zusammen und wird stets am Schleifenende überprüft. Solange das Abbruchkriterium nicht erfüllt ist, wird der Schleifeninhalt wiederholt abgearbeitet.

Kurzinfo

Bedingte Schleife mit Endkriterium (Wiederhole ... bis Bedingung erfüllt)

Syntax

REPEAT

UNTIL Bedingung

Parameter

Bedingung = Abbruchkriterium



ACHTUNG!

Da das Abbruchkriterium erst am Schleifenende überprüft wird, werden die Befehle innerhalb der Schleifenkonstruktion mindestens einmal ausgeführt. Um eine Endlosschleife zu vermeiden müssen die innerhalb der Schleife abgearbeiteten Befehle direkt oder indirekt Einfluss auf das Ergebnis der Abbruchüberprüfung haben.

Befehlsgruppe

CON

Querverweise

LOOP, WHILE .. DO .. ENDWHILE

Syntax-Beispiel

REPEAT /* Schleifenstart */

Befehlszeile 1

Befehlszeile n

UNTIL (A !=1) /* Abbruchbedingung */

Programmbeispiele

REPEA_01.M

DIM_01.M, ONINT_01.M, OUT_01.M, INKEY_01.M

■ RST ORIGIN
Kurzinfo

Temporärnullpunkt löschen

Syntax

RST ORIGIN

Beschreibung

Mit dem RST ORIGIN Befehl wird ein zuvor mit SET ORIGIN gesetzter Temporärnullpunkt wieder gelöscht, und alle Folgenden absoluten Positionierbefehle (POSA) beziehen sich wieder auf den Realnullpunkt.

Befehlsgruppe

INI

Querverweise

SET ORIGIN, DEF ORIGIN, POSA

Syntax-Beispiel

RST ORIGIN /* Temporärnullpunkt rücksetzen */

Programmbeispiele

TORIG_01.M, OUT_01.M, VEL_01.M

■ SAVE part
Kurzinfo

Arrays, aktuelle Achsenparameter oder Globale Parameter im EEPROM sichern

Syntax

SAVE part

part = ARRAYS
 AXPARS
 GLBPARS

Beschreibung

Mit diesem Befehl können Sie im Gegensatz zu SAVEPROM die Array-Elemente, Achsenparameter oder Globalen Parameter getrennt voneinander im EEPROM speichern.


NB!

Das EEPROM kann nur bis zu 10000 mal diesen Befehl ausführen.

Befehlsgruppe

INI

Querverweise

DELETE ARRAYS, SAVEPROM

■ SAVEPROM
Kurzinfo

Speicher in EEPROM sichern

Beschreibung

Wenn Array-Elemente oder Benutzerparameter (130–229) geändert werden, während das Programm läuft, kann man mit SAVEPROM die veränderten Werte speichern. Dazu müssen Sie SAVEPROM explizit auslösen.

SAVEPROM löst den selben Vorgang aus, wie er auch aus dem Menü "STEUERUNG" heraus gestartet werden kann.

Wenn Sie nur Array-Elemente oder nur Globale und Benutzerparameter sichern wollen, benutzen Sie die Befehle SAVE ARRAY und SAVE GLBPARS.


ACHTUNG!

Die Ausführungszeit von SAVEPROM hängt von der Menge der zu sichernden Daten ab. Es können bis zu 4 Sekunden sein.


ACHTUNG!

Bitte beachten Sie, dass Achsenparameter nicht mit SAVEPROM gespeichert werden. Dazu müssen Sie den Befehl SAVE AXPARS benutzen.


ACHTUNG!

Das EEPROM kann nur bis zu 10000 mal diesen Befehl ausführen.

Befehlsgruppe

INI

Syntax-Beispiel

PRINT "Einen Moment Geduld"
 SAVEPROM
 PRINT "Danke"

■ SET

Mit dem Befehl SET können bestimmte Achs- und globale Parameter temporär während der Programmlaufzeit verändert werden.

Die zulässigen Parameterkennungen finden Sie in der Übersicht der [Parameter-Referenz](#) in der linken Spalte, zum Beispiel KDER (12) für den **Differentialfaktor** oder POSERR (15) für den **Tolerierten Positionfehler**. Beispiele für globale Parameter sind SET PRGPAR (102) für die **aktivierte Programmnummer** oder SET I_BREAK (105) für den **Eingang für Abbruch**.

Kurzinfo

Setzt einen Achsparameter, einen globalen Parameter oder einen Benutzerparameter

Syntax

SET par v

Parameter

par = Parameterkennung

v = Parameterwert


ACHTUNG!

Die Parameteränderungen gelten nur während das Programm läuft. Nach dem Programmende oder -abbruch sind wieder die ursprünglichen Parameter gültig.

Die Parameteränderungen können mit dem Befehl SAVEPROM permanent gespeichert werden.

Befehlsgruppe

PAR

Querverweis

GET

Parameter-Referenz

Syntax-Beispiel

SET POSLIMIT 100000

/* Positive Fahrbereichsgrenze setzen */

SET KPROP 150

/* Proportionalfaktor verändern */

■ SETCURVE

Mit SETCURVE wird die CAM-Kurve, die in dem *array* beschrieben ist, ausgewählt. SETCURVE *array* muss immer vor den Befehlen CURVEPOS, SYNCCxx, SYNCCSTART oder SYNCCSTOP benutzt werden. Sobald der Befehl ausgeführt wird, sind die notwendigen Vorberechnungen bereits durchgeführt.

Kurzinfo

CAM-Kurve setzen

Syntax

SETCURVE array

Parameter

array = Name des Arrays bzw. der Kurve


ACHTUNG!

Vor dem Befehl SETCURVE bzw. am Anfang des Programmes muss die DIM-Anweisung mit dem Namen der Kurve bzw. des Arrays und der Anzahl der Array-Elemente stehen. Sind mehrere Arrays bzw. Kurven in der cnf-Datei, dann muss die Reihenfolge in der DIM-Anweisung mit der Reihenfolge der Arrays in der cnf-Datei übereinstimmen.


ACHTUNG!

Wenn SYNCC nicht aktiv ist:

Wird SETCURVE benutzt, wenn SYNCC nicht aktiv ist, dann wird durch den Befehl SETCURVE die Kurven-Master-Position zurückgesetzt, und zwar abhängig von der aktuellen Master-Position. Das bedeutet, dass CMASTERCPOS (SYSVAR 4230) aus MAPOS berechnet wird. Diese Position wird also nicht mehr durch SYNCC zurück gesetzt, sondern kann nur durch ein DEFMCPOS oder durch eine neue SETCURVE außerhalb des SYNCC-Modus zurück gesetzt werden.

Wenn SYNCC aktiv ist:

Wird SETCURVE aber benutzt, während SYNCC aktiv ist, wird CMASTERCPOS nicht verändert. Alle anderen Parameter wie POSFACT_N, POSFACT_Z, SYNCMSTART, SYNCMARKM, SYNCMARKS, SYNCOMPULSM, SYNCOMPULSS, SYNCMWINM, SYNCMWINS und alle Kurven-Array-Informationen werden nach dem nächsten Re-Start der Kurve aktualisiert .

Während SYNCC aktiv ist, kann die Position CMASTERCPOS nur durch einen Befehl DEFMCPOS, der mit dem nächsten Re-Start der Kurve ausgeführt wird, oder MOVESYNCORIGN, der sofort ausgeführt wird., beeinflusst werden.

CMASTERCPOS (SYSVAR) und CURVEPOS werden nun auch aktualisiert, sogar wenn SYNCC nicht mehr aktiv ist. Diese Werte werden aktualisiert nach einem Befehl SETCURVE (falls SYNCMSTART < 2000 ist) oder nach SYNCC und dem erster Master-Marker (falls SYNCMSTART = 2000). Nachdem der SYNCC Befehl angehalten wurde, wird fortgefahren, diese Werte zu aktualisieren, wenn SYNCMSTART < 2000.

Befehlsgruppe

PAR

Portabilität

Aktualisierung der CMASTERCPOS in Abhängigkeit von SYNCC ab Optionskarte, Version 4.50 aufwärts.

Querverweise

DIM, CMASTERCPOS, CURVEPOS

Syntax-Beispiel

DIM curve [280]

// siehe Anzahl der Elemente in der Titelleiste

// des CAM-Editors

SETCURVE curve

■ SETMORIGIN

Mit dem SETMORIGIN Befehl können Sie eine beliebige Position als neuen Nullpunkt für den Master setzen.

Kurzinfo

Beliebige Position als Nullpunkt für den Master setzen

Syntax

SETMORIGIN wert

Parameter

wert = absolute Position


ACHTUNG!

Der Befehl SETMORIGIN hebt den Befehl DEFMORIGIN auf.


ACHTUNG!

Um den Nullpunkt für den Master wieder zu ändern, müssen Sie ihn daher mit SETMORIGIN oder DEFMORIGIN neu setzen. RST ORIGIN hat für den Nullpunkt des Masters keine Auswirkung.

Befehlsgruppe

INI

Querverweise

DEFMORIGIN, MAPOS

Syntax-Beispiel

SETMORIGIN 10000

/* Nullpunkt für den Master auf 10000 setzen */

■ SET ORIGIN

Mit dem SET ORIGIN Befehl kann eine beliebige absolute Position vorübergehend als neuer Bezugspunkt für den absoluten Positionierbefehl (POSA) gesetzt werden. Diese Position wird Temporärnullpunkt genannt.

In Verbindung mit dem Befehl CURVEPOS kann man festlegen, dass die aktuelle Slave-Position mit dem entsprechenden Wert der Kurve übereinstimmt.

Kurzinfo

Absolutposition als Temporärnullpunkt setzen

Syntax

SET ORIGIN p

Parameter

p = Absolute Position in Bezug zum Realnullpunkt



ACHTUNG!

Es ist möglich, mehrere SET ORIGIN ohne ein vorheriges RST ORIGIN auszuführen. Die absolute Positionsangabe bezieht sich immer auf den Realnullpunkt. Der letzte ausgeführte SET ORIGIN bestimmt somit die Lage des Temporärnullpunktes in Bezug zum Realnullpunkt.

Befehlsgruppe

INI

Querverweise

RST ORIGIN, DEF ORIGIN, POSA, CURVEPOS, POSA CURVEPOS

Syntax-Beispiel 1

```
SET ORIGIN 50000
// Temporären Nullpunkt auf 50000 setzen
```

Syntax-Beispiel 2

```
SET ORIGIN (-CURVEPOS)
// Temporärnullpunkt auf den Beginn der
// Kurve setzen
```

Programmbeispiele

TORIG_01.M
OUT_01.M, VEL_01.M

■ SETVLT

Mit dem SETVLT Befehl können bestimmte Parameter im VLT temporär verändert und somit auch die Konfiguration des VLT temporär werden.

Da ausschließlich Ganzzahlenwerte übertragen werden, muss der zu übertragende Parameterwert mit dem zugehörigen Konvertierungsindex angepasst werden.

Die Liste der VLT-Parameter mit den zugehörigen Konvertierungsindizes finden Sie in der VLT-Betriebsanleitung.

Kurzinfo

Setzt einen VLT-Parameter

Syntax

SETVLT par v

Parameter

par = Parameternummer

v = Parameterwert



ACHTUNG!

Die Parameteränderungen werden nur im RAM gespeichert. Nach dem Aus- und Wiedereinschalten werden die ursprünglichen Parameter wieder eingelesen.

Befehlsgruppe

PAR

Querverweis

GETVLT

Syntaxbeispiel

```
SETVLT 202 60000
/* Parameter 202 "Maximaler Sollwert" auf 60 Hz
ändern */
/* -Konvertierungsindex = -3 (Multiplikation mit 103
beim Senden) +/-
```

■ SETVLTSUB

Mit SETVLT Befehlen können VLT-Parameter und somit kann auch die Konfiguration des VLT vorübergehend geändert werden, in diesem Fall auch Parameter mit Indexnummern.

Da ausschließlich Ganzzahlenwerte übertragen werden, muss der zu übertragende Parameterwert mit dem zugehörigen Konvertierungsindex angepasst werden.

Die Liste der VLT-Parameter mit den zugehörigen Konvertierungsindizes finden Sie in der VLT-Betriebsanleitung.


ACHTUNG!

Die Parameteränderungen werden nur im RAM gespeichert. Nach dem Aus- und Wiedereinschalten werden die ursprünglichen Parameter wieder eingelesen.

Kurzinfo

Setzt oder ändert einen VLT-Parameter mit Indexnummer

Syntax

SETVLTSUB par indxnr v

Parameter

par = Parameternummer
 indxnr = Indexnummer
 v = Parameterwert

Befehlsgruppe

PAR

Querverweis

GETVLTSUB

Syntax Beispiel

```
SETVLT 25 1 100
// Index 1 des Parameters 25 "Quick Menü"
// (nur VLT5000Flux) auf 100 "Konfiguration"
// setzen
```

■ STAT

Der STAT Befehl liefert den aktuellen Status der Lagereglereinheit sowie der Achse zurück. Zum Beispiel ob die Lageregelung abgeschaltet, die Bewegung beendet oder der Endschalter aktiv ist. Der Zustand der Programmausführung kann mit STAT nicht abgefragt werden, sondern nur mit AXEND.

Der Status wird aus vier Byte zusammengesetzt.

Kurzinfo

Status einer Achse und der Steuerung abfragen

Syntax

erg = STAT

Rückgabewert

erg = Achs- und Steuerungsstatus (4-Bytewert):

Byte 3	MSB	
Bit 0	1 = Achse fährt	
Bit 1	1 = Überlauf Slave Drehgeber	
Bit 2	1 = Überlauf Master Drehgeber	
Bit 3	1 = Lageregelung ist temporär abgeschaltet *)	
Byte 2	Statusbyte Lagereglereinheit	
Bit 7	1 = Lageregelung abgeschaltet	
Bit 2	1 = Position erreicht	
Bit 0,1,3-6	ohne Bedeutung	
Byte 1	wird nicht verwendet	
Byte 0	LSB	
Bit 7	1 = Endschalter aktiv	
Bit 6	1 = Referenzschalter aktiv	
Bit 2	1 = Lageregelung abgeschaltet	
Bit 0,1,3,4	nicht verwendet	

*) Erläuterung: d.h. die Achse befindet sich innerhalb des Toleranzbereiches des Regelfensters REGWINMAX / REGWINMIN. Sobald das Regelfenster verlassen wird, wird die Lageregelung wieder eingeschaltet.

Befehlsgruppe

I/O

Querverweise

AXEND

Syntax-Beispiel

PRINT STAT /* Statuswort ausgeben */

Programmbeispiel

STAT_01.M

■ SUBMAINPROG .. ENDPROG

Das Kennwort SUBMAINPROG leitet den Unterprogramm-bereich ein und das Kennwort ENDPROG beendet diesen speziellen Programm-bereich. Unter einem Unterprogramm versteht man Befehlsfolgen, die mit der GOSUB Anweisung von verschiedenen Programmpositionen aus aufgerufen und ausgeführt werden können.

Innerhalb des Unterprogramm-bereichs müssen alle verwendeten Unterprogramme enthalten sein. Es ist möglich den Unterprogramm-bereich an beliebiger Stelle innerhalb eines Programmes einzufügen, aus Gründen der Übersichtlichkeit wird jedoch empfohlen, den Unterprogramm-bereich an den Programm-anfang oder an das Programmende zu stellen.

Kurzinfo

Definition des Unterprogramm-bereichs

Syntax

```
SUBMAINPROG
ENDPROG
```



ACHTUNG!

Es darf nur einen Unterprogramm-bereich innerhalb eines Programmes geben.

Befehlsgruppe

CON

Querverweise

SUBPROG ...RETURN,
GOSUB, ON ERROR GOSUB, ON INT n GOSUB

Syntax-Beispiel

```
SUBMAINPROG
/* Beginn des Unterprogramm-bereichs */
Unterprogramm 1
Unterprogramm n
ENDPROG
/* Ende des Unterprogramm-bereichs */
```

Programmbeispiele

```
GOSUB_01.M
AXEND_01.M, ERROR_01.M, INCL_01.M, STAT_01.M
```

■ SUBPROG name .. RETURN

Die Anweisung SUBPROG kennzeichnet den Beginn eines Unterprogrammes. Direkt nach dem SUBPROG-Kennwort muss der Name des Unterprogrammes stehen. Der Name kann aus einem oder mehreren Zeichen bestehen und muss eindeutig sein, das heißt es darf nicht mehrere Unterprogramme mit dem gleichen Namen geben.

Mit der GOSUB Anweisung und dem Namen kann ein Unterprogramm zu jedem beliebigen Zeitpunkt aufgerufen und ausgeführt werden.

Ein Unterprogramm kann beliebig viele Befehlszeilen enthalten und auf alle Programmvariablen zugreifen. Der letzte Befehl innerhalb eines jeden Unterprogrammes muss die RETURN Anweisung sein, durch die das Unterprogramm verlassen und die Programmausführung mit dem auf die GOSUB Anweisung folgenden Befehl fortgesetzt wird.

Kurzinfo

Definition eines Unterprogrammes

Syntax

```
SUBPROG name
RETURN
```

Parameter

name = Name der Unterprogrammes



ACHTUNG!

Alle Unterprogramme müssen sich innerhalb des durch SUBMAINPROG und ENDPROG definierten Programm-bereichs befinden. Es ist nicht zulässig innerhalb eines Unterprogrammes ein zweites Unterprogramm zu deklarieren.

Befehlsgruppe

CON

Querverweise

SUBMAINPROG ... ENDPROG,
GOSUB, ON ERROR GOSUB, ON INT n GOSUB

Syntax-Beispiel

```
SUBMAINPROG          /* Beginn UP-Bereich */
SUBPROG up1          /* Beginn von up1 */
Befehlszeile 1
Befehlszeile n
RETURN                /* Ende von up1 */
ENDPROG              /* Ende UP-Bereich */
```

Programmbeispiele

```
GOSUB_01.M
AXEND_01.M, ERROR_01.M, IF_01.M, STAT_01.M
```

■ SWAPMENC

Dieser Befehl erlaubt es, intern Master- und Slave-Drehgeber zu vertauschen. Dies ist insbesondere dann sinnvoll, wenn man mit einer Steuerung zwei Motoren abwechselnd verwenden will.

Vor dem Befehl SWAPMENC ON/OFF muss zur Vermeidung eines Schleppfehlers immer ein MOTOR OFF ausgeführt werden. Auch müssen die Regelparameter oder Achsparameter geändert werden, falls die beiden Motoren unterschiedlich sind. Die Motorleitungen werden über Relais umgeschaltet.

Kurzinfo

Master- und Slave-Encoder intern tauschen

Syntax

SWAPMENC s

Parameter

s = Bedingung
 ON = Der Eingang des Master-Encoders ist der Istwert-Eingang für die Regelung.
 OFF = Der Eingang des Slave-Encoders ist der Istwert-Eingang für die Regelung


ACHTUNG!

Bei diesem Wechsel gehen keine Positionen verloren, auch wenn die Motoren von Hand bewegt werden, während der andere Motor geregelt ist. Man hat auf den jeweils nicht geregelten Antrieb auch immer Zugriff über MAPOS.

Befehlsgruppe

INI

Querverweise

MAPOS

Syntax-Beispiel

SWAPMENC ON
 // Slave-Encoder intern mit Master-Encoder tauschen

Beispiel

```
MOTOR OFF
OUT 1 1 // Motorleitungen umschalten
SET KPROP ... // Achsparameter ändern
SWAPMENC ON // intern Drehgeber umschalten
MOTOR ON // Regelung wieder einschalten
POSA 10000 // Motor, der am Master-Encoder
// angeschlossen ist, verfahren

MOTOR OFF
OUT 1 0 // Motorleitungen umschalten
SET KPROP ... // Achsparameter ändern
SWAPMENC OFF // intern Drehgeber umschalten
MOTOR ON // Regelung wieder einschalten
POSA 0 // jetzt wieder den Motor, der am
// Slave-Drehgeber angeschlossen ist, verfahren
```

■ SYNCC

Der Befehl SYNCC startet den CAM-Mode (Kurvenscheibensteuerung). Von diesem Augenblick werden die Kurvenpositionen des Masters hoch gezählt, abhängig von den tatsächlichen Master-Positionen und dem definierten Startverhalten in SYNCMSTART (62): Wo und wann angefangen wird, zu zählen. Mit dem Parameter SYNCMSTART = 2000 werden die Kurvenpositionen des Masters erst nach dem nächsten Master-Marker gezählt.

Kurzinfo

Synchronisation im CAM-Mode

Syntax

SYNCC num

Parameter

num = Anzahl der Kurven, die abgearbeitet werden sollen;
 0 = der Antrieb bleibt im CAM-Mode bis mit Befehlen wie MOTOR STOP, CSTART, POSA etc. ein anderer Mode gestartet wird.


ACHTUNG!

Der Befehl SYNCC startet weder den Slave noch unterbricht er eine Fahrbewegung (z.B. CVEL); nur SYNCCSTART startet tatsächlich.


ACHTUNG!

Der Antrieb bleibt so lange im CAM-Mode bis *num* Kurven erfolgreich abgearbeitet wurden. Wenn (nach *num* Kurven) die Synchronisation normal verlassen wird, wird – falls kein SYNCCSTOP mit einem entsprechenden Punktepaar definiert ist, – das Start-Stop-Punktepaar 2 benutzt, um den Antrieb anzuhalten. Dieser wird dann an der Position *slavepos* (siehe Parameter) stehen bleiben.

Befehlsgruppe

CAM

Querverweise

SYNCCSTART

Syntax-Beispiel

```
DIM curve [280] // siehe Anzahl der Elemente in
// Titelleiste des CAM-Editors
SETCURVE curve// Kurve setzen
SYNCC // Synchronisation im CAM-Mode
```

■ **SYNCCMM**

Der Befehl SYNCCMM bewirkt wie SYNCC eine Synchronisation im CAM-Mode, führt aber zusätzlich eine Markerkorrektur (nur, wenn der Master vorwärts fährt) durch.

Um den Abstand zwischen Sensor und Arbeitspunkt zu speichern, wird der Parameter SYNCMPULSM(58) benutzt. Damit kann die Markerposition ohne Änderung der Kurve korrigiert werden. Und es sind auch größere Sensorabstände als die eigentliche Kurvenlänge möglich. In diesem Fall wird für die Markerkorrektur ein FIFO benutzt (siehe Beispiel).

Marker kann der Nullimpuls des Drehgebers oder ein externes 24-Volt-Signal sein. (I5 = Master; I6 = Slave).

Kurzinfo

Synchronisation im CAM-Mode mit Markerkorrektur

Syntax

SYNCCMM num

Parameter

num = Anzahl der Kurven, die abgearbeitet werden sollen;
 0 = der Antrieb bleibt im CAM-Mode bis mit Befehlen wie MOTOR STOP, CSTART, POSA etc. ein anderer Mode gestartet wird.

ACHTUNG!
 Der Befehl SYNCCMM startet weder den Slave noch unterbricht er eine Fahrbewegung (z.B. CVEL); nur SYNCCSTART startet tatsächlich.

ACHTUNG!
 Der Antrieb bleibt so lange im CAM-Mode bis *num* Kurven erfolgreich abgearbeitet wurden. Wenn (nach *num* Kurven) die Synchronisation normal verlassen wird, wird – falls kein SYNCCSTOP mit einem entsprechenden Punktepaar definiert ist, – das Start-Stop-Punktepaar 2 benutzt, um den Antrieb anzuhalten. Dieser wird dann an der Position *slavepos* (siehe Parameter) stehen bleiben.

Befehlsgruppe

SYN, CAM

Querverweise

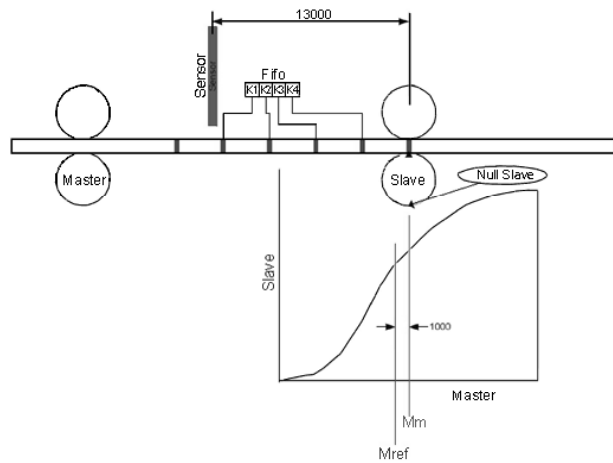
SYNCMPULSM (58)

Syntax-Beispiel

```
SETCURVE curve
SYNCCMM 1
    // 1 x Synchronisieren im CAM-Mode mit
    // Markerkorrektur
```

Beispiel

Wenn zum Beispiel die Kurvenlänge 3000 und der Abstand des Sensors zum Arbeitspunkt 13000 ist, gibt es ein FIFO mit 4 Registern und einen Offset von 1000, der betrachtet werden muss.



■ SYNCCMS

Der Befehl SYNCCMS bewirkt wie SYNCC eine Synchronisation im CAM-Mode, führt aber zusätzlich eine Markerkorrektur des Slaves durch. Hier wird nicht die Kurvenposition korrigiert, sondern die Slave-Position.

Es wird – im Gegensatz zu SYNCCMM – kein FIFO gebildet.

Der Marker kann der Nullimpuls des Drehgebers oder ein externes 24-Volt-Signal sein.

(I5 = Master; I6 = Slave).

Kurzinfo

Synchronisation im CAM-Mode mit Markerkorrektur des Slaves

Syntax

SYNCCMS num

Parameter

num = Anzahl der Kurven, die abgearbeitet werden sollen;

0 = der Antrieb bleibt im CAM-Mode bis mit Befehlen wie MOTOR STOP, CSTART, POSA etc. ein anderer Mode gestartet wird.


ACHTUNG!

Der Befehl SYNCCMS startet weder den Slave noch unterbricht er eine Fahrbewegung (z.B. CVEL); nur SYNCCSTART startet tatsächlich.


ACHTUNG!

Der Antrieb bleibt so lange im CAM-Mode bis *num* Kurven erfolgreich abgearbeitet wurden. Wenn (nach *num* Kurven) die Synchronisation normal verlassen wird, wird – falls kein SYNCCSTOP mit einem entsprechenden Punktepaar definiert ist, – das Start-Stop-Punktepaar 2 benutzt, um den Antrieb anzuhalten. Dieser wird dann an der Position *slavepos* (siehe Parameter) stehen bleiben.

Befehlsgruppe

CAM

Syntax-Beispiel

SETCURVE curve

SYNCCMS 0

// Synchronisieren im CAM-Mode mit

// Markerkorrektur des Slaves

■ SYNCCSTART

Der Befehl startet die Bewegung des Slaves. Mit *pnum* wählt man das Punktepaar aus, das festlegt, an welcher Master-Position die Synchronisation startet und wo sie beendet sein soll.

Beim Vorwärtsfahren startet die Synchronisation am A-Punkt und wird bis zum B-Punkt beendet. Beim Rückwärtsfahren wird sie am B-Punkt gestartet und bis zum A-Punkt beendet.

Kurzinfo

Slave zur Synchronisation im CAM-Mode starten

Syntax

SYNCCSTART pnum

Parameter

pnum = Start-(Stop-)Punktepaar-Nummer

pnum > 0 Bei Erreichen des entsprechenden A-Punktes wird mit dem Einkuppeln begonnen; vorausgesetzt, der Master fährt in positiver Richtung; die Einkuppelkurve wird im B-Punkt beendet.

Wenn A- und B-Punkt identisch sind, wird der Slave mit der eingestellten Maximalgeschwindigkeit – also ohne Kurve – eingekuppelt, sobald der Master diesen Punkt erreicht hat.

pnum = 0 Der Slave wird sofort mit der eingestellten Maximalgeschwindigkeit eingekuppelt (aufsynchronisiert). Dabei ist es egal, in welcher Richtung der Master fährt und ob er überhaupt fährt.

pnum < 0 Es wird ebenfalls das entsprechende Punktepaar verwendet, allerdings beginnt das Einkuppeln beim B-Punkt und wird beim A-Punkt – also in negativer Richtung beendet.

Befehlsgruppe

CAM

Syntax-Beispiel

SETCURVE curve

SYNCC 0

// Synchronisieren im CAM-Mode

SYNCCSTART 1

// Slave am A-Punkt vom Start-

// Stop-Punktepaar 1 einkuppeln

■ SYNCCSTOP

Der Befehl stoppt die Synchronisation ohne den SYNCC Modus zu verlassen. Der Slave wird entsprechend dem Punktepaar, das in *pnum* definiert ist, ausgekuppelt. Dann erst wird der Slave tatsächlich angehalten. Wenn der Stop-Punkt erreicht ist, muss der Slave auf *slavepos* sein.

Kurzinfo

Slave nach der CAM-Synchronisation anhalten

Syntax

SYNCCSTOP pnum slavepos

Parameter

- pnum = (Start-)Stop-Punktepaar
- pnum > 0 Bei Erreichen des entsprechenden A-Punktes wird mit dem Auskuppeln begonnen; vorausgesetzt, der Master fährt in positiver Richtung; die Auskuppelkurve wird im B-Punkt beendet.
Wenn A- und B-Punkt identisch sind, wird der Slave mit der eingestellten Maximalgeschwindigkeit – also ohne Kurve – ausgekuppelt, sobald der Master diesen Punkt erreicht hat.
 - pnum = 0 Der Slave wird sofort mit der eingestellten Maximalgeschwindigkeit ausgekuppelt. Dabei ist es egal, in welcher Richtung der Master fährt und ob er überhaupt fährt.
 - pnum < 0 Es wird ebenfalls das entsprechende Punktepaar verwendet, allerdings beginnt das Auskuppeln beim B-Punkt und wird beim A-Punkt – also in negativer Richtung beendet.
- slavepos = Position, an der der Slave nach dem Auskuppeln stehen soll.



ACHTUNG!

Wird das Programm ohne einen SYNCCSTOP Befehl verlassen, wird standardgemäß mit dem zweiten Punktepaar ausgekuppelt und an der in den Kurvdaten definierten Slave-Stop-Position angehalten.

Befehlsgruppe

CAM

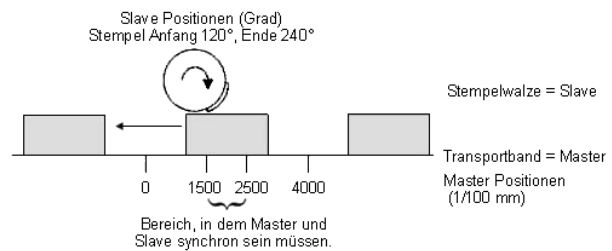
Querverweise

Kurvdaten, Slave-Stop-Position

Syntax-Beispiel

```

SETCURVE curve
SYNCC 0
    // Synchronisieren im CAM-Mode
SYNCCSTART 1
    // Slave mit Start-Punktepaar 1 starten
SYNCCSTOP 2 0
    // Slave mit Stop-Punktepaar 2 an der
    // Slave-Position 0 bzw. 3600 anhalten
    
```



ACHTUNG!

Beim Vorwärtsfahren beginnt das Auskuppeln am A-Punkt und endet am B-Punkt; beim Rückwärtsfahren umgekehrt.

■ SYNCERR

SYNCERR liefert den aktuellen Synchronisationsfehler in Benutzereinheiten [BE] zurück. Das ist der Abstand zwischen der aktuellen Master-Position (umgerechnet mit Getriebefaktor und Offset) und der aktuellen Position des Slaves.

Wenn der Parameter SYNCACCURACY (55) mit negativem Vorzeichen definiert wird, können Sie auch feststellen, ob die Synchronisation vorausläuft (negatives Ergebnis) oder hinterherläuft (positives Ergebnis).

Kurzinfo

Aktuellen Synchronisationsfehler des Slaves abfragen

Syntax

erg = SYNCERR

Rückgabewert

erg = aktueller Synchronisationsfehler des Slaves in BE und

- a) als absoluter Wert, wenn im Parameter SYNCACCURACY die Größe des Genauigkeitsfenster mit positivem Vorzeichen definiert ist;
- b) mit Vorzeichen, wenn in SYNCACCURACY die Größe des Fensters mit negativem Vorzeichen definiert ist.


ACHTUNG!

Bis Optionskarte Version < 5.00:

SYNCERR funktioniert nur im Synchronisationsbetrieb. Sobald man SYNCM oder SYNCNP verlässt, werden die Pulse nicht mehr gezählt. SYNCERR wird nur innerhalb eines SYNC-Befehls aktualisiert.

Mit Optionskarte Version 5.00 aufwärts wird SYNCERR aktualisiert, wenn SYNCNP oder SYNCM nicht mehr aktiv sind, z.B. nach einem MOTOR STOP.

Befehlsgruppe

I/O

Querverweise

TRACKERR, MAPOS, APOS
 SYNCPOSOFFS (54), SYNCMFACTM (49),
 SYNCMFACTS (50), SYNCFACTM (49),
 SYNCFACTS (50), SYNCACCURACY (55)

Syntax-Beispiel

PRINT SYNCERR
 // aktuellen Synchronisationsfehler abfragen

■ SYNCM

Der Befehl SYNCM bewirkt genau wie der folgende SYNCNP Befehl eine Winkel-/Positionssynchronisation mit dem Master, führt aber zusätzlich eine Markerkorrektur durch. Dabei wird bereits während des Anlaufs der Synchronisation auf den nächsten errechneten Marker aufsynchronisiert. So kann man zum Beispiel unterschiedliche Laufverhalten wie Schlupf ausgleichen.

Nachdem die Synchronisation hergestellt ist, wird bei jedem Marker bzw. entsprechend der Einstellung in Parameter SYNCMARKM (52) überprüft, welche Abweichung vorliegt. Diese wird als neuer Offset in die Regelung eingebracht und es wird sofort versucht, die Abweichung auszugleichen. Dabei werden allerdings die eingestellten Werte für Geschwindigkeit VEL sowie Beschleunigung ACC bzw. DEC nicht überschritten.

Kurzinfo

Winkel-/Positionssynchronisation mit dem Master mit Markerkorrektur

Syntax

SYNCM

Besonderheiten

Zu den Parametern, die bei der SYNCNP schon verwendet werden, sind hier auch noch SYNCREADY (56) und SYNCFAULT (57) von Bedeutung.


ACHTUNG!

Da folgende Parameter zu einer Überbestimmung führen können, sollten Sie darauf achten, dass die Werte sinnvoll sind, zueinander passen und mit den Angaben bei den Getriebefaktoren konsistent sind.

SYNCMARKM (52) und SYNCMARKS (53)
 SYNCMPULSM (58) und SYNCMPULSS (59)
 SYNCMTYPM (60) und SYNCMTYPS (61)


ACHTUNG!

SYNCM sollten Sie nur einmal aufrufen, denn die Synchronisierung läuft bis zum nächsten Fahr- oder Stopbefehl. Jeder weitere SYNCM-Befehl führt dazu, dass die Synchronisation von vorne beginnt, was normalerweise nicht beabsichtigt ist, außer Sie setzen den aktuellen SYNCERR zurück.


NEU

Wenn in SYNCMSTART (62) definiert, wird beim Start von SYNCM auf die erste Auswertung der Markerpulse gewartet und erst dann der Offset SYNCPOSOFFS (54) angewandt.

Markersignal

Der Marker kann der Nullimpuls des Drehgebers oder ein externes 24-Volt-Signal sein.
(I5 = Master; I6 = Slave).

Befehlsgruppe

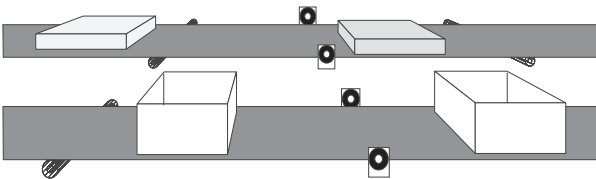
SYN

Querverweise

Parameter der Gruppe AXS

Syntax-Beispiel

```
/* Synchronisieren der Position mit Markerkorrektur */
SYNCM
```

Beispiel


Selbst wenn beide Bänder synchron laufen, würden die Deckel nie auf gleicher Höhe mit den Schachteln sein. Mit SYNCM wird durch die Auswertung der externen Marker die Positionsabweichung zwischen Master und Slave ermittelt und ausgeglichen.

SYNCP

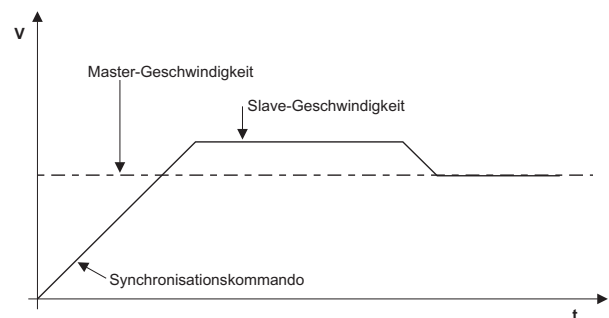
Der Befehl SYNCP führt eine Winkel-/Positionssynchronisation mit dem Master durch. Dabei wird die Position entsprechend der Getriebefaktoren zum Master synchron gehalten, das heißt bei einer Störung von außen wird anschließend versucht, die entsprechende Strecke wieder aufzuholen. Dabei werden allerdings die eingestellten Werte für Geschwindigkeit VEL sowie Beschleunigung ACC bzw. DEC nicht überschritten.

Die folgenden Parameter beeinflussen das Verhalten:

SYNCFACSTM (49), SYNCFACST (50)	Getriebefaktoren
SYNCPOSOFFS (54)	Positionsoffset
SYNCCACCURACY (55)	Genauigkeitsfaktor für Flag
REVERS (63)	Reversierungsverhalten

Bei der Aufsynchronisation wird wie folgt vorgegangen:

Wenn der Befehl SYNCP startet, wird die aktuelle Masterposition ermittelt und als Bezugsposition festgehalten. Aus der Master-Geschwindigkeit wird – unter Berücksichtigung der erlaubten Beschleunigung – die erforderliche Slave-Geschwindigkeit errechnet, um die Masterposition zu erreichen. Der Slave wird solange beschleunigt, bis die errechnete Position erreicht ist oder bis er dicht genug an der Bezugsposition ist, um diese zu erreichen.


Kurzinfo

Winkel-/Positionssynchronisation mit dem Master

Syntax

SYNCP


ACHTUNG!

Sobald die Abweichung zwischen Slave- und Masterposition kleiner ist als SYNC-ACCURACY (55), wird das ACCURACY-Flag gesetzt.

Wenn REVERS (63) so eingestellt ist, dass Zurückfahren nicht erlaubt ist, aber aus irgendeinem Grund der Slave weiter ist als der Master, (z.B. weil nur der Master rückwärts gefahren ist) wartet der Slave mit Geschwindigkeit 0. Dabei berücksichtigt der Slave seine Beschleunigungszeit und fährt ggf. schon los, bevor der Master die Slaveposition erreicht hat. Statt diesem Aufholverfahren kann man auch zuerst den Slave mit CVEL in die Nähe der Master-Geschwindigkeit fahren und dann SYNCNP auslösen. Eine Veränderung des SYNCPOSOFFS (54) während der Aufsynchronisation führt zu erneutem Aufsynchronisieren mit Rampen (siehe oben).


ACHTUNG!

SYNCNP sollten Sie nur einmal aufrufen, denn die Synchronisierung läuft bis zum nächsten Fahr- oder Stopbefehl. Jeder weitere SYNCNP-Befehl führt dazu, dass die Synchronisation von vorne beginnt, was normalerweise nicht beabsichtigt ist, außer Sie setzen den aktuellen SYNCERR zurück.

Befehlsgruppe

SYN

Querverweise

Parameter der Gruppe AXS

Syntax-Beispiel

SYNCP

/* Normales Synchronisieren der Position */

CVEL 50

/* Vor dem Synchronisieren Geschwindigkeit erreichen */

CSTART

WAITT 500

SYNCP

■ SYNCSTAT

Folgende Flags sind definiert und können mit SYNCSTAT abgefragt werden: READY, FAULT, ACCURACY und jeweils für Master und Slave die MHIT und MERR.

Kurzinfo

Flag für Synchronisationsstatus abfragen

Syntax

erg = SYNCSTAT

Rückgabewert

erg = Synchronisationsstatus mit folgender Bedeutung:

	Wert	Bit
SYNCREADY	1	0
SYNCFULT	2	1
SYNCCURACY	4	2
SYNCCMMHIT	8	3
SYNCCSMHIT	16	4
SYNCCMMERR	32	5
SYNCCSMERR	64	6

SYNCCURACY

Jede ms wird geprüft ob SYNCERR < SYNCCURACY (55) gilt und falls dies erfüllt ist, wird SYNCCURACY gesetzt, andernfalls wird das Flag zurückgesetzt. Diese Überprüfung findet sowohl bei SYNCP als auch bei SYNCM statt.

Dieses Flag wird nicht bei SYNCPV verwendet.

Beim Ausführen eines SYNCP oder SYNCM Befehles wird dieses Flag zurückgesetzt.

SYNCFULT / SYNCREADY

Bei jedem SYNCP bzw. SYNCM Befehl werden diese Flags zurückgesetzt. Danach wird bei jedem Markerpuls des Slaves (SYNCP) bzw. bei Vorhandensein eines Markerpulses des Masters und eines Markerpulses des Slaves (SYNCM) geprüft, ob SYNCCURACY gesetzt ist oder nicht.

Wenn es gesetzt ist, wird der Ready-Zähler erhöht und der Fault-Zähler auf 0 gesetzt, andernfalls wird der Fault-Zähler erhöht und der Ready-Zähler auf 0 gesetzt.

Ist der Ready-Zähler größer als der im Parameter SYNCREADY (56) vorgegebene Wert, wird das Flag SYNCREADY gesetzt, im anderen Fall wird es zurückgesetzt.

Ist der Fault-Zähler größer als der im Parameter SYNCFULT (57) definierte Wert, wird das Flag SYNCFULT gesetzt, andernfalls wird es zurückgesetzt.

SYNCMMHIT / SYNCSMHIT

SYNCMMHIT und SYNCSMHIT werden gesetzt, wenn der Mastermarker bzw. Slavemarker erkannt werden.

Bei jedem SYNCM Befehl werden diese Flags zurückgesetzt. Danach wird nach dem ersten Auftauchen eines Master-Markerpulses bzw. beim n-ten Markerpuls (Parameter SYNCMARKM 52) das Flag SYNCMMHIT gesetzt.

Analoges gilt für SYNCSMHIT beim Slave.


ACHTUNG!

Dieses Flag wird nicht mehr zurückgesetzt, es sei denn SYNCM wird neu gestartet oder mit dem Befehl SYNCSTATCLR explizit gelöscht.

SYNCMMERR / SYNCSMERR

Wenn im **Marker Window** SYNCMWINM (68) bzw. SYNCMWINS (69) ein Toleranzfenster definiert ist, werden SYNCMMERR bzw. SYNCSMERR gesetzt, sobald der maximal erlaubte Distanz erreicht ist und kein Marker erkannt wurde.

Beispiel:

Markerabstand SYNCMPULSM (58) = 30000

Marker Window SYNCMWINM (68) = 1000

Das Flag wird bei 31000 gesetzt, wenn kein Marker erkannt wurde.

Wenn im **Marker Window** 0 und somit kein Toleranzfenster definiert ist, wird bei jedem Markerpuls (bzw. bei jedem n-ten) geprüft, ob der Abstand zwischen den zwei zuletzt registrierten Markern kleiner ist als das 1,8-fache des im Parameter SYNCMPULSM (58) vorgegebenen **Markerabstandes**.

Wenn nicht, wird das entsprechende Flag gesetzt.

Bei jedem SYNCM Befehl werden diese Flags zurückgesetzt.


ACHTUNG!

Diese Flags werden automatisch zurückgesetzt: bei der nächsten erfolgreichen Markerkorrektur und bei erneutem Start von SYNCM oder durch den Befehl SYNCSTATCLR.

Querverweis

SYNCSTATCLR

Befehlsgruppe

SYN

Syntax-Beispiel

```
IF (SYNCSTAT & 4) THEN OUT 1 1
```

```
/* Wenn ACCURACY dann Ausgang setzen */
```

```
ENDIF
```

■ SYNCSTATCLR

Mit SYNCSTATCLR wert kann man die entsprechenden Bits im SYNCSTAT zurücksetzen und damit die Fehler-Flags MERR und die HIT-Flags MHIT zurücksetzen. Alle anderen Flags können nicht verändert werden.

Kurzinfo

Zurücksetzen der Flags MERR und MHIT

Syntax

SYNCSTATCLR wert

Parameter

wert = 8 = SYNCMMHIT

16 = SYNCSMHIT

32 = SYNCMMERR

64 = SYNCSMERR

Besonderheiten

Der SYNCSTATCLR Befehl sollte nur in einem Unterprogramm zur Fehlerbehandlung eingesetzt werden (siehe ON ERROR GOSUB).


ACHTUNG!

ERRCLR beinhaltet den Befehl MOTOR ON, der die Regelung automatisch wieder einschaltet. (Der Motor wird auf aktueller Position lagegeregelt.)

Befehlsgruppe

SYN

Querverweise

ON STATBIT, ON ERROR GOSUB, ERRNO, CONTINUE, MOTOR ON

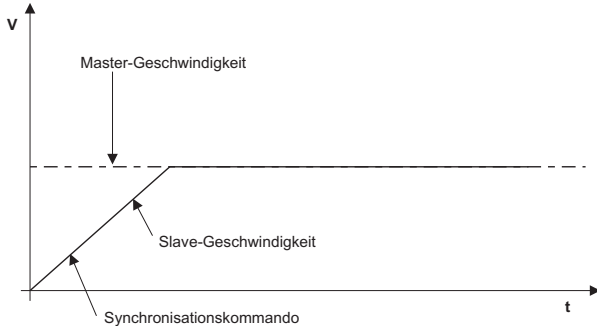
Syntax-Beispiel

```
SYNCSTATCLR 32
```

```
/* aktuelle Fehlermeldung löschen */
```

■ SYNCV

Mit SYNCV wird eine Geschwindigkeitssynchronisation mit dem Master durchgeführt. Dabei wird ausschließlich die Geschwindigkeit betrachtet und nicht versucht, die Position aufzuholen.



Bei der Aufsynchronisation sowie während der Synchronisation wird darauf geachtet, dass weder die voreingestellte Geschwindigkeit VEL noch die voreingestellte Beschleunigung ACC bzw. DEC überschritten wird.

Für die Synchronisation werden die Parameter der Getriebefaktoren verwendet: SYNCFACTM (49), SYNCFACTS (50).

Außerdem werden die Geschwindigkeiten nicht einfach durch Differenz der jetzigen Positionen minus letzte Positionen ermittelt (Master/Slave), sondern die Werte werden entsprechend der Einstellungen SYNCVFTIME (65) gefiltert. Dabei wird der Filter für den Slave aus der Maximalgeschwindigkeit bestimmt. Das heißt:

$VELMAX * 5$ entspricht der Drehgeberauflösung für die Filtertabelle, wobei VELMAX die Geschwindigkeit in qc/ms ist. (Die Formel ergibt sich unter der Annahme, dass die Filtertabelle für die Drehgeberauflösung mit einer Maximalgeschwindigkeit von 3000 U/min gemacht wurde).

Bei dem Übergang von Geschwindigkeitsregler zu Positionsregler wird versucht dies möglichst ruckfrei zu erledigen. Dazu wird die neue Sollposition so definiert, dass gilt:

$$\text{command_pos} = \text{actual_pos} + \text{error}$$

old_error, cvel, avel werden beibehalten. Ebenso wird SUMERR unverändert übernommen.

Kurzinfo

Geschwindigkeitssynchronisation mit dem Master

Syntax

SYNCV


ACHTUNG!

SYNCV sollten Sie nur einmal aufrufen, denn die Synchronisierung läuft bis zum nächsten Fahr- oder Stopbefehl. Jeder weitere SYNCV-Befehl führt dazu, dass die Synchronisation von vorne beginnt, was normalerweise nicht beabsichtigt ist, außer Sie setzen den aktuellen SYNCERR zurück.


ACHTUNG!

Schleppfehler werden im SYNCV Mode nicht überwacht, deshalb wird empfohlen, die Hardware-Encoder-Überwachung zu nutzen.

Befehlsgruppe

SYN

Querverweise

Parameter der Gruppe AXS

■ SYSVAR

Die Systemvariable SYSVAR – ein vorbereitetes Pseudo-Array – können Sie detaillierte Systeminformationen lesen. Diesen Index benötigen Sie auch, wenn Sie mit LINKSYSVAR die Systemvariable mit dem LCP-Display verknüpfen oder Aufzeichnungsdaten einer Testfahrt mit TESTSETP festlegen.

Kurzinfo

Systemvariable (Pseudo-Array) liest Systemwerte

Syntax

SYSVAR[n]

n = index

Befehlsgruppe

CON

Querverweise

LINKSYSVAR, TESTSETP

Systemprozessdaten

Index	Beschreibung
1	Eingangs Byte 0 (I1 ... I8)
2	Eingangs Byte 1 (VLT Eingänge 16 ... 33)
9	Ausgangs Byte 0
17	oberste 2 Byte, die vom SyncPos-Befehl STAT geliefert werden
22	Interner Millisekunden Counter (TIME): Wert den auch der SyncPos Befehl TIME liefert
28	Aktueller Motorstrom [1/100 Amp]
30	Motor Spannung [1/10 V]
31	VLT-Statuswort
32	Aktuelle Ausgangsfrequenz
33	Aktuelle Zeilennummer des SyncPos Programmes, falls mit #DEBUG NOSTOP gearbeitet wurde.
35	Ungefilterter Wert des Analog-Eingangs 1.

Achsprozessdaten

Index	Beschreibung
4096	Aktuelle Position Slave in qc (ohne Umrechnung in BE); vgl. APOS
4097	Sollposition Slave in qc (ohne Umrechnung in BE); vgl. CPOS
4098	Letzte Slave-Indexposition in qc (ohne Umrechnung in BE); vgl. IPOS
4099	Aktuelle Geschwindigkeit in qc/st. Hierbei ist st die mit _GETVEL eingestellte Abtastzeit.
4100	Aktuelle Geschwindigkeit Master (wie oben)
4101	Aktueller Positionsfehler in qc
4102	Enthält nach dem ersten Überlauf des Absolut-Encoders die Anzahl Umdrehungen des Encoders, vorausgesetzt die Strichzahl pro Umdrehung ist in ENCODER (2) richtig eingetragen.
4103	Wie oben für den Master
4105	Aktuelle Position Master ohne Umrechnung (qc) (vgl. MAPOS)
4106	Letzte Master Indexposition ohne Umrechnung (qc) (vgl. MIPOS)
4107	Interne aktuelle Geschwindigkeit (ACTPOS – letzte ACTPOS) (qc/1 ms)
4108	Interne Master-Geschwindigkeit (siehe oben)
4109	Aktuelle Frequenz der Mastersimulation (1/1000 Hz) (siehe PULSVEL)
4110	Gibt an, ob Mastersimulation aktiv ist oder nicht (1 bzw. 0)
4111	Aktueller Sollwert der von der Steuerung durch den Lageregler ausgegeben wird (zwischen -FFFF und FFFF bzw. -1048575 und 1048575 dezimal)

4113	Aktuell verwendete Timer für die PID-Schleife (TIMER)
4114	Aktuell verwendete Timer für den Profil-Generator (PROFTIME)
4115	Gibt an, ob negative Sollwerte ausgegeben werden (!=0) oder nicht (=0)
4116	Gibt an ob Sollwert mit 0-10V und Richtungsausgang ausgegeben wird (>0). Falls ja enthält dieser Parameter die Ausgangsnummer (1..8) die verwendet wird.
4117	Aktuelle Beschleunigung des virtuellen Masters.
4118	Zielfrequenz für virtuellen Master (Einheit siehe oben).
4119	Vlmode (abs / relativ ?)
4120	Anzahl der qc zwischen Index-Pulsen.
4121	Typ des Z-Pulses SYNCMTYPM
4122	Benutzer Sollwert, gegeben durch OUTAN, Skalierung siehe REG_REFERENCE
4123	Slave Encoder Typ (ENCODERTYPE = 0...2)
4124	Master Encoder Typ (MENCODERTYPE = 0...6)
4125	Slave Encoder Auflösung
4126	Master Encoder Auflösung
4127	Gibt an, ob der Verstärker bei Motor OFF auf Stop-Modus (!=0) oder Warten (==0) gesetzt ist..
4128	Liefert die gleiche Information wie MAVEL, jedoch immer die Information des echten Encoders, auch wenn der Master simuliert wird (MENCODERTYPE == 6).

Achsprozessdaten, Profilgenerator Werte

Index	Beschreibung
4218	Gibt alle 32 Flags des Profilgenerators aus. Diese sind: PG_FLAG_BUSY 1L // Flag für Busy Information PG_FLAG_COMMANDERR 2L // Flag für Command Error aufgetreten (wird nicht verwendet) PG_FLAG_POSREACHED 4L // Flag für Position erreicht PG_FLAG_INDEX_HIT 8L // Flag für Index erkannt PG_FLAG_WRAP_OCC 16L // Flag für Wraparound aufgetreten (wird nicht verwendet) PG_FLAG_POS_ERR 32L // Flag für Positionsfehler aufgetreten PG_FLAG_BRKPT_RCHD 64L // Flag für Breakpoint erreicht (wird nicht verwendet) PG_FLAG_FLOATING 128L // Flag für MOTOR OFF

4218	PG_FLAG_MOVING 1L << 8 // Flag für Achse fährt PG_FLAG_OVERFLOW 2L << 8 // Flag für Überlauf der Slave-Position PG_FLAG_OVERFLOWM 4L << 8 // Flag für Überlauf der Masterposition PG_FLAG_POSFLOAT 8L << 8 // Flag für Lageregelung temporär abgeschaltet PG_FLAG_INTERNTST 64L << 8 // Flag zum internen Gebrauch PG_FLAG_SYNCREADY 1L << 24 // Flag für Synchronisation fertig PG_FLAG_SYNCFAULT 2L << 24 // Flag für Synchronisation nicht erfolgt PG_FLAG_SYNCACCUR 4L << 24 // Flag für geforderte Genauigkeit erreicht (Syn) PG_FLAG_SYNCOMMHIT 8L << 24 // Flag für Master-Marker erkannt PG_FLAG_SYNCSMHIT 16L << 24 // Flag für Slave-Marker erkannt PG_FLAG_SYNCOMMERR 32L << 24 // Flag für Markerabstand Master überschritten PG_FLAG_SYNCSMERR 64L << 24 // Flag für Markerabstand Slave überschritten PG_FLAG_TESTFLAG 128L << 24 // Flag zum internen Gebrauch		4226 CSSTART Offset für den Start des Slaves der aktuellen Kurve in qc (ist für geschlossene Kurven immer 0) <hr/> 4227 CCOUNTER Anzahl der Kurven die abgearbeitet werden sollen (letzter SYNCC Befehl) <hr/> 4228 CCURVEPOS Kurvenposition des Slaves in BE Einheiten (aktualisiert in SETCURVE und wenn SYNCCxx aktiv ist) <hr/> 4229 CSLAVECPOSQ Aktuelle Kurvenposition des Slaves in qc (re- lativ zu CSSTART). <hr/> 4230 CMASTERCPOS Aktuelle Kurvenposition des Masters in MU Einheiten. (Wie in SETCURVE initialisiert und nach SYNCCxx aktualisiert.) Siehe auch CURVEPOS <hr/> 4231 GETCMDVEL Liest VEL (intgr part * 128) mit Vorzeichen (siehe auch 4186) <hr/> 4240 PFG_G_STARTKORR Enthält den ersten Korrekturwert nach Start von SYNCM. Dieser gibt an, wie viele Mar- kerfehler vom Startvorgang kompensiert wer- den muss. (Wird gefiltert falls SYNCOFFTIME und SYNCMFTIME gesetzt sind). <hr/> 4241 PFG_G_STARTKORRREST Enthält den Rest des Startkorrekturwertes, der noch abgearbeitet werden muss. Skaliert mit PFG_G_SCALES SHIFT. <hr/> 4242 PFG_G_KORRFILT Enthält den gefilterten Korrekturwert skaliert mit PFG_G_SCALES SHIFT. <hr/> 4243 PFG_G_LASTMMDIST Enthält den zuletzt gemessenen Abstand zwischen zwei Master Marken (qc master) <hr/> 4244 PFG_G_MMARKCORR Enthält den Getriebe Korrekturfaktor der be- rechnet wurde, skaliert mit PFG_G_SCALES SHIFT. <hr/> 4245 PFG_G_KORRUNFILT Enthält den letzten ungefilterten Korrektur- wert (qc - slave). <hr/> 4246 PFG_G_MDISTMARK Enthält den Abstand der aktuellen Master Po- sition vom letzten Master Marker in % des nominellen Markerabstands. <hr/> 4247 PFG_G_SDISTMARK Enthält den Abstand der aktuellen Slave-Po- sition gemessen vom letzten Slave-Marker in % des nominellen Markerabstands. <hr/> 4248 PFG_G_STARTKORRVAL STARTKORRVAL ist der Wert um den bei je- der Marker-Korrektur der Start-Korrekturwert abgebaut wird.
------	---	--	---

 Achsprozessdaten, Kurvenprofil

Index Beschreibung

4220	CINDEX Aktueller Index in der Kurven-Interpolations- Umgebung (Nummer des aktuellen Interpo- lationspunktes 0..Intno-1)		
4221	CVINDEX Aktuell benutzter Index. Der Wert ist gleich CINDEX, wenn weder Start noch Stopp aktiv sind. In diesem Fall zeigt CVINDEX auf die Start- oder Stopp-Kurvendaten.		
4222	CMAXINDEX Maximal erlaubter Index (Intno - 1)		
4223	CIMPS Kurvenposition innerhalb des aktuellen Inter- polations-Intervals (Ganzzahliger Teil des 64-Bit-Wertes)		
4224	CMILEN Länge des Interpolations-Intervals in MU Ein- heiten (Ganzzahliger Teil des 64-Bit-Wertes)		
4225	CWRAP Aktueller Kurvenzähler (0 .. CCOUNTER)		

4249	PFG_G_LASTSMDIST	Zuletzt gemessener Abstand zwischen zwei Slave Marker in qc - Slave.
4250	PFG_G_MARKERFILTER	Tau für den PT Filter um den mittleren Markerabstand zu berechnen.
4251	PFG_G_KORRTAU	Tau für den PT Filter für die Berechnung von PFG_G KORREKTUR
4252	PFG_G_INTMMERROR	Summe der Markerabstandsfehler (aktueller - mittlerer Abstand)
4253	PFG_G_MMARKERR	Gefilterte Summe aller Fehler (skalierter Wert)

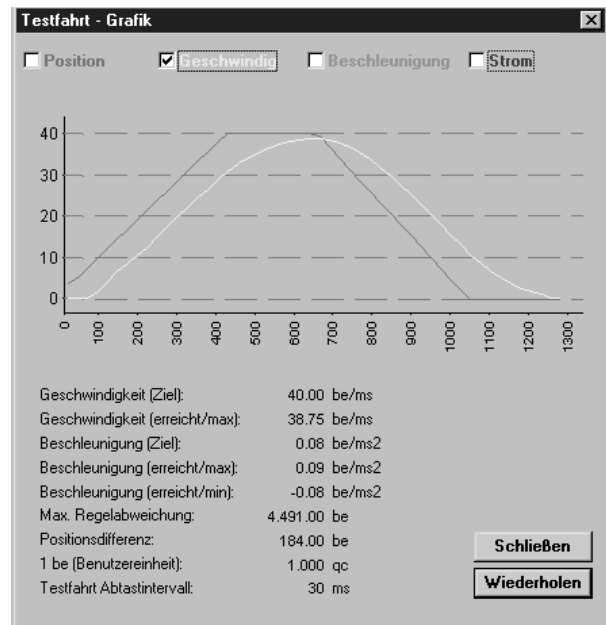


ACHTUNG!

Die Werte der Systemvariablen sind interne, hardware-abhängige Werte, die sich ändern können.

TESTSETP

Standardgemäß können Sie aus dem Menü "TESTFAHRT" eine Testfahrt auslösen, die Soll- und Istposition, Geschwindigkeit, Beschleunigung und Strom aufzeichnet und deren Ergebnis Sie in der Testfahrt-Grafik sehen können:



Mit den beiden Befehlen TESTSETP und TESTSTART können Sie darüber hinaus weitere oder andere Parameter, zum Beispiel die Master-Position aufzeichnen. Und im Gegensatz zu "TESTFAHRT" können Sie diese Daten während der Ausführung eines Programmes aufzeichnen.

Mit TESTSETP legen Sie die Parameter der Aufzeichnung fest (welche Parameter wie oft aufgezeichnet werden sollen und in welches Array) und mit TESTSTART starten Sie dann die Aufzeichnung.

Kurzinfo

Aufzeichnungsdaten für Testfahrt festlegen

Syntax

TESTSETP ms wi1 wi2 wi3 arrayname

Parameter

- ms = Abstand in Millisekunden zwischen zwei Messungen
- wi 1...3 = Indizes der drei Werte, die aufgezeichnet werden sollen. Es gelten die Vereinbarungen für das Systemarray. Es werden immer drei Werte aufgezeichnet.
- arrayname = Name des Arrays, das für die Aufzeichnung verwendet wird.

Array-Format

Innerhalb des Arrays werden die Werte wie folgt gespeichert (alle Werte 4 Byte):

Bezeichnung	Inhalt	Bedeutung
Version	000	Version der Datenstruktur
ms	1	Abstand zwischen zwei Messungen in ms
wi1	i	Wert, der an Stelle 1 aufgezeichnet wurde (Index)
wi2	i	Wert, der an Stelle 2 aufgezeichnet wurde (Index)
wi3	i	Wert, der an Stelle 3 aufgezeichnet wurde (Index)
Anzahl	nn	Gibt an wie viele Messungen folgen
Daten	...	Messdaten
...	...	(insgesamt nn*3)
Anzahl	0-mm	Anzahl Messungen (falls weitere vorliegen)
Daten	...	(s.o.)

ACHTUNG!

Bitte achten Sie darauf, dass die Größe des Arrays für die Aufzeichnung ausreicht.

Für den Header benötigen Sie 6 Elemente, für die Anzahl 1 Element und für jede Messung 3 Elemente. Bei 100 Messungen benötigen Sie also 307 Elemente.

Befehlsgruppe

I/O

Querverweise

TESTSTART, DIM, SYSVAR
"AUFZEICHNUNG ANZEIGEN"

Syntax-Beispiel

```
DIM tstfahrtarray[307]
// Array mit 307 Elementen
TESTSETP 3 4096 4105 4101 tstfahrtarray
// aktuelle Slave-Position, aktuelle Master-Position
// und aktuellen Schleppabstand aufzeichnen
... Positionierfahrt starten ...
TESTSTART 100 // Aufzeichnung starten
```

TESTSTART

Mit diesem Befehl starten Sie die Aufzeichnung einer Testfahrt mit den in TESTSETP definierten Inhalten. Die aufgezeichneten Daten können Sie sich dann auch – soweit sinnvoll – mit "TESTFAHRT" → "AUFZEICHNUNG ANZEIGEN" grafisch darstellen lassen. Dafür stehen die vier Grafiken Position, Geschwindigkeit, Beschleunigung und Strom zur Verfügung.

Kurzinfo

Aufzeichnung starten

Syntax

TESTSTART anz

Parameter

anz = Anzahl der durchzuführenden Messungen


ACHTUNG!

Sollte in einem Array nicht genügend Platz für anz Messungen sein, wird der Fehler O_ERR 71 „Feldgrenzen überschritten“ ausgelöst.

Befehlsgruppe

I/O

Querverweise

TESTSETP, "AUFZEICHNUNG ANZEIGEN"

Syntax-Beispiel

```
SYNCP // Synchronisieren der Position
WAITI 1 ON // Wenn Taste gedrückt wird
TESTSTART 200 // Aufzeichnung starten
// (200 Messungen)
```

Syntax-Beispiel

```
NOWAIT ON
// nicht warten bis Position erreicht ist
VEL 50
POSA 100000
// Positionierung mit Geschwindigkeit 50% starten
WHILE (APOS<50000) DO
// Warten bis Position 50000 erreicht ist
ENDWHILE
VEL 100
// Geschwindigkeit auf 100% erhöhen
TESTSTART 200
// Aufzeichnung starten (200 Messungen)
DELAY 20 // 20 ms warten
POSA 100000
// Positionierung mit neuer Geschwindigkeit starten
NOWAIT OFF
// Warten bis Positionierung zu Ende
```

■ TIME

Mit dem TIME Befehl kann die interne Systemzeit ausgelesen werden. Der TIME Befehl eignet sich vor allem, um die Ausführungszeit einer Befehlssequenz oder Maschinenzykluszeiten zu berechnen.

Kurzinfo

Systemzeit auslesen

Syntax

erg = TIME

Rückgabewert

erg = Systemzeit in Millisekunden seit Einschalten



ACHTUNG!

Bitte beachten Sie, dass der Wert, wenn er MLONG erreicht hat, auf -MLONG wechselt.

Befehlsgruppe

I/O

Syntax-Beispiel

PRINT TIME

/* aktuelle Systemzeit ausgeben */

timestop1 = TIME

/* aktuelle Systemzeit zwischenspeichern */

Programmbeispiele

ACC_01.M, DELAY_01.M, EXIT_01.M, GOSUB_01.M

■ TRACKERR

Mit TRACKERR können Sie den aktuellen Schleppabstand der Achse in Benutzereinheiten (unter Berücksichtigung des Vorzeichens) abfragen. Dies ist die Differenz zwischen Sollwert (CPOS) und aktueller Position (APOS).

Kurzinfo

aktuellen Schleppabstand einer Achse abfragen

Syntax

erg = TRACKERR

Rückgabewert

erg = aktueller Schleppabstand der Achse in BE

Befehlsgruppe

I/O

Querverweise

APOS, CPOS

POSERR (15)

Syntax-Beispiel

PRINT TRACKERR

/* aktuellen Schleppabstand der Achse abfragen */

■ VEL

Mit dem Befehl VEL wird die Geschwindigkeit für die nächsten absoluten und relativen Positioniervorgänge und die maximal zulässige Geschwindigkeit für Synchronisationsvorgänge bestimmt. Der Wert bleibt solange gültig, bis mit einem weiteren VEL Befehl eine neue Geschwindigkeit gesetzt wird.

Der Geschwindigkeitswert wird zu den Parametern **Maximalgeschwindigkeit** VELMAX (1) und **Geschwindigkeitsteiler** VELRES (22) in Bezug gesetzt. Wenn der übergebene Geschwindigkeitswert gleich der Normierungsgröße ist, wird mit der im Parameter VELMAX (1) festgelegten Drehzahl verfahren.

Anmerkung: Die Geschwindigkeit des Slaves wird im Synchronisationsmodus auch durch den VEL Befehl begrenzt.

Kurzinfo

Geschwindigkeit für relative und absolute Bewegungen sowie die maximal zulässige Geschwindigkeit bei Synchronisationsvorgängen setzen

Syntax

VEL v

$$\text{Sollgeschwindigkeit (U/min)} = v \cdot \frac{\text{VELMAX (1)}}{\text{VELRES (22)}}$$

Parameter

v = Normierter Geschwindigkeitswert



ACHTUNG!

Wurde vor einem Positionier- oder Synchronisierbefehl noch keine Geschwindigkeit definiert, wird mit dem in Parameter DFLTVEL (33) festgelegten Wert verfahren.

Soll während des Positioniervorganges die Geschwindigkeit geändert werden, ist dies bei NOWAIT ON möglich, wenn dem VEL Befehl nochmal ein POSA auf die gewünschte Zielposition folgt. Die maximal zulässige Geschwindigkeit kann jederzeit mit dem Befehl VEL geändert werden, wenn dem Befehl VEL nochmals ein SYNCV, SYNCP oder SYNCM folgt.

Befehlsgruppe

REL, ABS

Querverweis

ACC, POSA, POSR

Parameter: VELMAX (1)

Syntax-Beispiel

VEL 100 /* Geschwindigkeit 10 */

Programmbeispiele

VEL_01.M

■ WAITAX

Der WAITAX Befehl ist für die Verwendung bei aktivem NOWAIT Modus vorgesehen. Mit diesem Befehl wird im NOWAIT ON der Zustand erreicht, dass nach einem Positionierbefehl mit der weiteren Programmabarbeitung gewartet wird, bis die Achse ihre Sollposition erreicht hat.

Kurzinfo

Warten bis Zielposition erreicht ist

Syntax

WAITAX

Befehlsgruppe

CON

Querverweise

NOWAIT ON/OFF, POSA, POSR, AXEND, STAT, WAITI

Syntax-Beispiel

WAITAX

/* Warten bis die Achse Bewegung beendet hat */

WAIT AX

/* Alternative Schreibweise */

Programmbeispiele

WAIT_01.M, VEL_01.M

■ WAITI

Der WAITI Befehl wartet mit der weiteren Programmausführung, bis der entsprechende Eingang den gewünschten Signalzustand aufweist.

Kurzinfo

Warten auf bestimmten Eingangszustand

Syntax

WAITI n s

Parameter

n = Nummer des Eingangs 1 – 8 oder 16 – 33

s = erwarteter Zustand

ON = High-Signal anliegend

OFF = Low-Signal anliegend



ACHTUNG!

Wenn der erwartete Eingangszustand nie auftritt, bleibt das Programm an diesem Befehl „hängen“. Für das sichere Erkennen eines Signalzustandes ist eine minimale Signallänge notwendig! Informieren Sie sich über die Beschaltung und technischen Daten der Eingänge im Kapitel Ein- und Ausgangsklemmen und in der VLT5000 Bedienungsanleitung.

Befehlsgruppe

CON

Querverweis

ON INT GOSUB, DELAY, WAITT, WAITAX

Kapitel Ein- und Ausgangsklemmen

Syntax-Beispiel

WAITI 4 ON

/* Warten bis an Eingang 4 High-Pegel anliegt */

WAITI 4 1 /* 3 alternative Schreibweisen: */

WAIT I 4 ON

WAIT I 4 1

WAITI 6 OFF

/* Warten bis an Eingang 6 Low-Pegel anliegt */

WAITI 6 0 /* 3 alternative Schreibweisen: */

WAIT I 6 OFF

WAIT I 6 0

Programmbeispiele

WAIT_01.M

■ WAITNDX

Warten auf Index mit Timeout Überprüfung. Es wird so lange gewartet, bis entweder der Index der Achse gefunden oder die Zeit (Timeout) überschritten wurde.

Kurzinfo

Warten auf Index

Syntax

WAITNDX t

Parameter

t = Timeout (maximale Wartezeit) in ms



ACHTUNG!

Wenn die Zeit überschritten wurde, wird ein Fehler ausgelöst, der zum Beispiel mit einer ON ERROR Funktion ausgewertet werden kann.

Befehlsgruppe

CON

Querverweise

WAITI, WAITP, INDEX

Syntax-Beispiel

CVEL 1

CSTART

WAITNDX 10000

/* Wartet maximal 10 sec darauf, dass die Achse die Indexposition erreicht */

OUT 1 1

■ WAITP

Der WAITP Befehl bewirkt, dass mit der weiteren Programmausführung gewartet wird, bis die Position p erreicht ist.

Wenn aus der Geschwindigkeit und der aktuellen Position hervorgeht, dass der Punkt p bereits überschritten wurde, wird der Befehl ebenfalls beendet.

Kurzinfo

Warten auf Position

Syntax

WAITP p

Parameter

p = absolute Position auf die gewartet wird



ACHTUNG!

Aktive ON INT oder ON PERIOD Befehle können die Genauigkeit und Reproduzierbarkeit beeinflussen.

Befehlsgruppe

CON

Querverweise

DELAY, WAITI, WAITAX

Syntax-Beispiel

```
NOWAIT ON
POSA 100000
WAITP 50000 /* Warten bis 50000 erreicht ist */
OUT 1 1 /* Ausgang 1 setzen */
NOWAIT OFF
```

■ WAITT

Mit dem WAITT Befehl können Sie eine definierte Programmverzögerung erzielen. Der Übergabeparameter gibt dabei die Verzögerungszeit in Millisekunden an.

Kurzinfo

Zeitverzögerung

Syntax

WAITT t

Parameter

t = Verzögerungszeit in Millisekunden (maximal MLONG)



ACHTUNG!

Wenn während der Verzögerungszeit ein Interrupt auftritt, wird nach der Abarbeitung der Interrupt-Prozedur der komplette Wartevorgang von neuem begonnen.

Im allgemeinen sollte statt WAITT der DELAY Befehl wegen seines konstanten Zeitverhaltens verwendet werden!

Befehlsgruppe

CON

Querverweise

DELAY, WAITI, WAITAX

Syntax-Beispiel

```
WAITT 5000 /* 5000 Sekunden warten */
WAIT T 5000 /* Alternative Schreibweise */
```

Programmbeispiel

WAIT_01.M

■ WHILE .. DO .. ENDWHILE

Mit der WHILE .. DO .. ENDWHILE Konstruktion kann man den eingeschlossenen Programmbereich in Abhängigkeit von einem beliebigen Kriterium ein- oder mehrfach wiederholen.

Das Schleifenkriterium setzt sich aus einer oder mehreren Vergleichsoperationen zusammen und wird stets am Schleifenanfang überprüft, was bei einem negativen Ergebnis bereits bei der ersten Überprüfung dazu führen kann, dass die Befehle innerhalb der Schleife nicht ausgeführt werden und das Programm sofort nach der ENDWHILE Anweisung fortgesetzt wird.

Kurzinfo

Bedingte Schleife mit Überprüfung des Abbruchkriteriums am Schleifenanfang (Während Bedingung erfüllt, wiederhole ...)

Syntax

WHILE Bedingung DO
ENDWHILE

Parameter

Bedingung = Abbruchkriterium

 **ACHTUNG!**

In Abhängigkeit von dem Schleifenkriterium kann es vorkommen, dass der Schleifeninhalt nicht abgearbeitet wird. Um eine Endlosschleife zu vermeiden, müssen die innerhalb der Schleife abgearbeiteten Befehle direkt oder indirekt Einfluss auf das Ergebnis der Abbruchüberprüfung haben.

Befehlsgruppe

CON

Querverweise

LOOP, REPEAT ..UNTIL..

Syntax-Beispiel

WHILE (A != 1 AND B == 0) DO

Befehlszeile 1

Befehlszeile n

ENDWHILE

Programmbeispiele

WHILE_01.M, INKEY_01.M

■ _GETVEL

Mit dem _GETVEL Befehl können Sie die Abtastzeit für AVEL und MAVEL verändern. AVEL und MAVEL arbeiten standardgemäß mit einer Abtastzeit von 20 ms, dadurch ist die Auflösung besser. Allerdings liegt nur alle 20 ms ein neuer Wert vor.

Der Befehl _GETVEL dauert genauso lange wie der zugewiesene Wert, z.B. dauert _GETVEL 200 ca. 200 ms.

Kurzinfo

Abtastzeit für AVEL und MAVEL verändern

Syntax

var = _GETVEL t

Anzeige der Werte in BE/sec bei AVEL bzw.

qc/sec bei MAVEL.

Parameter

t = Abtastzeit in Millisekunden

Befehlsgruppe

I/O

Querverweise

AVEL, MAVEL

Syntax-Beispiel

var = _GETVEL 200

Damit wird die Messauflösung wesentlich besser, allerdings erhält man Veränderungen immer erst mit einer Verzögerung von 200 ms.

Programmbeispiele

GSVEL_01.M, DIM_01.M

■ #DEBUG

Mit Einführung des Debug-Modus in der PC-Benut-

zeroberfläche werden die #DEBUG Befehle ab
VLT5000/SyncPos Software Version 3.xx/4.2x

VLT5000/FluxSyncPos Software Version 5.xx/4.2x
nicht mehr ausgeführt.

Vorhandene #DEBUG Befehle müssen nicht unbedingt aus dem Programm entfernt werden, da sie ignoriert werden.

Querverweise

Menü "ENTWICKLUNG"

→ "VORBEREITEN EINZELSCHRITT"

→ "START"

→ "EINZELSCHRITT"

→ "BEENDEN DEBUG"

■ #INCLUDE

#INCLUDE ist eine Anweisung für den Compiler, während der Übersetzung eines Programmes in steuerungsspezifische Befehle an der entsprechenden Programmposition den Inhalt der angegebenen Datei einzufügen. Die #INCLUDE Anweisung ist also kein echter Befehl, der eine Reaktion innerhalb der SyncPos-Option auslöst, sondern eine Anweisung für das Übersetzungsprogramm, also eine Compiler-Direktive.

#INCLUDE kann an jeder beliebigen Programmposition und auch mehrfach innerhalb eines Programmes eingesetzt werden. Es muss jedoch darauf geachtet werden, dass die in der einzubindenden Datei enthaltenen Befehle auch an der aktuellen Programmposition verwendet werden dürfen und dass der Befehlsaufbau korrekt ist.

Die #INCLUDE Anweisung eignet sich vor allem, um häufig benötigte Unterprogramme in separaten Dateien abzuspeichern und innerhalb des Bereichs SUBMAINPROG .. ENDPORG in die Anwendung zu integrieren.

Kurzinfo

Einfügen des Inhalts einer Datei an der aktuellen Programmposition

Syntax

#INCLUDE datei

Parameter

datei = Vollständiger Name der einzubindenden Datei (Pfadangaben sind unzulässig)

**ACHTUNG!**

Die einzubindende Datei muss sich im aktuellen Verzeichnis befinden. Der angegebene Dateiname muss die Endung „.M“ haben. Die innerhalb der einzubindenden Datei enthaltenen Befehle müssen eine korrekte Syntax aufweisen.

Befehlsgruppe

CON

Syntax-Beispiel

```
#INCLUDE INC_UP01.M
/* Inhalt von Datei INC_UP01.M einfügen */
```

Programmbeispiele

```
INCL_01.M + INCSTA01.M + INCPOS01.M +
INCIN01.M
```

■ Parameter-Referenz
■ VLT-Parameter und SyncPos-Parameter

Grundsätzlich unterscheiden wir zwischen VLT-Parametern und Parametern des SyncPos Motion Controllers.

Die Parameter der Gruppe 7.. sind sowohl VLT- als auch SyncPos-Parameter.

Alle VLT-Parameter mit Ausnahme der Gruppen 7., 8. und 9.. werden in der VLT5000/VLT5000Flux Betriebsanleitung beschrieben.

Im folgenden Abschnitt wird erklärt, wie Sie die VLT Parameter inklusive der Gruppe 7.. und die SyncPos-Parameter ändern.

Die VLT-Parameter der Gruppen 8.. und 9.. sind nur verfügbar, wenn eine Feldbus-Option installiert ist. Sie finden die notwendigen Informationen in den entsprechenden Handbüchern (Feldbus-Option).

Werkseinstellungen der Parameter

... sind die Werte, die ab Werk im SyncPos Motion Controller fest gespeichert sind. Sie sind nach der Auslieferung aktiv und können vom Benutzer verändert werden.

Sie können die VLT-Parameter auf die Werkseinstellung durch eine manuelle Initialisierung des VLT's oder mit der Funktion "KOPIEREN VON PARAMETERSÄTZEN" zurücksetzen. Alle Details finden Sie in der VLT5000/VLT5000Flux Betriebsanleitung.

Auch die Werkseinstellungen der SyncPos-Parameter können Sie jederzeit wieder einstellen: Entweder im Menü "STEUERUNG" → "RESET" → "PARAMETER" oder → "VOLLSTÄNDIG". Ebenso erhalten Sie wieder die Werkseinstellungen, wenn Sie mit "STEUERUNG" → "SPEICHER" → das "EEPROM LÖSCHEN".

Benutzerparameter

Die Parameter 710 bis 779 und 795 bis 799 sind Benutzerparameter, die vom Anwender definiert und in einem SyncPos Programm benutzt werden können.

Die Benutzerparameter werden mit den Befehlen LINKXPAR und LINKGPAR definiert. Dabei können ein Parametername für die Darstellung im LCP, der Eingabebereich und die Art der Parameterausführung festgelegt werden:

0 = Online führt die Änderung des Parameters sofort nach der Eingabe im Display aus.

1 = Offline führt die Änderung des Parameters erst aus, wenn die OK-Taste gedrückt wird.

Ein Benutzerparameter kann mit einem globalen oder Achsparameter der SyncPos-Option verknüpft werden. Es gibt zwei Arten von globalen Parametern:

- Vordefinierte globale Parameter wie I_ERRCLR (107)
- Freie interne Parameter, auf die Sie über das SyncPos-Programm zugreifen können. Dafür werden die Parameter 130 bis 229 genutzt.

Syntax Beispiele:

```
/* Vordefinierter globaler Parameter */
LINKGPAR I_ERRCLR 710 " " 0 33 0
/* Freier interner Parameter */
LINKGPAR 132 711 "name" 0 100000 0
/* Achsparameter */
LINKXPAR POSERR 712 "position error" 0 100000 0
```

Die Parameter 795–799 sind Read-only-Parameter, die eingesetzt werden können, um Daten am LCP auszulesen. Diese Parameter können in der ersten oder zweiten Displayzeile ausgelesen werden. Welcher der Parameter 795–799 an welcher Stelle des VLT-Displays angezeigt wird, können Sie in den VLT-Parametern 9, 10, 11 und 12 auswählen:

- Benutzerparameter 95 [90]
- Benutzerparameter 96 [91]
- Benutzerparameter 97 [92]
- Benutzerparameter 98 [93]
- Benutzerparameter 99 [94]

Ändern und Speichern der Parameter

VLT-Parameter, die Sie über das LCP-Display und SyncPos-Parameter, die Sie im Menü "STEUERUNG" → "PARAMETER" → "ACHSEN" ändern, werden im EEPROM gespeichert und bleiben auch beim Ausschalten erhalten.

VLT-Parameter, die Sie im SyncPos-Programm mit dem SETVLT Befehl ändern, werden nur im RAM gespeichert und sind daher nach dem Ausschalten verloren.

SyncPos-Options-Parameter, die mit dem SET Befehl des SyncPos-Programmes geändert werden, sind nur während der Programmausführung aktiv. Sie können diese Änderungen aber auch im EEPROM speichern, so dass sie auch nach dem Ausschalten erhalten bleiben: Entweder mit dem Befehl SAVEPROM oder durch Drücken der Taste OK am VLT-Display.


ACHTUNG!

Bitte beachten Sie, dass ein EEPROM begrenzte Lebensdauer hat; nach heutiger Kenntnis kann man es aber schon ca. 10000 mal (mit steigender Tendenz) reprogrammieren.

■ VLT Parameter

- 0..
- 1..
- 2..
- 3..
- 4..
- 5..
- 6..
- 7..
- 8..
- 9..

Die Parameter 009, 010, 011 and 012 (Zeile 1 und 2 im Display) haben 5 weitere Auswahlmöglichkeiten:

	VLT5000	VLT5000Flux
Benutzerparameter 95	[90]	[90]
Benutzerparameter 96	[91]	[91]
Benutzerparameter 97	[92]	[92]
Benutzerparameter 98	[93]	[93]
Benutzerparameter 99	[94]	[94]

Parameter 115 (Schlupfausgleich): Änderung der Werkseinstellung von 100% auf 0%.

Änderung der Werkseinstellungen: Von auf

Parameter 200	nur im Uhrzeigersinn 132Hz [0]	beide Richtungen 132 Hz [1]
Parameter 203	Min ... Max [0]	-Max ... +Max [1]
Parameter 207	Leistungsabhängig	0,05 Sek.
Parameter 208	Leistungsabhängig	0,05 Sek.

Werkseinstellungen VLT5000:

Parameter 300...303, 305...308, 314 geändert auf:	"Keine Funktion"	[0]
Parameter 319 + 321 geändert auf:	"Option 0 ... 20 mA"	[91]
Parameter 323 + 326 geändert auf:	"Steuerwort bit 11/12"	[33]

4 zusätzliche Auswahlmöglichkeiten für Parameter 319 + 321:

	Option digital	[90]
	Option 0 ... 20 mA	[91]
	Option 4 ... 20 mA	[92]
	Option 0 ... 32000 Puls	[93]

Werkseinstellungen VLT5000Flux:

Geänderte Werkseinstellungen:

Parameter 300...303, 305...308, 314 geändert auf:	"keine Funktion"	[0]
Parameter 319 + 321 geändert auf:	"Option 0 ... 20mA"	[90]
Parameter 323 + 326 geändert auf:	"Steuerwort bit11/12"	[33]
Parameter 341 + 355 geändert auf:	"Option digital"	[90]

2 zusätzliche Auswahlmöglichkeiten in Parameter 319 + 321:

	Option 0 ... 20mA	[90]
	Option 4 ... 20mA	[91]

2 zusätzliche Auswahlmöglichkeiten in Parameter 341 + 355:

	Option digital	[90]
	Option Puls-Ausgang	[91]

Nur verfügbar, wenn eine Feldbus-Option installiert ist (siehe Handbuch Feldbus-Option).

Gruppe 7: nächste Seite

Gruppe 7:

VLT Par. # LCP Zugriff	Optionsparameter # und Name	Parameter Lesen/Schreiben Anwendungsprogramm
700	–	GETVLT/SETVLT Parameter Nummer Ex.: var = GETVLT 700
701 702 703 704 705 706 707 708 709	102 PRGPAR 11 KPROP 12 KDER 13 KINT 21 KILIM 35 BANDWIDTH 36 FFVEL 37 FFACC 65 SYNCVFTIME	GET/SET Parameter Name Ex.: var = GET KPROP
710-779	130 – 229	GET/SET Parameter Nummer Ex.: var = GET 131
780 781	– – Programm-ID	GETVLT/SETVLT Parameter Nummer Ex.: var = GETVLT 780
795-799	130 – 229	GET/SET Parameter Nummer Ex.: var = GET 132


ACHTUNG!

Generell ist für ein gutes Steuerverhalten die Optimierung der VLT-Parameter auf den Motor, z.B. mit Hilfe von AMA, sehr wichtig ist.

Parameter 205

Parameter 205, Maximalsollwert, muss in Anpassung an die maximale Drehzahl (SyncPos-Options-Parameter VELMAX (1)) eingestellt werden, bevor die Steuerparameter optimiert werden.

System Control 700

Wenn "SYNCPOS EIN" = 0 ausgewählt ist, wird der VLT durch den SyncPos Motion Controller gesteuert. Wenn "SYNCPOS AUS" = 1 ausgewählt ist, wird der VLT nicht von der SyncPos-PID-Regelung gesteuert. Dann können Sie den VLT „traditionell“ über die Eingänge oder eine serielle Schnittstelle steuern. Als dritte Einstellung kann man "ENABLE SP. W/O MONI" wählen und damit den Fehler 13 (Status Fehler VLT NICHT BEREIT) vermeiden.

Inhalt

Steuerung durch Optionskarte oder nicht

Bereich

0 ... 2 ★ 0
 0 = Enable SyncPos, SyncPos EIN
 1 = Disable SyncPos, SyncPos AUS
 2 = Enable SyncPos ohne VLT Überwachung

D.TIME CMP.ACT. 780

Der **aktivierte Totzeitausgleich** – der Bestandteil des VLT5000 Kontrollalgorithmus (VCC+) ist – verursacht Instabilität im Stillstand, wenn mit geschlossener Regelschleife gearbeitet wird. Um solche Instabilitäten zu vermeiden, schalten Sie mit dem Parameter 780 den **Aktiven Totzeitausgleich** aus. Wählen Sie AUS, wenn der VLT vom SyncPos Motion Controller für Drehzahl-, Positions- oder Synchronisationsmodus gesteuert wird. Wählen Sie EIN, wenn der VLT im Drehzahlmode mit offener Regelschleife gesteuert wird. Der aktivierte Totzeitausgleich sorgt dann für eine bessere Leistungssteuerung.

Inhalt

aktivierter Totzeitausgleich

Kompatibilität

In VLT5000Flux nicht verfügbar

Bereich

0 ... 1 ★ 0
 0 = AUS
 1 = EIN

Programm ID 781

Bei kundenspezifischen Programmen, die ab Werk geladen werden, muss diese Typencode-Nummer in Parameter 781 eingegeben werden, d.h. die folgende Zeile muss am Anfang des SyncPos-Anwendungsprogramms stehen:

```
SETVLT 781 xy
Wobei xy der Typen-Code ist (Byx).
```

Der Wert des Parameters 781 wird während des Endtests in der Produktion ausgelesen und mit dem Typencode des Antriebs verglichen, um zu überprüfen, dass das Programm korrekt installiert wurde.

Inhalt

Typencode-Nummer.

Bereich

0 ... 1000 ★ 0

■ VLT Parameterliste

PNU	Parameter Beschreibung	Werks-einstellung	Bereich	Ändern während des Betriebes	4-Setup	Konvertierungs-index	Daten typ
700	System control	0	0 ... 1	Ja	Nein	0	6
701	Programm Nummer	-1	-1 ... 127	Ja	Nein	0	4
702	PID, Proportional Faktor	30	0 ... 65000	Ja	Nein	0	4
703	PID, Derivative Faktor	0	0 ... 65000	Ja	Nein	0	4
704	PID, Integral Faktor	0	0 ... 65000	Ja	Nein	0	4
705	PID, Integral Bandbreite	1000	0 ... 1000	Ja	Nein	0	4
706	PID, BANDWIDTH	1000	0 ... 65000	Ja	Nein	0	4
707	PID, Geschwindigkeits-Feed-forward	0	0 ... 65000	Ja	Nein	0	4
708	PID, Beschleunigungs-Feed-forward	0	0 ... 65000	Ja	Nein	0	4
709	PID, Geschwindigkeitsfilter	0	-500000 ... 500000	Ja	Nein	0	4
710	Benutzerparameter 10	0	Anwenderdefiniert ¹	Ja	Nein	0	4
711	Benutzerparameter 11	0	Anwenderdefiniert ¹	Ja	Nein	0	4
...							
778	Benutzerparameter 78	0	Anwenderdefiniert ¹	Ja	Nein	0	4
779	Benutzerparameter 79	0	Anwenderdefiniert ¹	Ja	Nein	0	4
780	Aktiver Totzeitausgleich (in VLT5000Flux nicht verfügbar)	AUS	0 ... 1	Nein	Nein	0	6
781	Programm ID	0	0...1000	Nein	Nein	0	6
795	Benutzerparameter 95 (nur lesen)	0	Anwenderdefiniert ¹	Nur lesen	Nein	0	4
796	Benutzerparameter 96 (nur lesen)	0	Anwenderdefiniert ¹	Nur lesen	Nein	0	4
797	Benutzerparameter 97 (nur lesen)	0	Anwenderdefiniert ¹	Nur lesen	Nein	0	4
798	Benutzerparameter 98 (nur lesen)	0	Anwenderdefiniert ¹	Nur lesen	Nein	0	4
799	Benutzerparameter 99 (nur lesen)	0	Anwenderdefiniert ¹	Nur lesen	Nein	0	4

Ändern während des Betriebes

Ja Der Parameter ändern kann geändert werden, während der VLT Frequenzrichter in Betrieb ist.
 Nein Der VLT Frequenzrichter muss angehalten werden, um einen Parameter zu ändern.

4-Setup Parametersätze

Ja Der Parameter kann in jedem der 4 Parametersätze individuell programmiert werden, so dass der gleiche Parameter 4 verschiedene Datenwerte aufweisen kann.
 Nein Der Datenwert des Parameters in jedem der vier Parametersätze ist gleich.

Konvertierungsindex

Der Konvertierungsindex wird benutzt, um Parameterwerte mit Dezimalstellen seriell zu übertragen. Bei Konvertierungsindex 0 wird der Parameterwert mit 1 multipliziert.

Datentyp

Bezeichnet den Datentyp des Parameterwertes:
 4 = Integer 32 bit
 6 = vorzeichenlos 16 bit

1) Der maximale Bereich ist -2^{32} bis $2^{32}-1$ (-2147483648 bis 2147483647), aber der aktuelle Bereich wird festgelegt, wenn man die Benutzerparameter im Anwendungsprogramm mit LINKGPARG or LINKXPAR definiert.

■ Alle SyncPos-Parameter in alphabetischer Übersicht

Die Parameter für das SyncPos-Programm sind der besseren Übersicht wegen auch in Gruppen eingeteilt:

Globale Parameter GL

Die Programmiersprache SyncPos und die Benutzeroberfläche sind bereits für Mehrachsen-Anwendungen vorbereitet. Daher sind alle achsunabhängigen Parameter wie die **Aktivierte Programmnummer** PRGPAR (102) und die I/O-Parameter wie **Fehler löschen** I_ERRCLR (107) in der Gruppe globale Parameter GL zusammengefasst.

Achsparameter AX...

Gruppe der achsspezifischen Parameter, die mit den Befehlen GET und SET bearbeitet werden:

- AXE Achsparameter Encoder
- AXH Achsparameter Homefahrt
- AXI Achsparameter Eingänge/Ausgänge (Ein-/Ausgangskonfiguration)
- AXR Achsparameter Regelung (Alles was zum Regelverhalten gehört).
- AXS Achsparameter Synchronisierung
- AXV Achsparameter Geschwindigkeit (Velocity) (Geschwindigkeits- und Beschleunigungswerte)

Die Spalte Parametergruppe (AXV, AXR etc.) verweist auf die Dialogfenster "STEUERUNG" → "PARAMETER" → "ACHSEN", bzw. → "GLOBAL" in denen Sie die Parameter setzen und ändern können.

Parameterkennung	Parameterbezeichnung	VLT- Par.Nr.	SyncPos Par.-Nr.	Par.- Gruppe	Einheit	Werks- einstellg.
ACCMAXQC	Maximalbeschleunigung		10	AXV	qc/ms ² *	1/65536
BANDWIDTH	Bandbreite innerhalb der der PID-Filter aktiv ist	706	35	AXR	%	1000
DFLTACC	Default-Beschleunigung		34	AXV	%	50
DFLTVEL	Default-Geschwindigkeit		33	AXV	%	50
ENCODER	Auflösung des Istwertgebers (Geberstrichzahl)		2	AXE	Strich- zahl/Umdr.	500
ENCODERTYPE	Encodertyp für Encoder 0 (Slave)		27	AXE	-	0
ENDSWMOD	Verhalten bei Endschalter		44	AXI	-	0
ERRCOND	Verhalten im Fehlerfall		43	AXI	-	0
ESCCOND	Verhalten bei Programmabbruch		70	AXI	-	0
FFACC	Beschleunigungs-Feed-forward	708	37	AXR	%	0
FFVEL	Geschwindigkeits-Feed-forward	707	36	AXR	%	0
HOME_FORCE	Homefahrt erzwingen?		3	AXH	-	0
HOME_OFFSET	Nullpunkt-Offset bezügl. Maschinennullpunkt (MN) bzw. Index		42	AXH	qc	0
HOME_RAMP	Rampe für Homefahrt		41	AXH	%	10
HOME_VEL	Geschwindigkeit für Homefahrt		7	AXH	%	10
HOMETYPE	Homefahrt Verhalten		40	AXH	-	0
I_BREAK	Eingang für Abbruch		105	GLI	-	0
I_CONTINUE	Programm fortsetzen		106	GLI	-	0
I_ERRCLR	Fehler löschen		107	GLI	-	0
I_NEGLIMITSW	Endschalter negativ		47	AXI	-	0
I_POSLIMITSW	Endschalter positiv		46	AXI	-	0
I_PRGCHOICE	Eingang für Programmwahl Anfang		104	GLI	-	0
I_PRGSTART	Eingang für Programmstart		103	GLI	-	0
I_REFSWITCH	Referenzschalter Eingang		45	AXI	-	0
KDER	Differentialwert für PID-Regelung	703	12	AXR	-	1
KILIM	Grenzwert für Integralsumme für PID-Regelung	705	21	AXR	-	0
KINT	Integralwert für PID-Regelung	704	13	AXR	-	0
KPROP	Proportionalwert für PID-Regelung	702	11	AXR	-	30

Programmierbarer SyncPos Motion Controller

Parameterkennung	Parameterbezeichnung	VLT- SyncPos		Par.- Gruppe	Einheit	Werks- einstellg.
		Par.Nr.	Par.-Nr.			
MENCODER	Auflösung des Encoder 1 (Master)		30	AXE	Strich- zahl/Umdr.	500
MENCODERTYPE	Encodertyp für Master-Encoder		67	AXE	-	0
NEGLIMIT	Negativer Software-Endschalter		4	AXI	qc	-500000
O_AXMOVE	Ausgang für Fahrbefehl in Aktion		64	AXI	-	0
O_BRAKE	Ausgang für mechanische Bremse		48	AXI	-	0
O_ERROR	Ausgang für Fehler		108	GLS	-	0
POSDRCT	Drehrichtung		28	AXE	-	1
POSERR	Maximal tolerierter Positionsfehler		15	AXR	qc	2000
POSFAC_T_N	Nenner Benutzerfaktor		26	AXE	-	1000
POSFAC_T_Z	Zähler Benutzerfaktor		23	AXE	-	1000
POSLIMIT	Positiver Software-Endschalter		5	AXI	qc	500000
POSUNITS	Einheit für Positionsanzeige (BE)		29	AXE	-	0
PRGPAR	Aktiviere Programmnummer	701	102	GLS	-	-1
PROFTIME	Abtastzeit für Profilgenerator		29	AXR	ms	1000
RAMPMIN	Kürzeste Rampe		31	AXV	msec	1000
RAMP_TTYPE	Rampenform		32	AXV		0
REGWMAX	Größe des Regelfensters (Aktivierung)		38	AXR	qc	0
REGWMIN	Größe des Regelfensters (Deaktivierung)		39	AXR	qc	0
REVERS	Reversierungsverhalten Slave		63	AXR	-	0
SWNEGLIMACT	Negativer Software-Endschalter aktiv		19	AXI	-	0
SWPOSLIMACT	Positiver Software-Endschalter aktiv		20	AXI	-	0
SYNCACCURACY	Größe des Genauigkeitsfensters für Positionssynchronisation		55	AXS	qc	1000
SYNCFACTM	Synchronisationsfaktor Master (M:S)		49	AXS	qc	1
SYNCFACTS	Synchronisationsfaktor Slave (M:S)		50	AXS	qc	1
SYNCFAULT	Markeranzahl für Fault		57	AXS	-	10
SYNCMARKM	Markeranzahl Master		52	AXS	-	1
SYNCMARKS	Markeranzahl Slave		53	AXS	-	1
SYNCMFTIME	Definiert die Filterzeit für die Markerkorrektur		18	AXS	1 ms	0
SYNCMFPAR	Markerfilter Konfiguration		17	AXS	-	0
SYNCMMAXCORR	Begrenzt die max. Korrektur der Markerkorrektur		6	AXS	qc	0
SYNCMPULSM	Markerabstand Master		58	AXS	qc	500
SYNCMPULSS	Markerabstand Slave		59	AXS	qc	500
SYNCMSTART	Startverhalten für Markersynchronisation		62	AXS	-	0
SYNCMTYPM	Markertyp Master		60	AXS	-	0
SYNCMTYPS	Markertyp Slave		61	AXS	-	0
SYNCMWINM	Toleranzfenster für das Auftreten der Mastermarker		68	AXS	qc	0
SYNCMWINS	Toleranzfenster für das Auftreten der Slavemarker		69	AXS	qc	0
SYNCOFFTIME	Geschwindigkeitsausgleich eines Offsets		16	AXS	ms	0
SYNCPOSOFFS	Positionsoffset bei Positionssynchronisation		54	AXS	qc	0
SYNCREADY	Markeranzahl für Ready		56	AXS	-	1
SYNC_TTYPE	Typ der Synchronisation		51	AXS	-	0
SYNCVELREL	Tolerierte Abweichung des Folgeantriebs von der Master-Geschwindigkeit in %		66	AXS	%	0
SYNCVFTIME	Geschwindigkeitsfilter	709	65	AXS	t_filt (µsec)	0
TESTTIM	Messzeit im Zielfenster		24	AXI	ms	0
TESTVAL	Grenzwert für Messwert im Zielfenster		25	AXI	qc	1
TESTWIN	Zielfenster-Größe		8	AXI	qc	0
TIMER	Abtastzeit für PID-Regelung		14	AXR	msec	1
VELMAX	Maximalgeschwindigkeit		1	AXV	U/min	3000
VELMAXQC	Maximalgeschwindigkeit (interner Parameter, nicht veränderbar)		9	AXV	qc/ms* 1/65536	
VELRES	Geschwindigkeitsteiler		22	AXV		100

■ Alle SyncPos-Parameter im Detail

Die Reihenfolge der Parameter wird durch die interne Parameternummer bestimmt. Orientieren Sie sich am besten zuerst in der Übersicht; dann finden Sie die Detail-Informationen ganz schnell anhand der internen Parameternummer.

Allgemeines zu den Parameterwerten

Einige Grenzwerte sind auf Grund der besseren Lesbarkeit mit 1 Mrd. angegeben. Der exakte Wert beträgt jedoch 1.073.741.824.


ACHTUNG!

Die durch den Parameter **Benutzerfaktor** POSFACT_Z (23) und POSFACT_N (26) definierte Einheit BE wird nicht bei allen Weg-/Positions-Parametern verwendet. Für verschiedene Parameter zum Beispiel I_POSLIMITSW und **Tolerierter Positionsfehler** POSERR (15) gilt die Einheit Quadcount (qc).
4 Quadcounts entsprechen einer Gebereinheit (bei Inkrementalgebern).

Abkürzungen

In der folgenden Beschreibung steht für die Grenzwerte die Variable MLONG mit dem Wert:
MLONG = 1.073.741.824

Werkseinstellung

Mit ★ ist jeweils die Werkseinstellung gekennzeichnet.

Eingabebereiche

Die Überschreitung der angegebenen Eingabebereiche wird vom Programm nicht geprüft, da es wegen der großen Wertebereiche keine sinnvollen Kontrollmöglichkeiten gibt.


ACHTUNG!

Schon innerhalb der angegebenen Bereiche kann es durch die großen Leistungsunterschiede der Motoren und den vielseitigen Anwendungsmöglichkeiten zu unsinnigen Eingaben kommen. Es liegt daher in der Verantwortung des Programmierers und der Anwender, auf die zulässigen Leistungsbereiche der Antriebe und des Systems zu achten.

1 VELMAX

VELMAX definiert die Nenngeschwindigkeit des Antriebes. Der Wert wird in U/min angegeben und wird zur Berechnung von Rampen und Istgeschwindigkeiten benötigt.


ACHTUNG!

Die Nenngeschwindigkeit bezieht sich auf die Drehzahl des Encoders.

Inhalt

Maximalgeschwindigkeit

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Einheit

U/min

Bereich

1 ... 65535

★ 3000

2 ENCODER

Der Parameter **Geberstrichzahl** enthält die Auflösung des Istwertgebers (Inkremental- oder Absolutdrehgeber) bezogen auf eine Encoderumdrehung. Aus dieser Information wird die Anzahl der Quadcounts (qc) pro Umdrehung berechnet. Quadcount ist die für alle Wegmessungen zugrunde liegende Einheit. Quadcounts werden durch die Auswertung aller Flanken von der A- und B-Spur erzeugt. Ein Geberstrich eines Inkrementalgebers entspricht dabei vier Quadcounts. Bei Absolut-Drehgebern werden die absoluten Werte 1 : 1 zurückgeliefert.

Die Anzahl der Quadcounts pro Umdrehung wird für die Indexpulssuche bei der Referenzfahrt sowie für die Umrechnung von Geschwindigkeiten und Beschleunigungen in interne Einheiten benötigt. Die **Geberstrichzahl** liefert auch die Information, ob bei einer HOME- oder INDEX-Fahrt das Indexsignal verpasst wurde. Wenn mehr als eine volle Umdrehung ausgeführt wurde, ohne einen Indeximpuls zu registrieren, erfolgt eine entsprechende Fehlermeldung.

Inhalt

Auflösung des Istwertgebers (Geberstrichzahl)

Parametergruppe

Achsparemeter Encoder AXE
CAM-Editor: Registerkarte Encoder

Einheit

Geberstrichzahl/Umdrehung

Bereich

1 ... MLONG ★ 500



ACHTUNG!

Es sind keine negativen Werte erlaubt. Ob mit oder ohne Indexpuls gearbeitet wird, wird im Parameter HOMETYPE (40) angegeben.

Grenzwerte

Um eine einwandfreie Funktion der SyncPos-Option zu gewährleisten, darf das Produkt aus der **Geberstrichzahl** ENCODER (2) und **Maximalgeschwindigkeit** in Geberumdrehungen/s die Grenzfrequenz der Encoder-Eingangsstufe (220 kHz) nicht überschreiten:

$$\text{ENCODER} * \text{VELMAX} \left[\frac{\text{Geberumdrehung}}{\text{s}} \right] \leq 220 \text{ kHz}$$

3 HOME_FORCE

Wenn dieser Parameter auf Ja = 1 gesetzt ist, muss eine Homefahrt durchgeführt werden, bevor irgendeine andere Positionierfahrt durchgeführt werden kann.

Bei einem Fahrbefehl ohne erfolgreich durchgeführte Homefahrt wird der Fehler O.ERR 6 ausgelöst.

Inhalt

Homefahrt erzwingen?

Parametergruppe

Achsparemeter Homefahrt AXH
CAM-Editor: Registerkarte Homefahrt

Bereich

0; 1 ★0

0 = Homefahrt wird nicht erzwungen. Nach dem Einschalten gilt die aktuelle Position als Realnullpunkt

1 = Homefahrt wird erzwungen

Nach dem Einschalten des VLT sowie nach dem Ändern von Achsparemetern muss vor einem Fahrbefehl – ob direkt oder durch ein Programm ausgeführt – zwingend zuerst eine Homefahrt erfolgen.

Intern kann der Parameter auch den Wert 255 enthalten, der anzeigt, dass ein Anfahren der Home-Position erzwungen wurde und bereits erfolgt ist.



ACHTUNG!

Aus Sicherheitsgründen und zur Vermeidung von Fehlpositionierungen sollte der Parameter immer 1 gesetzt und damit eine Homefahrt erzwungen werden.

Es muss in diesem Fall jedoch berücksichtigt werden, dass alle Programme vor dem ersten Fahrbefehl einen HOME Befehl ausführen müssen, um eine einwandfreie Funktion zu erhalten.

4 NEGLIMIT

NEGLIMIT gibt die **negative Wegbegrenzung** für alle Fahrbewegungen an. Wird dieser Wert überschritten, wird ein Fehler ausgelöst. NEGLIMIT ist nur aktiv, wenn SWNEGLIMACT (19) gesetzt ist. Ein Positionierbefehl, der außerhalb der eingestellten Grenzen liegt, wird nicht ausgeführt.

Inhalt

Negativer Software-Endschalter

Parametergruppe

Achsparameter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Ein-/Ausgänge

Einheit

qc

Bereich

-MLONG ... MLONG ★ -500000



ACHTUNG!

Die Wegbegrenzung wird bei der Verwendung des Befehls DEF ORIGIN automatisch angepasst, so dass die ursprüngliche Position des Verfahrbereichs erhalten bleibt.

5 POSLIMIT

POSLIMIT gibt die **positive Wegbegrenzung** für alle Fahrbewegungen an. Wird dieser Wert überschritten, wird ein Fehler ausgelöst. POSLIMIT ist nur aktiv, wenn SWPOSLIMACT (20) gesetzt ist. Ein Positionierbefehl außerhalb der eingestellten Grenzen wird nicht ausgeführt.

Inhalt

Positiver Software-Endschalter

Parametergruppe

Achsparameter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Ein-/Ausgänge

Einheit

qc

Bereich

-MLONG ... MLONG ★ 500000



ACHTUNG!

Die Wegbegrenzung wird bei der Verwendung des Befehls DEF ORIGIN automatisch angepasst, so dass die ursprüngliche Position des Verfahrbereichs erhalten bleibt.

6 SYNCMMAXCORR

SYNCMMAXCORR wird benutzt, um die maximale Korrektur, die durch die Markerkorrektur vorgenommen wird, zu begrenzen. Der Wert wird in qc (Slave) eingegeben. Der Befehl arbeitet mit SYNCM und SYNCC.

Nachdem die PFG_G_KORREKTUR berechnet ist, wird PFG_G_KORREST auf das Minimum der SYNCMMAXCORR gesetzt. Dieser Wert wird dann für die Korrektur benutzt.



ACHTUNG!

Wenn man SYNCMFTIME (18) oder SYNCVFTIME (65) (negativ) gesetzt hat, wird diese Korrektur über eine gewisse Zeit gedehnt, abhängig von diesen Faktoren.

Inhalt

Begrenzt die maximale Korrektur, die durch die Markerkorrektur vorgenommen wird.

Parametergruppe

AXS Synchronisation;
(Noch nicht im Parameter-Dialogfenster realisiert.)

Einheit

qc

Bereich

0 ... MLONG ★ 0
0 = keine Begrenzung

7 HOME_VEL

HOME_VEL bestimmt die **Home-Geschwindigkeit**, mit der die Fahrt zum Referenzschalter durchgeführt wird. Die Geschwindigkeitsangabe ist bezogen auf die Nenngeschwindigkeit und abhängig von dem Parameter VELRES (22). Standardgemäß ist die Angabe in % von der Nenngeschwindigkeit.

$$\text{Home Geschwindigkeit} = \text{HOME_VEL} \cdot \frac{\text{VELMAX (1)}}{\text{VELRES (22)}}$$

Inhalt

Geschwindigkeit für Homefahrt

Parametergruppe

Achsparemeter Homefahrt AXH
CAM-Editor: Registerkarte Homefahrt

Einheit

% (VELRES = 100)

Bereich

-VELRES ... VELRES ★ 10
Mit einem negativen Vorzeichen wird in der anderen Richtung gesucht.


ACHTUNG!

Da immer in der gleichen Drehrichtung (abhängig vom Vorzeichen) nach dem Referenzschalter gesucht wird, sollte dieser an den Grenzen des Fahrbereichs angebracht werden. Nur so kann sichergestellt werden, dass sich der Antrieb bei einer Homefahrt aus allen Positionen auch tatsächlich in Richtung des Referenzschalters und nicht von ihm weg bewegt.

Um eine gute Repetierbarkeit der Referenzfahrt zu erhalten, sollte mit höchstens 10 % der maximalen Drehzahl verfahren werden.

8 TESTWIN

TESTWIN gibt die Größe des Zielfensters an. Eine Position gilt erst dann als erreicht, wenn die Soll-Fahrt (Trapez) abgearbeitet ist, die aktuelle Position innerhalb des Fensters liegt und die Geschwindigkeit kleiner ist als TESTVAL ist. (Voraussetzung: TESTWIN und TESTTIM sind aktiviert.)

Hierbei ist die Geschwindigkeit TESTVAL in qc/TESTTIM angegeben.

Der Controller wartet mit dem Ausführen des jeweils nächsten Befehls, bis die Istposition innerhalb des Zielfensters liegt.

Wenn TESTWIN nicht aktiv ist (Parameter 0), gilt das Ziel als erreicht, wenn die Sollposition gleich der Zielposition ist. Diese muss jedoch nicht mit der tatsächlichen Position des Antriebs übereinstimmen.

Inhalt

Zielfenster-Größe

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Ein-/Ausgänge

Einheit

qc

Bereich

0 ... MLONG ★ 0
0 = inaktiv
TESTWIN muss immer kleiner als TESTVAL sein.


ACHTUNG!

Wird das Zielfenster um die Endposition zu klein gewählt, könnte sich der Antrieb in einer sehr kleinen Umgebung um die Endposition bewegen, ohne das Zielfenster zu erreichen, so dass das Programm bei dem entsprechenden Positionierbefehl „hängen“ bleibt.

Ein Zielfenster von 0 deaktiviert die Überwachung der aktuellen Position und überwacht lediglich die Kommandoposition.

9 VELMAXQC

VELMAXQC bestimmt die **Maximalgeschwindigkeit**. Auf diese Geschwindigkeit beziehen sich alle Angaben, die mit Hilfe von VELRES (22) gemacht werden. Die Sampletime beträgt 1 ms. VELMAXQC ist ein interner Parameter, der nicht verändert werden kann. Er wird automatisch aus VELMAX (1) berechnet, die der Benutzer in U/min eingibt. Diese internen Werte sind entscheidend für den zulässigen Wertebereich. In der Praxis spielen diese Grenzwerte jedoch keine Rolle, da sie weit über der, durch die Hardware bestimmte Encoder-Eingangsfrequenz liegen.

Inhalt

Maximalgeschwindigkeit

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Einheit

qc/ms * 1/65536

Bereich

1 ... MLONG

10 ACCMAXQC

ACCMAXQC legt die maximale Beschleunigung fest. Dies ist die Zeitspanne, die der Antrieb bei angeschlossener Last benötigt, um die Maximaldrehzahl zu erreichen. Auf diese Beschleunigung beziehen sich alle anderen Angaben, die über die Normierung VELRES (22) gemacht werden. Errechnet wird dieser interne Parameter – er kann nicht verändert werden – automatisch aus RAMPMIN (31), die der Benutzer in ms eingibt. Diese internen Werte sind entscheidend für den zulässigen Wertebereich. In der Praxis spielen diese Grenzwerte jedoch keine Rolle.

Inhalt

Maximale Beschleunigung

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Einheit

qc/(ms)² * 1/65536

Bereich

1 ... MLONG

Der Parameter wird direkt in ms eingegeben, aber automatisch in interne Werte umgerechnet. Diese internen Werte sind entscheidend für den zulässigen Wertebereich. In der Praxis spielen diese Grenzwerte jedoch keine Rolle.


ACHTUNG!

Wird eine zu niedrige Anlaufzeit eingegeben, die eine unter den gegebenen mechanischen Bedingungen nicht erreichbare Beschleunigung bedingt, tritt im Betrieb typischerweise ein Schleppfehler auf.

11 KPROP (702)

Der **Proportionalfaktor** KPROP gibt den linearen Korrekturfaktor an, mit dem die Abweichung zwischen der aktuellen Soll- und Istposition bewertet und eine entsprechende Korrektur des Motordrehmoments vorgenommen wird

Faustregel:

KPROP größer = Antrieb wird „steifer“

KPROP zu hoch = Neigung zum starken Überschwingen

Inhalt

Proportionalwert für PID-Regelung

Parametergruppe

Achsparemeter Regelung AXR

CAM-Editor: Registerkarte Regelung

Bereich

1 ... 65000 ★ 30

12 KDER (703)

Der **Differentialfaktor** KDER ist der Korrekturfaktor, mit dem die Änderungsgeschwindigkeit eines Motorpositionsfehlers bewertet wird.

Der Differentialanteil wirkt der, durch einen hohen Proportionalanteil verursachten Überschwingneigung entgegen und „dämpft“ das System. Ein zu groß gewählter Differentialanteil führt jedoch zu einem „nervösen“ Antrieb.

Inhalt

Differentialwert für PID-Regelung

Parametergruppe

Achsparemeter Regelung AXR

CAM-Editor: Registerkarte Regelung

Bereich

0 ... 65000 ★ 0

13 KINT (704)

Der **Integralfaktor** KINT ist der Gewichtungsfaktor, mit dem im Zeitpunkt n die Summe aller Motorpositionsfehler bewertet wird.

Der Integralanteil des PID-Filters bewirkt ein entsprechend zeitlich anwachsendes, korrigierendes Motordrehmoment. Durch den Integralanteil wird ein statischer Positionsfehler zu Null ausgeregelt, auch wenn eine konstante Last am Motor anliegt.

Ein zu großer Integralanteil führt jedoch zu einem "nervösen" Antrieb.

Inhalt

Integralwert für PID-Regelung

Parametergruppe

Achsparemeter Regelung AXR

CAM-Editor: Registerkarte Regelung

Bereich

0 ... 65000 ★ 0

14 TIMER

Der Parameter TIMER bestimmt die Abtastzeit des Regelalgorithmus.

Erhöhen Sie den Wert der Werkseinstellung zum

Beispiel

- bei sehr kleinen Pulsfrequenzen wie 1 bis 2 qc per Abtastzeit;
(Sie brauchen mindestens 10 bis 20 qc per Abtastzeit.)
- oder bei sehr trägen Systemen mit einer großen Totzeit. Würde man hier mit 1 ms regeln, würden große Motoren schwingen.


ACHTUNG!

Beachten Sie, dass der Parameter einen direkten Einfluss auf die PID-Schleife hat, wenn Sie zum Beispiel den TIMER verdoppeln, wirkt KPROP (11) doppelt so stark.

Inhalt

Abtastzeit für PID-Regelung

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

ms

Bereich

1 ... MLONG ★ 1
Sinnvollerweise setzt man den Wert nicht höher als 1000 (= 1 s). Das wäre bereits eine sehr träge Regelung.

15 POSERR

Der *maximal Tolerierte Positionsfehler* POSERR definiert die erlaubte Toleranz zwischen der aktuellen Istposition und der errechneten Kommando-position. Wenn der mit POSERR definierte Wert überschritten wird, wird die Lageregelung abgeschaltet und ein Schleppfehler ausgelöst.

Inhalt

Maximal tolerierter Positionsfehler (Schleppabstand)

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

qc

Bereich

1 ... MLONG ★ 20000

Der Schleppabstand hat keinen Einfluss auf die Positioniergenauigkeit, sondern bestimmt lediglich, wie exakt der theoretisch errechnete Verfahrensweg eingehalten werden muss, ohne dass ein Fehler ausgelöst wird.


ACHTUNG!

Der Schleppabstand darf aus Sicherheitsgründen nicht zu groß gewählt werden, um Mensch und Maschine nicht zu gefährden. Zu kleine Werte für den Schleppabstand können hingegen häufige Fehlermeldungen zur Folge haben. Als Richtwert kann die vierfache Strichzahl des Drehgebers angesetzt werden, was wiederum einer Drehgeberumdrehung entspricht.

16 SYNCOFFTIME
Inhalt

Ausgleichsgeschwindigkeit eines Offsets
(1. Einsynchronisieren; 2. neuer Offset)

Beschreibung

Der Offset-Filter SYNCOFFTIME beeinflusst auch die Art, wie ein neuer SYNCPOSOFFS Wert gehandhabt wird. Der Offset, der ausgeführt werden muss, wird Schritt für Schritt realisiert. Eine Schrittweite, die pro Abtastperiode (ms) auszuführen ist, wird wie folgt berechnet:

$$\text{Schrittweite} = \text{SYNCPULSM} / \text{SYNCOFFTIME}$$

(Integer Anteil).

Daher wird es also SYNCOFFTIME dauern, um einen Offset von SYNCPULSM auszuführen. SYNCOFFTIME beeinflusst auch die Marker-Startkorrektur und die Korrektur der Markerfehler (siehe SYNCMFTIME).

Kompatibilität

Standardbefehl ab Optionskarte, Version 5.04 aufwärts.

Parametergruppe

AXS Synchronisation
(Ist noch nicht in der Benutzeroberfläche realisiert.)

Einheit

1 ms

Bereich

0 ... MLONG ★ 0

17 SYNCMFPAR
Inhalt

Konfiguration für den Markerfilter

Beschreibung

Dieser Parameter SYNCMFPAR wird benutzt, um das Verhalten des Markerfilters zu beeinflussen, siehe SYNCMFTIME.

Kompatibilität

Standardbefehl ab Optionskarte, Version 5.04 aufwärts.

Parametergruppe

AXS Synchronisation
(Ist noch nicht in der Benutzeroberfläche realisiert.)

Bereich

0 or 1 or 2 or 4 or 16 ★ 0
 0 = Jedes Mal, wenn ein echter Master-Marker gefunden wird, wird die Markerfilter-Konstante wie folgt berechnet:
 Gefilterte alte Master Geschwindigkeit *
 SYNCMFTIME / (SYNCPULSM * 3)
 1 = Berechnet die Markerfilter-Konstante als SYNCMFTIME/300
 = Getriebekorrektur ist durchgeführt
 4 = Statt SYNCOFFTIME wird die Korrekturzeit benutzt.
 16 = Es wird keine Korrektur bezüglich des Fehlers der Markerabstände gemacht.
 Für weitere Beschreibungen der Auswahlmöglichkeiten siehe SYNCMFTIME

18 SYNCMFTIME
Inhalt

Definiert die Filterzeit für die Markerkorrektur

Anwendungsbeispiel

Die Zeitungsproduktion benötigt dieses Art des Filters um eine Förderkette mit den Zeitungen, die aus der Druckmaschine kommen, zu synchronisieren. Weil der Zeitungsfluss nicht ganz konstant ist, wären die Bewegungen der Kette sehr hart und dynamisch, falls man ohne Filter synchronisieren würde. Mit allen anderen Arten des Filterns würde das System beginnen in sinusförmigen Wellen zu schwingen. Benutzt man aber diese komplexe Filtermethode, dann arbeitet die Synchronisation sehr gut und löst das Problem

Beschreibung

SYNCMFTIME wird in ms eingegeben und wie folgt benutzt:


ACHTUNG!

Der Master-Geschwindigkeitsfilter SYNCVFTIME (65) wird zur besseren Auflösung in 1/1000 ms eingegeben, der Markerfilter (SYNCMFTIME) dagegen in Einheiten von 1ms.

Beispiel:

```
SET SYNCVFTIME -50000
```

```
SET SYNCMFTIME 2000
```

Das heißt, dass die Master-Geschwindigkeit über eine Periode von 50 ms gemittelt wird. Ein Markerfehler wird also innerhalb von 2000 ms korrigiert.

Der aktuelle gefilterte Markerabstand kann mit SYSVAR 4238 Index ausgelesen werden, wenn dieser Filter durch das Setzen von SYNCMFTIME aktiviert wurde. Um diesen Filterwert zu erhalten, wird intern berechnet, wie viele Marker passieren, wenn man mit der maximal erlaubten Geschwindigkeit über eine Periode von SYNCMFTIME fährt.

Die Parameter SYNCMFTIME, SYNCOFFTIME (16) und SYNCMFPAR (17) werden benutzt, um das Verhalten des Markerfilters zu beeinflussen (s.u.).

Kompatibilität

Wenn SYNCMFTIME = 0, verhält sich das System genau so wie bis zur Optionskarte Version 5.04, das heißt die Filterzeit für die Markerkorrektur hängt vom Wert des Parameters SYNCVFTIME (65) ab.

Das Filtern wird wie folgt gehandhabt:

Markerfilter Berechnung

nur wenn SYNCMFTIME > 0

Wenn SYNCMFPAR = 1, kann jedes Mal, wenn ein echter Master-Marker erkannt wird, die Markerfilter-Konstante als SYNCMFTIME/300 berechnet werden.

Wenn SYNCMFPAR = 0, kann jedes Mal, wenn ein echter Master-Marker erkannt wird, die Markerfilter-Konstante wie folgt berechnet werden: Gefilterte alte Master Geschwindigkeit * SYNCMFTIME / (SYNCPULS * 3); das heißt, dass der konstante Markerfilter als Zeitkonstante zum Filtern benutzt wird. Dann sollte die Zeit, die benötigt wird, um eine Reaktion entsprechend eines konstanten Eingangswertes zu erhalten, nahezu SYNCMFTIME sein.

Die Berechnung ist notwendig, weil der Filter bei jedem Marker ausgeführt wird und nicht jede ms.

Dieser Markerfilter-Konstante wird nun benutzt, um den Markerabstand zu filtern. Das Ergebnis wiederum wird benutzt, um die notwendige Getriebekorrektur wie folgt zu berechnen:

Getriebekorrektur = (SYNCPULSM - gefilterten Markerabstand) / gefilterten Markerabstand

Filter Master-Geschwindigkeit und Getriebekorrektur

Pro Abtastperiode wird die gefilterte Master-Geschwindigkeit neu berechnet (Differenz der aktuellen und der letzten Master-Position).

Wenn SYNCVFTIME < 0 wird die gefilterte alte Master-Geschwindigkeit mit einer konstanten Filterzeit = SYNCVFTIME / 1000 berechnet.

Dan wird die gefilterte alte Master-Geschwindigkeit gleich der aktuellen Master-Geschwindigkeit gesetzt.

Wenn SYNCMFTIME > 0 und SYNCMFPAR = 2 wird die Getriebekorrektur durchgeführt, indem man zum aktuellen Getriebeverhältnis die mit dem Getriebeverhältnis multiplizierte Master-Geschwindigkeit addiert.

Startkorrektur

nur wenn SYNCMFTIME > 0

Die Startkorrektur ist jene Korrektur, die ausgeführt werden muss, sobald die Startbedingung erfüllt ist. Das heißt, entweder mussten die ersten zwei Marker beobachtet werden (SYNCMSTART 1,6) oder es musste die Master-Geschwindigkeit erreicht und zusätzlich die ersten zwei Marker beobachtet werden (SYNCMSTART 2,3,4,5). Diese Startkorrektur wird so aufgeteilt, dass sie nach SYNCOFFTIME erledigt sein wird. (Tatsächlich wird sie durch die Anzahl der

Marker geteilt, die in SYNCOFFTIME mit der aktuellen Master-Geschwindigkeit passiert wurden und der erhaltene Wert wird zur normalen Markerkorrektur addiert).

Wenn SYNCOFFTIME = 0, wird die Startkorrektur sofort erledigt, das bedeutet, dass die Korrektur innerhalb von zwei Markern durchgeführt wird.

Marker Korrektur
 SYNCMFTIME > 0

Ab SyncPos-PC-Software Version 5.04:
 Zuerst wird vom Markerfehler die verbleibende Startkorrektur abgezogen. Dann wird Korrekturfilterzeit gesetzt, und zwar entsprechend dem Parameter SYNCOFFTIME (Die Master-Geschwindigkeit hängt von der Markeranzahl ab; siehe Startkorrektur).

Nun wird die Summe aller Markerabstandsfehler in einen Markerfilter geleitet, um die gefilterte Summe zu berechnen. Danach wird die gefilterte Summe der Fehler von der ungefilterten abgezogen. Dieses Ergebnis wird schließlich benutzt, um die Markerkorrektur wiederum zu korrigieren.

Diese bereinigte Korrektur wird in den Korrekturfilter eingegeben. Das Ergebnis dieses Korrekturfilters wird gespeichert (plus den Anteil der Startkorrektur, falls notwendig).

Dann wird versucht, diese Markerkorrektur über einen Markerabstand zu spreizen. Das wird durch Teilen der Korrektur durch die Anzahl der Samples erreicht, die notwendig ist, um einen Markerabstand mit der aktuellen Master-Geschwindigkeit zu passieren. Der Wert wird gespeichert und bei jeder Abtastperiode benutzt, um die berechnete Slave-Position zu korrigieren.

Folgende SYNCMFPAR Werte verändern das Verhalten:

SYNCMFPAR & 4 → statt SYNCOFFTIME wird die Korrekturzeit benutzt.

SYNCMFPAR & 16 → Es wird keine Korrektur durchgeführt, die den Markerabstand betrifft.

Markerkorrektur
 SYNCMFTIME = 0

Im ersten Fall – wenn Markerkorrektur > 0 – wird die Korrektur über ein Zeit von (-SYNCVFTIME /100) ms gespreizt.

Im zweiten Fall wird die Korrektur zur Sollposition sofort addiert.

In jedem Fall wird die Reaktion natürlich durch die aktuelle Beschleunigung und Verzögerung begrenzt.

Parametergruppe
 AXS Synchronisation
 (Noch nicht im Parameter-Dialogfenster realisiert.)

Einheit

1 ms

Bereich

-MLONG ... MLONG ★ 0
 0 = Wenn SYNCVFTIME (65) negativ ist, wird die Markerkorrektur durch SYNCVFTIME / 100 gespreizt.

19 SWNEGLIMACT

Durch Setzen dieses Parameters (auf 1) wird dem VLT mitgeteilt, dass der negative Software-Endschalter überwacht werden soll. Dann wird bei jeder Bewegung überprüft, ob die Zielposition außerhalb des zulässigen Verfahrbereichs liegt: In diesem Fall wird eine Fehlermeldung ausgelöst und die Antriebsregelung abgeschaltet.

Im Positioniermodus bedeutet dies, dass der entsprechende Positioniervorgang nicht gestartet wird und der Fehler durch einen ERRCLR Befehl behoben werden kann.

Im Drehzahlmodus kann der Fehler erst beim Überfahren der Wegbegrenzung erkannt werden, wodurch sich der Antrieb beim Auftreten der Fehlermeldung bereits außerhalb des zulässigen Verfahrbereichs befindet. In diesem Fall ist es notwendig den Antrieb von Hand wieder in den zulässigen Bereich zurück zu bewegen und den Fehler zu löschen oder im Menü "STEUERUNG" → "PARAMETER" → "ACHSEN" vorübergehend den entsprechenden **Software-Endschalter** abzuschalten und den Fehler zu löschen.

Inhalt

Negativer Software-Endschalter aktiv

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
 CAM-Editor: Registerkarte Ein-/Ausgänge

Bereich

0; 1 ★ 0
 0 = inaktiv
 1 = aktiv

20 SWPOSLIMACT

Durch Setzen dieses Parameters (auf 1) wird dem VLT mitgeteilt, dass der positive Software-Endschalter überwacht werden soll. Bei jeder Bewegung wird in dann überprüft, ob die Zielposition außerhalb des zulässigen Verfahrbereiches liegt und im gegebenen Fall eine Fehlermeldung ausgelöst und die Antriebsregelung abgeschaltet.

Im Positioniermodus bedeutet dies, dass der entsprechende Positioniervorgang nicht gestartet wird und der Fehler durch einen ERRCLR Befehl behoben werden kann.

Im Drehzahlmodus kann der Fehler erst beim Überfahren der Begrenzung erkannt werden, wodurch sich der Antrieb beim Auftreten der Fehlermeldung bereits außerhalb des zulässigen Verfahrbereiches befindet. Dann müssen Sie den Antrieb von Hand wieder in den zulässigen Bereich zurück bewegen und den Fehler löschen oder im Menü "STEUERUNG" → "PARAMETER" → "ACHSEN" vorübergehend den entsprechenden **Software-Endschalter** abschalten und den Fehler löschen.

Inhalt

Positiver Software-Endschalter aktiv

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
 CAM-Editor: Registerkarte Ein-/Ausgänge

Bereich

0; 1 ★ 0
 0 = inaktiv
 1 = aktiv

21 KILIM (705)

Das **Integrationslimit** KILIM gibt den maximalen Wert an, der in das PID-Filter über den Integralanteil eingehen kann. So lässt sich verhindern, dass durch einen hohen aufaddierten Fehler eine zu starke Gegenregelung erfolgt, die das System zum Schwingen bringen würde.

Inhalt

Grenzwert für Integralsumme für PID-Regelung

Parametergruppe

Achsparemeter Regelung AXR
 CAM-Editor: Registerkarte Regelung

Bereich

0 ... 65000 ★ 0
 0 = AUS

22 VELRES

Der **Geschwindigkeitsteiler** VELRES definiert eine Bezugsgröße für die Geschwindigkeitswerte der Fahrbefehle und Parameter.

Die Angabe der Geschwindigkeit und Beschleunigung kann dann in ganzen Zahlen, bezogen auf diese Normierung, erfolgen. Der Wert 100 bedeutet, dass sich die Angaben in den Befehlen auf 100 beziehen, also in Prozent.

Inhalt

Geschwindigkeitsteiler

Parametergruppe

Achsparemeter Geschwindigkeit AXV
 CAM-Editor: Registerkarte Geschwindigkeit

Bereich

1 ... MLONG ★ 100

23 POSFACT_Z

Wegangaben in Fahrbefehlen erfolgen in Benutzereinheiten (BE) und werden intern in Quadcounts umgerechnet. So ist es durch eine entsprechende Wahl dieser Normierungsgröße möglich, mit beliebigen technischen Maßangaben (zum Beispiel in mm oder Grad) zu arbeiten.

Der Faktor ist ein Bruch, der sich aus Zähler und Nenner zusammensetzt:

$$\text{POSFACT_Z (23)} / \text{POSFACT_N (26)} = 1 \text{ BE}$$

Die Normierung bestimmt, wie viele Quadcounts eine Benutzereinheit ergeben: wenn der Faktor zum Beispiel 50375/1000 beträgt, entspricht eine BE genau 50,375 qc.

Im CAM-Mode wird der Parameter benutzt, um die Einheit für den Slave-Antrieb festzulegen, damit man auch im CAM-Editor mit sinnvollen Einheiten arbeiten kann. Siehe Beispiel 2.

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ BE}$$

vorausgesetzt dass:

$$\text{Getriebefaktor} = \frac{\text{Motorumdrehungen}}{\text{Umdrehungen am Abtrieb}}$$

Encoder = Inkremental-Encoder (bei absoluten Encodern entfällt der Multiplikator 4)

Skalierfaktor = Anzahl der Benutzereinheiten BE (qc), die einer Umdrehung am Antrieb entsprechen

Außerdem kann man die Kurven mit diesem Faktor stauchen oder strecken, ohne jeweils neue Kurven definieren zu müssen. Die Verwendung von Zähler und Nenner für den Getriebefaktor führt zu einem sehr präzisen Ergebnis, da in fast allen Fällen Übersetzungen als Bruch darstellbar sind.

Inhalt

Zähler Benutzerfaktor, bzw. im CAM-Mode
Umrechnung der Einheiten qc in BE

Parametergruppe

Achsparameter Encoder AXE
CAM-Editor: Registerkarte Encoder

Bereich

1 ... MLONG /max. Position (BE) ★ 1000

Der obere Grenzwert hängt vom der maximalen Zielposition ab:

$$\text{max. Position (BE)} * \text{POSFACT_Z} < \text{MLONG}$$

hier gilt:

Beispiel: POSA max. Position (BE)

Beispiel 1: Welle oder Spindel

25 Motorumdrehungen ergeben 1 Spindelumdrehung; Getriebefaktor = 25/1

Encoder-Auflösung (Inkrementaldrehgeber) = 500

Spindelsteigung = 1 Umdrehung der Spindel = 5 mm

Skalierfaktor, wenn mit 1/10 mm Auflösung gearbeitet werden soll = 5 * 10 = 50

$$\frac{25/1 * 500 * 4}{50} \text{ qc} = \frac{25 * 10 * 4}{1} \text{ qc} = \frac{1000}{1} \text{ qc} = 1 \text{ BE}$$

Benutzerfaktor Zähler [23] = 1000

Benutzerfaktor Nenner [26] = 1

Beispiel 2: Walze

Getriebefaktor = 5/1

Encoder-Auflösung (Inkrementaldrehgeber) = 500

Ein Umdrehung der Walze ist 360 Grad. Wir wollen mit einer Auflösung von 1/10 Grad arbeiten, daher wird eine Umdrehung Walze in 3600 Arbeitseinheiten eingeteilt:

Skalierfaktor = 3600

$$\frac{5/100 * 500 * 4}{3600} \text{ qc} = 1 \text{ BE}$$

$$\frac{5 * 500 * 4}{3600} \text{ qc} = \frac{25}{9} \text{ qc} = 1 \text{ BE} = \frac{\text{POSFACT_Z (23)}}{\text{POSFACT_N (26)}}$$

Benutzerfaktor Zähler [23] = 25

Benutzerfaktor Nenner [26] = 9

24 TESTTIM

Nach dem Erreichen des Zielfensters wird zweimal die aktuelle Position gemessen und mit dem Parameter TESTVAL (25) verglichen. Wenn das Ergebnis kleiner ist als TESTVAL, gilt die Position als erreicht, andernfalls wird erneut gemessen. TESTTIM gibt den Zeitabstand zwischen diesen beiden Messungen an.

Inhalt

Messzeit im Zielfenster

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Ein-/Ausgänge

Einheit

ms

Bereich

0 ... 10 ★ 0


ACHTUNG!

Die Einschränkung auf 10 ms ist dadurch begründet, dass die Funktion *diffval* wirklich wartet und so lange auch keine Endschalter- und Schleppfehler-Überwachung aktiv sind. Deswegen sollte dieser Wert nicht zu lange sein.

25 TESTVAL

Nachdem das Zielfenster erreicht wurde, wird mit Abstand TESTTIM (24) zweimal die Position ausgelesen und der Abstand mit diesem **Zielfenster-Grenzwert** TESTVAL verglichen. Das Ergebnis entscheidet darüber ob die Position als erreicht gilt oder nicht.


ACHTUNG!

Bei langen Zeitangaben muss berücksichtigt werden, dass das Erreichen der Zielposition auf jeden Fall um diese Zeit verzögert wird.

Inhalt

Grenzwert für Messwert im Zielfenster

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Ein-/Ausgänge

Einheit

qc

Bereich

1 ... 65535 ★ 1
TESTVAL muss immer größer gesetzt werden, als TESTWIN.

26 POSFACT_N

Wegangaben in Fahrbefehlen erfolgen in Benutzereinheiten (BE) und werden intern in Quadcounts umgerechnet. So ist es durch eine entsprechende Wahl dieser Normierungsgröße möglich, mit beliebigen technischen Maßangaben (zum Beispiel mm) zu arbeiten.

Der Faktor ist ein Bruch, der sich aus Zähler und Nenner zusammensetzt:

$$1 \text{ BE} = \text{POSFACT_Z (23)} / \text{POSFACT_N (26)}$$

Die Normierung bestimmt, wie viele Quadcounts eine Benutzereinheit ergeben: wenn der Faktor zum Beispiel 50375/1000 beträgt, entspricht eine BE genau 50,375 qc.

Im CAM-Mode wird der Parameter benutzt, um die Einheit für den Slave-Antrieb festzulegen, damit man im Kurveneditor mit sinnvollen Einheiten arbeiten kann. Siehe Voraussetzung der Formel und Beispiel bei POSFACT_Z (23).

$$\text{Getriebefaktor} = \frac{\text{Motorumdrehungen}}{\text{Umdrehungen am Abtrieb}}$$

Außerdem kann man die Kurven mit diesem Faktor stauchen oder strecken, ohne jeweils neue Kurven definieren zu müssen.

Die Verwendung von Zähler und Nenner für den Getriebefaktor führt zu einem sehr präzisen Ergebnis, da in fast allen Fällen die Übersetzungen als Bruch darstellbar sind.

Inhalt

Nenner Benutzerfaktor
bzw. im CAM-Mode Umrechnung der Einheiten qc in BE

Parametergruppe

Achsparemeter Encoder AXE
CAM-Editor: Registerkarte Encoder

Bereich

1 ... MLONG ★ 1000

Beispiel

Siehe POSFACT_Z (23)

27 ENCODERTYPE

Der Parameter legt den Typ des Encoders für den Slave fest.

Bei der Verwendung eines linearen Absolut-Drehgebers können Fehler auftreten. Mit den Encodertypen „3“ und „4“ kann ein möglicher Sprung in den Positionsdaten erkannt werden, sofern er größer als die Encoder-Auflösung/2 ist. Die Korrektur erfolgt mittels eines künstlichen Positionswertes, der sich aus der letzten Geschwindigkeit errechnet. Sollte die Störung länger als 100 Auslesungen (> 100 ms) anliegen, wird nicht mehr korrigiert, was dann tatsächlich zu einem Schleppfehler führt.

Die gesamte Anzahl aufgetretener Störungen wird in einer internen Variablen gespeichert, die über SYSVAR[16] ausgelesen werden kann.

Inhalt

Encodertyp für Slave-Encoder (0)

Parametergruppe

Achsparemeter Encoder AXE
CAM-Editor: Registerkarte Encoder

Bereich

0 ... 4 und 100 ... 104 ★ 0
 0 = inkrementaler Encoder
 1 = Absoluter Encoder, Standard ca. 262 kHz
 2 = Absoluter Encoder, ca. 105 kHz
 3 = absoluter Encoder ohne Überlauf (linear), jedoch mit Fehlerkorrektur, ca. 262 kHz
 4 = absoluter Encoder ohne Überlauf (linear), jedoch mit Fehlerkorrektur ca. 105 kHz
 100 ... 104 = wie 0 ... 4, jedoch wird dann die Hardware-Überwachung des Encoders eingeschaltet. Wenn die Encoder-Leitungen unterbrochen sind oder bei Kurzschluss, wird der Fehler 92 ausgegeben.


ACHTUNG!

Mit absoluten Drehgebern können folgende Befehle nicht benutzt werden: DEF ORIGIN, HOME, INDEX und WAITNDX.


ACHTUNG!

Mit absoluten Drehgebern können folgende Befehle nur dann benutzt werden, wenn externe Marker eingesetzt werden: IPOS, MIPOS

28 POSDRCT

Normalerweise ist es für die Lageregelung erforderlich dass ein positiver Sollwert auch eine positive Änderung der Position bewirkt. Falls dies nicht der Fall ist (Drehgeber vertauscht oder Motor falsch angeschossen o.ä.) kann durch den Parameter POSDRCT = 2 der Sollwert intern umgedreht werden.

Bei manchen Anwendungen ist es wünschenswert, die Drehrichtung in Benutzereinheiten umzudrehen. Dies wird durch das negative Vorzeichen bei POSDRCT erreicht.

Die Richtung der Synchronisation (Verhältnis zum Master) kann durch negativen SYNCFAC TM umgedreht werden.

Bei einer Synchronisation im CAM-Mode können Sie mit POSDRCT für den Slave eine positive Drehrichtung bestimmen, die Voraussetzung für die CAM-Funktionalität ist.

Inhalt

Positive Drehrichtung

Parametergruppe

Achsp parameter Encoder AXE
CAM-Editor: Registerkarte Encoder

Bereich

-2 ... 2 ★ 1

- 1 = Keine Veränderung, d.h. positive Sollwerte ergeben positive Drehgeberwerte.
- 1 = Das Vorzeichen der Benutzereinheit wird gedreht. Positive Sollwerte ergeben demnach positive Drehgeberwerte, die aber negativ angezeigt werden. Dies gilt für alle Ausgaben (APOS, CPOS, ...), alle Benutzereingaben (POSA, POSR, ...) und alle Synchronisationsfaktoren sowie die Geschwindigkeiten (CVEL, HOME_VEL).
- 2 = Das Vorzeichen des Sollwertes wird intern getauscht (Plus wird Minus und umgekehrt). Dies kommt einem Umdrehen der Motorleitungen gleich, bzw. dem Vertauschen der A- und B-Spur beim Drehgeber.
- 2 = Wie „2“, also das Vorzeichen des Sollwertes wird intern getauscht und zusätzlich wird wie bei „-1“ das Vorzeichen der Benutzereinheit negiert.

29 PROFTIME

Der Parameter bestimmt die Abtastzeit für den Profilergenerator, der unabhängig von der Abtastzeit der PID-Regelung (TIMER) ist.

Bei anspruchsvollen Regelaufgaben im Hintergrund (SYNCP, SYNCM, SYNCC) kann die Ausführungszeit des SyncPos-Programmes drastisch ansteigen. In solchen Fällen kann die Abtastzeit des Profilergenerators auf 2000 erhöht werden, um mehr Zeit für das SyncPos-Programm zur Verfügung zu haben. Höhere Werte als 2000 sind kaum von Vorteil.



ACHTUNG!

Die VEL, ACC und DEC Befehle müssen nach einem SET PROFTIME Befehl gesetzt werden.

Inhalt

Abtastzeit für Profilergenerator

Parametergruppe

Achsp parameter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

ms

Bereich

1000, 2000 ★ 1000
1000 = 1 ms
2000 = 2 ms

30 MENCODER

MENCODER gibt die Strichzahl des Master-Encoders in Strichen pro Umdrehung an.

Inhalt

Auflösung des Master-Encoders

Parametergruppe

Achsp parameter Encoder AXE
CAM-Editor: Registerkarte Encoder

Einheit

Geberstriche/Umdrehung

Bereich

1 ... MLONG ★ 500

31 RAMPMIN

Der Parameter RAMPMIN legt die kürzeste Rampe (*maximale Beschleunigung*) fest. Er gibt an wie lange die Beschleunigungsphase minimal dauert, um die Nenngeschwindigkeit zu erreichen.


ACHTUNG!

Wenn man mit der SyncPos-Optionskarte arbeitet, sollte man die Rampen immer über die Optionskarte einstellen und nicht im VLT. Die VLT-Rampen müssen dabei immer auf Minimum stehen.

Inhalt

Kürzeste Rampe

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Einheit

ms

Bereich

1 ... 65535 ★ 1000

32 RAMPTYPE

RAMPTYPE legt die Rampenform (Trapez oder sinusförmig) fest.

Inhalt

Rampenform

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Bereich

0; 1 ★ 0
0 = Trapez
1 = sinusförmig

33 DFLTVEL

DFLTVEL gibt die Default-Geschwindigkeit an, die immer dann verwendet wird, wenn keine Geschwindigkeit im Verfahrssatz definiert wurde. Der Wert bezieht sich auf den **Geschwindigkeitsteiler** VELRES (22).

Inhalt

Default-Geschwindigkeit

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Einheit

$1/V_{VELRES}$;
Standard = $1/100 = \%$

Bereich

1 ... VELRES ★ 50

34 DFLTACC

DFLTACC gibt die Beschleunigung an, die verwendet wird, wenn keine explizite Angabe vorliegt. Die Angabe erfolgt im Verhältnis zu RAMPMIN (31) bezogen auf den Parameter VELRES (22).

Inhalt

Default-Beschleunigung

Parametergruppe

Achsparemeter Geschwindigkeit AXV
CAM-Editor: Registerkarte Geschwindigkeit

Einheit

$1/V_{VELRES}$
Standard = $1/100 = \%$

Bereich

1 ... VELRES ★ 50

35 BANDWIDTH (706)

Mit dieser Funktion können Sie die Bandbreite, in der der PID-Regelalgorithmus wirken soll, begrenzen, um zum Beispiel bei schwingungsgefährdeten Systemen das Aufschaukeln der Schwingungen zu vermeiden.

Wenn Sie mit BANDWIDTH arbeiten, müssen Sie für die Parameter FFVEL (36) und FFACC (37) wesentlich höhere Werte eintragen, um eine entsprechende Regelung zu erreichen. Ein so eingestelltes System ist zwar nicht mehr so dynamisch, aber dafür wesentlich stabiler und neigt weniger zu unkontrollierten Schwingungen.

Inhalt

Bandbreite innerhalb der der PID-Filter aktiv ist.

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

‰

Der Wert 1000 bedeutet, dass der PID-Filter den vollen Sollwert ausgeben kann. Bei einem BANDWIDTH von 500 werden nur 50 % des Sollwertes ausgeben. Kleinere Werte als 1000 begrenzen also den P-Anteil entsprechend.

Bereich

0 ... 1000 ★ 1000

36 FFVEL (707)

Wenn eine Regelung in der Bandbreite begrenzt ist, muss eine Grundgeschwindigkeit vorgegeben werden, damit ausgeschlossen wird, dass die Regelung durch die eingestellte Begrenzung das Fahren des Antriebs gänzlich verhindert. FFVEL gibt den Wert an, mit dem Geschwindigkeits-Feed-Forward durchgeführt wird.

Beim Arbeiten mit einem normalen PID-Algorithmus muss FFVEL immer die selbe Größe wie der KDER-Anteil haben, um eine typische KDER-Dämpfung zu erreichen.

Inhalt

Geschwindigkeits-Feed-Forward

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

%

Bereich

0 ... 65000 ★ 0

37 FFACC (708)

Geben Sie eine Grundbeschleunigung vor, wenn Sie die Regelung in der Bandbreite begrenzt haben.

Damit verhindern Sie, dass die Regelung durch die eingestellte Begrenzung überhaupt nicht beschleunigt. FFACC gibt den Wert an, mit dem Beschleunigungs-Feed-Forward durchgeführt wird.

Bei einem normalen PID-Algorithmus beträgt dieser Wert 0.

Inhalt

Beschleunigungs-Feed-Forward

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

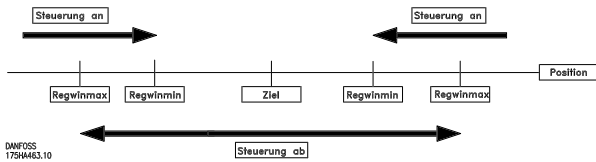
%

Bereich

0 ... 65000 ★ 0

38 REGWMAX

Die Parameter REGWMAX und REGWMIN (39) werden benutzt, um die Lageregelung innerhalb von definierten Bereichen (Regelfenster) an- und abzuschalten zu können: REGWMAX gibt dabei die Größe des Fensters an, außerhalb dessen die Regelung wieder beginnen soll.


Inhalt

Größe des Regelfensters (Aktivierung)

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

qc

Bereich

0 ... MLONG ★ 0

39 REGWMIN

REGWMIN gibt die Größe des Fensters an, innerhalb dessen die Regelung deaktiviert werden soll, bis wieder das Regelfenster REGWMAX (38) erreicht wird.

Inhalt

Größe des Regelfensters (Deaktivierung)

Parametergruppe

Achsparemeter Regelung AXR
CAM-Editor: Registerkarte Regelung

Einheit

qc

Bereich

0 ... MLONG ★ 0

40 HOMETYPE
Inhalt

Homefahrt Verhalten

Parametergruppe

Achsparemeter Home AXH
CAM-Editor: Registerkarte Homefahrt

Bereich

0 ... 3 ★ 0
 0 = Bis zum Referenzschalter mit Home-Geschwindigkeit und -Richtung fahren, dann Reversieren und langsam den Schalter verlassen, anschließend auf den nächsten Indeximpuls fahren.
 1 = wie 0, aber ohne Suchen des Indeximpulses
 2 = wie 0 aber ohne Reversieren, sondern in gleicher Richtung weiter aus dem Schalter heraus
 3 = wie 1 aber ohne Reversieren

41 HOME_RAMP

Beschleunigung, die für die Homefahrt verwendet wird. Die Angabe bezieht sich auf die minimale Rampe, die im Parameter RAMPMIN (31) definiert wird. Die Einheit ergibt sich durch den Parameter VELRES (22), standardgemäß in % von der minimalen Rampe; 50 % bedeutet dann halb so schnell, d.h. doppelt so lange.

Für HOME_RAMP ergibt sich folgender Zusammenhang:

$$\text{HOME_RAMP [ms]} = \frac{\text{VELRES (22)}}{\text{HOME_RAMP (41)}} \cdot \text{RAMPMIN (31) [ms]}$$

Inhalt

Rampe für Homefahrt normiert auf Einheit VELRES (22)

Parametergruppe

Achsparemeter Home AXH
CAM-Editor: Registerkarte Homefahrt

Einheit

VELRES

Bereich

1 ... 65535 ★ 10


ACHTUNG!

HOME_RAMP kann nie einen höheren Wert haben als die DFLTACC (34).

42 HOME_OFFSET

HOME_OFFSET wird benutzt, um einen Offset (Versatz) einzuführen, vergleichbar mit dem Referenzschalter oder Indexpuls. Nach der Homefahrt wird der Antrieb auf HOME_OFFSET positioniert. An dieser Stelle wird auch der Nullpunkt definiert.

Inhalt

Nullpunkt-Offset bezüglich Maschinennullpunkt (MN) bzw. Index

Parametergruppe

Achsparemeter Homefahrt AXH
CAM-Editor: Registerkarte Homefahrt

Einheit

qc

Bereich

-MLONG ... MLONG ★ 0

43 ERRCOND
Inhalt

Verhalten im Fehlerfall

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Eingänge/Ausgänge

Bereich

0 ... 3 ★ 0

- 0 = Standard, d.h. Antrieb geht in Freilauf (COASTING), Regelschleife wird aufgetrennt.
- 1 = wie 0, aber Bremse wird aktiviert; siehe O_BRAKE (48)
- 2 = Motor Stop mit max. Verzögerung (Stop-Rampe), anschließend stillstandgeregelt
- 3 = wie 2, zusätzlich wird die Bremse aktiviert, aber erst nach MOTOR STOP.
Alle weiteren Aktivitäten, wie MOTOR OFF u.ä. müssen in der ON_ERROR-Routine gesetzt werden.

44 ENDSWMOD

ENDSWMOD gibt an, wie sich die Steuerung bzw. der VLT bei Erreichen eines Endschalters verhalten soll.

Fehlerverhalten siehe ERRCOND (43).

Inhalt

Verhalten bei Endschalter

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Eingänge/Ausgänge

Bereich

0 ... 1 ★ 0
0 = Fehler auslösen
1 = Motor Stop mit maximaler Verzögerung

45 I_REFSWITCH

I_REFSWITCH legt fest, welcher Eingang der Optionskarte als Referenzschalter dienen soll. Dabei kann man festlegen ob auf eine positive oder negative Flanke reagiert werden soll.

Verhalten bei Erreichen siehe HOMETYPE (40).

Inhalt

Referenzschalter Eingang

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Eingänge/Ausgänge

Bereich

1 ... 8 = Reaktion auf positive Flanke an Eingang 1 ... 8
0 = keine Funktion ★ 0
-1 ... -8 = Reaktion auf negative Flanke an Eingang 1 ... 8

46 I_POSLIMITSW

I_POSLIMITSW legt fest, welcher Eingang als positiver Endschalter interpretiert werden soll. Dabei kann man festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.

Inhalt

Endschalter positiv

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Eingänge/Ausgänge

Bereich

1 ... 8 = Reaktion auf positive Flanke an Eingang 1 ... 8
0 = keine Funktion ★ 0
-1 ... -8 = Reaktion auf negative Flanke an Eingang 1 ... 8

47 I_NEGLIMITSW

I_NEGLIMITSW legt fest, welcher Eingang als negativer Endschalter interpretiert wird. Dabei kann man festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.

Inhalt

Endschalter negativ

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Eingänge/Ausgänge

Bereich

1 ... 8 = Reaktion auf positive Flanke an Eingang 1 ... 8
0 = keine Funktion ★ 0
-1 ... -8 = Reaktion auf negative Flanke an Eingang 1 ... 8

48 O_BRAKE

O_BRAKE gibt den Ausgang an, mit dem die Bremse aktiviert werden kann.

NEU:

Wenn ein Ausgang für die Bremse definiert ist, bleibt dieser auch dann aktiviert, wenn das Programm mit ESC abgebrochen wird.

O_BRAKE wird nur bei einem Abbruch oder Fehler der Optionskarte gesetzt, wenn ERRCOND (43) auf 1 oder 3 gesetzt ist.

Eine positive Zahl bedeutet, dass der Ausgang high (24 Volt) ist, wenn die Funktion aktiv ist. Eine negative Zahl bedeutet, dass der Ausgang low (0 Volt) ist, wenn die Funktion aktiv ist.


ACHTUNG!

Der Ausgang für die Bremse muss immer mit einem OUT Befehl im Programm zurückgesetzt werden.

Inhalt

Ausgang für Bremse

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI
CAM-Editor: Registerkarte Eingänge/Ausgänge

Bereich

-8 ... 8 und
-14, -11, 11, 14 ★ 0

Beispiel

```
ON ERROR GOSUB err_handle
SET O_BRAKE -1
SET ERRCOND 1
/* Hauptprogrammschleife */
...
SUBPROG err_handle
    WAIT 1
    ERRCLR
    OUT 1 1
RETURN
```


49 SYNCFACTM

Die Synchronisation wird mit einem Verhältnis von qc beschrieben (Master : Slave); SYNCFACTM bestimmt den Synchronisationsfaktor für den Master. SYNCFACTM (49) und SYNCFACTS (50) ermöglichen den Ausgleich unterschiedlicher Getriebefaktoren bzw. die Anpassung der Slavegeschwindigkeit im Verhältnis zur gegebenen Mastergeschwindigkeit.

$$\text{Slave Geschwindigkeit} = \text{Master Geschwindigkeit} * \frac{\text{SYNCFACTS (50)}}{\text{SYNCFACTM (49)}}$$

In Verbindung mit der CAM-Synchronisation werden die Parameter SYNCFACTM und SYNCFACTS zur Umrechnung der qc in MU-Einheiten benutzt. Dadurch kann der Anwender im CAM-Editor mit sinnvollen Einheiten arbeiten. Siehe Beispiel 2.

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ MU}$$

vorausgesetzt dass:

$$\text{Getriebefaktor} = \frac{\text{Motorumdrehungen}}{\text{Umdrehungen am Abtrieb}}$$

- Encoder = Inkremental-Encoder
(bei absoluten Encodern entfällt der Multiplikator 4)
- Skalierfaktor = Anzahl der Benutzereinheiten BE (qc), die einer Umdrehung am Antrieb entsprechen

Inhalt

Synchronisationsfaktor Master (M:S)
bzw. im CAM-Mode Umrechnung der Einheiten qc in MU

Parametergruppe

Achsparameter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

qc

Bereich

- MLONG ... MLONG ★ 1
-MLONG ... -1 = dreht die Richtung der Synchronisation (Verhältnis zum Master) um

Beispiel 1

Wenn der Master zweimal so schnell fahren soll wie der Slave, dann ist das Verhältnis:
2 : 1 (SYNCFACTM = 2; SYNCFACTS = 1)

Beispiel 2: Transportband

Die Eingabe soll in 1/10 mm Auflösung möglich sein. Der Antrieb ist mit dem Transportband mit einer Getriebeübersetzung von 25:11 verbunden; das heißt der Motor macht 25, das Zahnriemenrad 11 Umdrehungen.

Getriebefaktor = 25/11

Inkrementaldrehgeber direkt am Master-Antrieb;
Encoder-Auflösung = 4096

Das Zahnriemenrad hat 20 Zähne/Umdrehung, 2 Zähne entsprechen 10 mm, daher entspricht 1 Umdrehung = 100 mm Transportbandvorschub. Skalierfaktor ist demnach 1000

$$\frac{25/11 * 4096 * 4}{1000} \text{ qc} = 1 \text{ MU}$$

$$\frac{25 * 4096 * 4}{1000 * 11} \text{ qc} = \frac{2048}{55} \text{ qc} = 1 \text{ MU} = \frac{\text{SYNCFACTM (49)}}{\text{SYNCFACTS (50)}}$$

Um mit 1/10 Grad Einteilung zu arbeiten, setzen Sie die Parameter

- SYNCFACTM = 2048
SYNCFACTS = 55

Beispiel 3: Berechnung des Skalierfaktors bei einem Reibantrieb:

Der Abtrieb sei mit einem Reibrad (Radius 60 mm) versehen; wir wollen mit einer Auflösung von 1/10 mm arbeiten:

Ein Umdrehung am Abtrieb berechnet sich demnach:

$$\text{Skalierfaktor} = 2 \Pi r * 10 = 2 \Pi * 60 * 10 = 3969,91$$

$$\text{Skalierfaktor} = 3970$$

Da durch das Aufrunden auf jeden Fall ein Fehler entsteht, muss nach jeder vollen Umdrehung einen Markerabgleich durchgeführt werden.

50 SYNCFACTS
Inhalt

Synchronisationsfaktor Slave (M:S)
bzw. im CAM-Mode Umrechnung der Einheiten qc in MU

Beschreibung

Die Synchronisation wird mit einem Verhältnis von qc beschrieben (Master : Slave). SYNCFACTS bestimmt den Synchronisationsfaktor für den Slave. Die Parameter SYNCFACTM (49) und SYNCFACTS (50) ermöglichen den Ausgleich unterschiedlicher Getriebefaktoren bzw. die Anpassung der Slave-Geschwindigkeit im Verhältnis zur gegebenen Master-Geschwindigkeit.

$$\text{Slave Geschwindigkeit} = \text{Master Geschwindigkeit} * \frac{\text{SYNCFACTS (50)}}{\text{SYNCFACTM (49)}}$$

In Verbindung mit der CAM-Synchronisation werden die Parameter SYNCFACTM und SYNCFACTS zur Umrechnung der qc in MU-Einheiten benutzt. Dadurch kann der Anwender im CAM-Editor mit sinnvollen Einheiten arbeiten. Siehe Voraussetzung der Formel und Beispiel bei SYNCFACTM (49).

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ MU}$$

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

qc

Bereich

1 ... 2 * MLONG/max. Master-Geschwindigkeit ★ 1

wobei gilt:

Einheit der max. Master-Geschwindigkeit
= QC/PROFTIME(29)

Beispiel

Siehe SYNCFACTM (49)

51 SYNCTYPE
Inhalt

Synchronisationstyp: Normale Synchronisation (0)
oder mit Look-ahead Synchronisation (1).

Beschreibung

Die Art, wie die Synchronisation durchgeführt wird, kann geändert werden:

Im Standardfall (SYNCTYPE=0) wird die Positionsdifferenz ausgeglichen.

Das bedeutet, dass die aktuelle Master-Position (wo der Master jetzt ist) mit der künftigen Slave-Position (wo der Slave in 1 msec sein wird) verglichen wird. So wird man immer hinter dem Master fahren, so lange man nicht INTEGRAL benutzt.

Wenn man SYNCTYPE = 1 wählt, vergleicht das System die aktuelle Master-Position mit der aktuellen geforderten Position. Das heißt, dass das System versuchen wird, die Differenz der Positionen auf Null zu bringen, egal wie PID gesetzt ist.


ACHTUNG!

Vergegenwärtigen Sie sich, dass SYNCERR auf eine aktuellen (neuen) Master-Sollposition angleicht, minus der aktuellen Positionen des Slaves plus unerledigter Filterfehler und Korrekturen.

Parametergruppe

Achsparemeter Synchronisation AXS
(Noch nicht im Parameter-Dialogfenster realisiert.)

Bereich

0 ... 1 ★ 0

0 = Die aktuelle Master-Position wird mit der künftigen Slave-Position (wo der Slave in 1 msec sein wird) verglichen. (So wird es mit der PC- Version < 5.04 gemacht.)

1 = Die aktuelle Master-Position wird mit der aktuellen Soll-Position verglichen.

52 SYNCMARKM

SYNCMARKM und SYNCMARKS müssen entsprechend dem Verhältnis der Anzahl der Markersignale des Master zum Slave gesetzt werden. Ein Verhältnis von 1:1 bedeutet, dass sich jeder Slavemarker auf jeden Mastermarker abstimmt. Ein Verhältnis von 2:1 bedeutet, dass sich jeder Slavemarker auf jeden zweiten Mastermarker abstimmt.

Inhalt

Markeranzahl Master

Parametergruppe

Achsparameter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Bereich

0 ... 65535 ★ 1

53 SYNCMARKS

SYNCMARKM und SYNCMARKS müssen entsprechend dem Verhältnis der Anzahl der Markersignale des Master zum Slave gesetzt werden. Ein Verhältnis von 1:1 bedeutet, dass sich jeder Slavemarker auf jeden Mastermarker abstimmt. Ein Verhältnis von 2:1 bedeutet, dass sich jeder Slavemarker auf jeden zweiten Mastermarker abstimmt.

Inhalt

Markeranzahl Slave

Parametergruppe

Achsparameter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Bereich

0 ... 65535 ★ 1

Beispiel

Der Mastermarker ist ein externes Signal, der meldet, dass ein Transportgut ankommt; der dazugehörige Slavemarker ist der Indeximpuls vom Motor. Wenn der Motor immer 3 Umdrehungen benötigt, bis ein Gut ankommt, dann bedeutet das, dass immer 3 Indeximpulse vergehen müssen, bis ein Marker kommt. Daraus ergibt sich ein Verhältnis von 3 : 1; nur jeder 3. Slavepuls wird ausgewertet.

54 SYNCPOSOFFS

Setzt den Offset für die Positionssynchronisation (SYNCM, SYNCP). Der Offset ist auch bei einer Positionssynchronisation mit Markerkorrektur gültig. Dieser **Positionsoffset** kann jederzeit per Befehl während der Synchronisation online verändert werden.

Inhalt

Positionsoffset bei Positionssynchronisation


ACHTUNG!

Der Offset für die Positionssynchronisation wird sofort ausgeführt, wenn der Befehl SYNCP folgt.

NEU

Beim Start von SYNCM dagegen wird auf die erste Auswertung der Markerpulse gewartet. Erst dann wird der Offset angewandt.

Kompatibilität

Zur Vermeidung von Kompatibilitätsproblemen sollten Sie das Startverhalten von SYNCM mit SYNCMSTART festlegen.

Parametergruppe

Achsparameter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

qc

Bereich

-MLONG/SYNCFAC(50) ...
MLONG/SYNCFAC(50) ★ 0

55 SYNCACCURACY

Gibt an, wie groß die Differenz zwischen aktueller Master- und Slaveposition bei einer Positionssynchronisation (SYNCP und SYNCM) sein darf, damit die geforderte Genauigkeit (ACCURACY) noch erfüllt ist. SYNCERR dagegen liefert den tatsächlichen Synchronisationsfehler des Slaves in Benutzereinheiten bzw. im CAM-Mode in BE.

Ob SYNCACCURACY erfüllt wird, können Sie im Programm mit SYNCSTAT abfragen.

SYNCACCURACY ist wichtig für die Markersynchronisation um READY melden zu können, da andernfalls vorher n-mal SYNCERR abgefragt und verglichen werden müsste.

Inhalt

Größe des Genauigkeitsfensters für Positionssynchronisation

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

qc
bzw. im CAM-Mode: BE

Bereich

-MLONG ... MLONG ★ 1000

0 ... MLONG	Positives Vorzeichen liefert den absoluten Wert an SYNCERR.
-MLONG ... -1	Negatives Vorzeichen liefert den Synchronisationsfehler an SYNCERR mit Vorzeichen. Daraus lässt sich dann erkennen, ob die Synchronisation voraus- oder nachläuft.

56 SYNCREADY

Gibt an wie oft bei einer Markersynchronisation (SYNCP und SYNCCMM) eine Synchronisationsbewertung mit ACCURACY durchgeführt sein muss, bis READY erfüllt ist.

Dabei wird bei jeder Korrektur ACCURACY geprüft. Wenn ACCURACY erfüllt ist, wird 1 addiert, bis die vorgegebene Markeranzahl erreicht ist.

Die Synchronisationsbewertung wird immer erst nach n Markerimpulsen beim Master SYNCMARKM (52) durchgeführt.

ACCURACY und READY können Sie mit SYNCSTAT abfragen.

Inhalt

Markeranzahl für READY

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Bereich

0 ... 65535 ★ 1

57 SYNCFAULT

Gibt an wie oft bei einer Markersynchronisation (SYNCP und SYNCCMM) nicht ACCURACY auftauchen darf, bis FAULT eintritt.

Diesen Zustand können Sie im Programm mit SYNCSTAT abfragen.

Inhalt

Markeranzahl für Fault

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Bereich

0 ... 65535 ★ 10

58 SYNCMPULSM

SYNCMPULSM gibt an wie viele qc (Master) zwischen zwei Mastermarkern liegen.
 Wenn man den Drehgeber-Indeximpuls als Marker-signal benutzt, beträgt der Abstand zwischen zwei Markern die Auflösung (qc) des Drehgebers.
 Wenn externe Markersignale benutzt werden, können Sie den Markerabstand mit dem Programm "Marker count" (siehe Programmbeispiele im Kapitel 7) messen, falls er nicht bekannt ist.
 SYNCMPULSM gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNCCMM).
 Bei einer CAM-Synchronisation wird statt des Abstandes zwischen zwei Master-Markern der Abstand des Sensors zur Arbeitsposition in MU angegeben. (Der Abstand ergibt sich automatisch durch die Mastertaktlänge [Mt].)
 Wenn der Parameter größer als eine Mastertaktlänge [Mt] ist, wird automatisch ein Marker-FIFO-Register für die Handhabung der Markerkorrektur gebildet.

Inhalt

Abstand zwischen zwei Mastermarkern, bzw. im CAM-Mode Abstand zwischen Sensor und Arbeitsposition

Parametergruppe

Achsparemeter Synchronisierung AXS
 CAM-Editor: Registerkarte Synchron.

Einheit

qc
 im CAM-Mode: MU

Bereich

0 ... 2 * MLONG / (SYNCMARKM(52) * SYNCFAC(50) * n)
 wobei
 n = Anzahl der Marker, die zwischen Master und Slave sein können, wenn der Slave versucht, den Master beim Start einzuholen.
 ★ 500

59 SYNCMPULSS

SYNCMPULSS gibt an wie viele qc (Slave) zwischen zwei Markern (Slave) liegen, bzw. im CAM-Mode der Abstand des Sensors zur Arbeitsposition in BE.
 SYNCMPULSS gilt nur für Synchronisationen mit Markerkorrektur (und SYNCCMS).

Inhalt

Markerabstand Slave, bzw. im CAM-Mode Abstand zwischen Sensor und Arbeitsposition

Parametergruppe

Achsparemeter Synchronisierung AXS
 CAM-Editor: Registerkarte Synchron.

Einheit

qc
 im CAM-Mode: BE

Bereich

0 ... 2 * MLONG / 9 ★ 500

60 SYNCMTYPM

Stellen Sie den Signal- bzw. Markertyp für den Master ein: Indexpuls des Drehgebers oder externer Marker.
 SYNCMTYPM gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNCCMM) oder wenn Sie den Befehl MIPOS in Ihrem Programm verwenden wollen.
 Master Markersignal: Input 5

Inhalt

Markertyp Master

Parametergruppe

Achsparemeter Synchronisierung AXS
 CAM-Editor: Registerkarte Synchron.

Querverweis

MIPOS, SYNCMTYPS

Bereich

0 ... 3 ★ 0
 0 = Indexpuls (positive Flanke)
 1 = Indexpuls (negative Flanke)
 2 = Externer Marker (positive Flanke)
 3 = Externer Marker (negative Flanke)

61 SYNCMTYPS

Stellen Sie den Signal- bzw. Markertyp für den Salve ein: Indexpuls des Drehgebers oder externer Marker. SYNCMTYPS gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNCCMS) oder wenn Sie den Befehl IPOS in Ihrem Programm verwenden wollen.

Slave Markersignal: Input 6

Inhalt

Markertyp Slave

Parametergruppe

Achsparameter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Querverweis

IPOS, SYNCMTYPM

Bereich

0 ... 3 ★ 0

- 0 = Indexpuls (positive Flanke)
- 1 = Indexpuls (negative Flanke)
- 2 = Externer Marker (positive Flanke)
- 3 = Externer Marker (negative Flanke)

62 SYNCMSTART

SYNCMSTART gibt an ob beim Starten der Synchronisation auf den jeweils voreilenden, nachfolgenden oder auf den dichtesten Markerimpuls des Masters aufsynchronisiert werden soll.

SYNCMSTART gilt nur für Synchronisationen mit Markerkorrektur (SYNCM und SYNCCMM).

Inhalt

Startverhalten für Synchronisationen mit Markerkorrektur

Parametergruppe

Achsparameter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Bereich

0 ... 6, 1000 ... 1006 und 2000 ★ 0

- 0 = Der Slave-Marker, der dem ersten Master-Marker (nach SYNCM) folgt, wird mit dem ersten Master-Marker abgeglichen.
- 1 = Der erste Slave-Marker (nach SYNCM) wird mit dem folgenden Master-Marker abgeglichen.
- 2 = Nach Erreichen der Master-Geschwindigkeit werden die nächsten zwei Marker abgeglichen. (Korrektur durch Aufholen oder Abbremsen)
- 3 = Nach Erreichen der Master-Geschwindigkeit wird der nächste Slave-Marker mit dem davor liegenden Master-Marker abgeglichen. (Korrektur durch Aufholen).
- 4 = Nach Erreichen der Master-Geschwindigkeit wird der nächste Slave-Marker mit dem nachfolgenden Master-Marker abgeglichen. (Korrektur durch Abbremsen).
- 5 = Nach Erreichen der Master-Geschwindigkeit wird der nächste Slave-Marker mit dem Master-Marker abgeglichen, der am dichtesten folgt. (Korrektur durch Aufholen oder Abbremsen, je nach kürzestem Abstand.)
- 6 = Nach dem Befehl SYNCM werden die ersten zwei Marker genommen und auf diese aufsynchronisiert.
- 1000 ... 1006 = wie oben, aber ein bestehender Offset wird erst bei der nächsten Markerkorrektur angewendet.
- 2000 = im CAM-Mode: Das Zählen (Beobachten) der Masterpulse in MU beginnt mit dem Mastermarker.


ACHTUNG!

Der Parameter 2000 wirkt nur bei Kurvenscheibensteuerungen.

63 REVERS

REVERS legt das Verhalten beim Reversieren (Fahren in negativer Richtung) fest: ob Reversierung erlaubt, nur erlaubt, wenn der Master reversiert oder grundsätzlich nicht erlaubt ist.

REVERS ist immer gültig, auch bei Positionierbefehlen, VEL und "TESTFAHRT".

Wenn Sie das automatische Reversieren bei der "TESTFAHRT" verhindern wollen, stellen Sie den Wert auf 1 oder 2 ein.

Inhalt

Reversierungsverhalten Slave

Parametergruppe

Achsparemeter Regelung AXR

CAM-Editor: Registerkarte Regelung

Bereich

0 ... 2 und 10 ... 12 ★ 0

0 = Reversierung immer erlaubt, Sollwert ± 10 V

1 = Reversierung nur erlaubt, wenn der Master reversiert, Sollwert ± 10 V

2 = Reversierung nicht erlaubt, Sollwert ± 10 V

64 O_AXMOVE

Mit diesem Parameter setzen Sie die Ausgangsnummer, die durch die O_AXMOVE Funktion kontrolliert werden muss. Der Ausgang ist immer aktiviert, sobald ein Fahrbefehl aktiv ist, unabhängig in welchem Modus (Positions-, Geschwindigkeits- oder Synchronisationsbefehl).

Diese Funktion eignet sich nicht für die Motorüberwachung, denn der Motor könnte stillstehen, obwohl die Steuerung in Bewegung ist.

Eine positive Zahl bedeutet, dass der Ausgang high (24 Volt) ist, wenn die Funktion aktiv ist. Eine negative Zahl bedeutet, dass der Ausgang low (0 Volt) ist, wenn die Funktion aktiv ist.

Inhalt

Ausgang für Fahrbefehl in Aktion

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI

CAM-Editor: Registerkarte Ein-/Ausgänge

Bereich

-8 ... 8 ★ 0

65 SYNCVFTIME (709)

Dieser Parameter konfiguriert den Geschwindigkeitsfilter, der für die Geschwindigkeitssynchronisation verwendet wird. Da bei der Geschwindigkeitssynchronisation nur mit der jeweils aktuellen Master-Geschwindigkeit gearbeitet wird und diese sehr kleine Werte annehmen kann (z.B. 2 qc/ms), wirkt sich eine kleine Schwankung der Geschwindigkeit bereits dramatisch aus. Um dies zu glätten wird die folgende Filterfunktion verwendet:

Cmdvel
 = Old_Cmdvel + (Actvel – Old_Cmdvel) * ms/τ_filt
 Hierbei gilt:

- Cmdvel = Soll-Geschwindigkeit
- Old_Cmdvel = Letzte Soll-Geschwindigkeit
- Actvel = Aktuelle Geschwindigkeit des Masters
- ms = Abtastzeit (fest 1ms)
- τ_filt = Filterzeit Konstante

Dabei wird der Wert für τ_filt standardgemäß aus einer Tabelle genommen, in Abhängigkeit von der Drehgeberauflösung des Masters. Dieser Wert kann durch den Parameter SYNCVFTIME überschrieben werden und wird immer dann verwendet, wenn SYNCVFTIME ungleich Null ist.

Wird der Geschwindigkeitsfilter mit einer negativen Zahl definiert, gilt der entsprechende Wert auch für Winkel-/Positionssynchronisation SYNCP und für Markerkorrektur SYNCM.

Es wird in diesem Fall ebenso gefiltert wie oben beschrieben, zusätzlich jedoch der gemachte Fehler aufsummiert. Diese Fehlersumme wird jeweils zu 1000/(t*10) in die Berechnung mit einbezogen, so dass über längere Zeiträume keine Positionsabweichung entstehen kann.

Der von SYNCERR zurückgelieferte Wert enthält immer den gemachten Fehler, so dass dieser auch bei der Bewertung der Synchronität einfließt. Ebenfalls wird bei Markerkorrektur der Korrekturwert langsamer ausgeglichen, und zwar mit dem selben Faktor wie die Fehlersummen.

Setzt man zum Beispiel einen Filterfaktor von –100000 (100 ms) wird eine Markerkorrektur innerhalb von 1 Sekunde (100 ms * 10) ausgeglichen. Diese ermöglicht eine „Zähmung“ der Synchronisation ohne die Beschleunigung einzuschränken.

Inhalt

Geschwindigkeitsfilter

Syntax

SET SYNCVFTIME wert
 wert = Filterzeit Konstante

Parametergruppe

Achsparemeter Synchronisierung AXS
 CAM-Editor: Registerkarte Synchron.

Einheit

τ_filt (µsec)

Bereich

–MLONG ... MLONG ★ 0
 –999 ... 999 = Standardtabelle

Standardtabelle

Drehgeber Auflösung	t_filt (msec)
250	39500
256	38600
500	19500
512	19000
1000	9500
1024	9300
2000	4500
2048	4400
2500	3500
4096	1900
5000	1400

66 SYNCVELREL

Dieser Parameter gibt an, um wieviel Prozent der Folgeantrieb von der Geschwindigkeit des Masters abweichen darf, während er versucht die Synchronisation wieder herzustellen. Zum Beispiel bei der Veränderung von SYNCPOSOFFS (54), beim Synchronisationsstart oder bei der Korrektur der Abweichung bei der Markerauswertung. Dabei gilt Folgendes: Muss der Slave aufholen, fährt er mit der maximalen Geschwindigkeit die erlaubt ist, wobei dies entweder die mit VEL eingestellte Geschwindigkeit oder die durch

$MAVEL + MAVEL * SYNCVELREL/100$
berechnete, je nachdem welche von beiden kleiner ist. (MAVEL ist aktuelle Mastergeschwindigkeit).

Muss der Slave abbremsen und auf den Master warten, fährt er minimal mit der Geschwindigkeit $MAVEL - MAVEL * SYNCVELREL/100$.

Das heißt, wenn SYNCVELREL zum Beispiel 50 ist, wird der Folgeantrieb nicht langsamer als MAVEL/2.

Inhalt

Tolerierte Abweichung des Folgeantriebs von der Master-Geschwindigkeit in %

Syntax

SET SYNCVELREL wert
wert = Prozentwert

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

%

Bereich

0 ... MLONG ★ 0
0 = Funktion ist ausgeschaltet, d.h.
keine Beschränkung

67 MENCODERTYPE

Der Parameter legt den Drehgebertyp für den Master fest.

Mit dem Encodertyp „6“ lässt sich ein Master per APOSS-Befehl simulieren, zum Beispiel wenn die Master-Position über den Bus gelesen wird. Die simulierten Master-Positionen werden mit der Systemvariablen SYSVAR[4105] gesetzt und gelesen.

Inhalt

Drehgebertyp für Master-Encoder

Parametergruppe

Achsparemeter Encoder AXE
CAM-Editor: Registerkarte Encoder

Bereich

0 ... 2, 6 und 100 ... 102 ★ 0
0 = Inkrementaler Encoder
1 = Absoluter Encoder, Standard ca. 262 kHz
2 = Absoluter Encoder, ca. 105 kHz
6 = Die Master-Position wird nicht vom Drehgeber gelesen, sondern mit der Systemvariablen SYSVAR[4105] gesetzt.
(siehe Programmbeispiel)
100 ... 102 = wie 0 ... 2, jedoch wird dann die Hardware-Überwachung des Encoders eingeschaltet. Wenn die Encoder-Leitungen unterbrochen sind oder bei Kurzschluss, wird der Fehler 92 ausgegeben.


ACHTUNG!

Mit absoluten Drehgebern können folgende Befehle nicht benutzt werden: DEF ORIGIN, HOME, INDEX und WAITNDX.


ACHTUNG!

Mit absoluten Drehgebern können folgende Befehle nur dann benutzt werden, wenn externe Marker eingesetzt werden: IPOS, MIPOS.

Programmbeispiel

syncc_msim.m

68 SYNCMWINM

Das **Marker Window Master** SYNCMWINM gibt an, wie groß die erlaubte Toleranz für das Auftreten der Marker ist.

Mit der Werkseinstellung „0“ wird das Fenster nicht überwacht, das heißt es wird immer auf den nächsten Marker synchronisiert, auch wenn dieser einen wesentlich größeren Abstand hat.

Mit jeder anderen Einstellung werden nur Marker akzeptiert, die innerhalb des Fensters liegen. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Es wird auch der entsprechende andere Marker ignoriert und erst beim nächsten Mal wieder korrigiert – also kein Aufholen zum nächsten Marker.

Nach dem Start von SYNCM bzw. nach SYNCSTART beginnt die Überwachung erst, wenn der erste Marker gefunden ist.

Inhalt

Toleranzfenster für das Auftreten der Mastermarker (Marker monitoring)

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

qc
bzw. im CAM-Mode: MU


ACHTUNG!

Änderungen des Parameters werden sofort aktiv – nicht erst nach dem nächsten SYNCM-Befehl.

Bereich

0 ... MLONG bzw. max. **Markerabstand**
SYNCMPULSM (58) ★ 0

0 = Fenster wird nicht überwacht.
1 ... MLONG = Es wird nur ein Marker akzeptiert, der innerhalb des Fensters liegt. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Dieses Flag kann mit einem Interrupt (ON STATBIT) zurückgesetzt werden.

Beispiel

Markerabstand SYNCMPULSM = 30000

Toleranzfenster SYNCMWINM = 1000

Es wird nur der Marker akzeptiert, der innerhalb des Intervalls von 29000 bis 31000 liegt.

69 SYNCMWINS

Das **Marker Window Slave** definiert die erlaubte Toleranz für das Auftreten der Marker. Mit der Werkseinstellung „0“ wird das Fenster nicht überwacht, das heißt es wird immer auf den nächsten Marker synchronisiert, auch wenn dieser einen wesentlich größeren Abstand hat.

Mit jeder anderen Einstellung werden nur Marker akzeptiert, die innerhalb des Fensters liegen. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Es wird auch der entsprechende andere Marker ignoriert und erst beim nächsten Mal wieder korrigiert – also kein Aufholen zum nächsten Marker.

Nach dem Start von SYNCM bzw. nach SYNCSTART beginnt die Überwachung erst, wenn der erste Marker gefunden ist.

Inhalt

Toleranzfenster für das Auftreten der Slavemarker

Parametergruppe

Achsparemeter Synchronisierung AXS
CAM-Editor: Registerkarte Synchron.

Einheit

qc
bzw. im CAM-Mode: BE


ACHTUNG!

Parameteränderungen werden sofort aktiv – nicht erst nach dem nächsten SYNCM-Befehl.

Bereich

0 ... MLONG bzw. max. Markerabstand
SYNCMPULSS (59) ★ 0

0 = Fenster wird nicht überwacht.
1 ... MLONG = Es wird nur ein Marker akzeptiert, der innerhalb des Fensters liegt. Wenn innerhalb des Toleranzfensters kein Marker kommt, wird das entsprechende Flag (SYNCSTAT) gesetzt und keine Markerkorrektur durchgeführt. Dieses Flag kann mit einem Interrupt (ON STATBIT) zurückgesetzt werden.

70 ESCCOND

Mit diesem Parameter legen Sie fest, wie der VLT bei Programmabbruch durch ESC reagieren soll.

Inhalt

Verhalten bei Programmabbruch

Parametergruppe

Achsparemeter Eingänge/Ausgänge AXI

CAM-Editor: Registerkarte Ein-/Ausgänge

Bereich

0 ... 2 ★ 0

0 = Der Motor wird mit maximaler Verzögerung gestoppt, die Bremse wird aktiviert (falls definiert), die Mastersimulation wird gestoppt. Die Ausgänge bleiben im aktuellen Status.

1 = Wie 0, aber alle Ausgänge einschließlich der VLT5000-Ausgänge 11, 14, 42 und 45 (bzw. der VLT5000Flux Ausgänge 26 und 46) werden auf „0“ gesetzt.

Ausnahme: Der Bremsausgang wird – falls definiert – immer aktiviert.

2 = Wie „0“, aber alle Ausgänge einschließlich der VLT5000-Ausgänge 11, 14, 42 und 45 (bzw. der VLT5000Flux Ausgänge 26 und 46) werden auf „1“ gesetzt.

Ausnahme: Der Bremsausgang wird – falls definiert – immer aktiviert.

102 PRGPAR (701)

Mit dem PRGPAR legen Sie fest, welches Programm nach Ablauf eines per Autostart (Autokennung) ausgeführten Programmes gestartet werden soll. Dieser Parameter kann auch von Programmen oder per Display geändert und gespeichert werden.

Wenn Sie innerhalb eines Programmes mit SET diesen Programmparameter setzen, z.B. SET PRGPAR 5 wird nach Ablauf des Programmes das Programm Nr. 5 gestartet. So kann man von einem SyncPos-Programm aus andere Programme starten und auch mehrere Programme miteinander verketteten.

Wenn keine Programmnummer aktiviert ist und auch kein Eingang für Programmstart I_PRGSTART (103) gesetzt ist, wird wieder das mit Autokennung versehene Programm gestartet.


ACHTUNG!

Wenn kein Autostartprogramm definiert ist, kann auch kein Programm über PRGPAR (102) gestartet werden; dies erfordert immer ein beendetes Autostartprogramm.

Inhalt

Aktivierte Programmnummer

Parametergruppe

Globale Parameter GLS

Portabilität

Parameterwert +1000 ab Optionskarte Version 5.00

Bereich

-1 ... 127 (optional + 1000) ★ -1

-1 = Programmnummer ist nicht aktiviert, d. h. es wird nach Auto-Exec kein Programm gestartet

0...127 = aktivierte Programmnummer wird gestartet nach dem Einschalten (und AutoExec)

0...127 +1000 = aktivierte Programmnummer wird gestartet, aber beim Einschalten ist der Motor stromlos (MOTOR OFF)

-1 + 1000 = (999) wie -1, aber der Motor ist nach dem (Strom-)Einschalten noch ausgeschaltet.

Beispiel

SET PRGPAR 5

// Programm-Nr. 5 wird sofort nach dem Einschalten // und nach dem Ausführen der AutoExec gestartet

SET PRGPAR 1005

// Programm-Nr. 5 wird gestartet, aber beim // Hochfahren ist der Motor aus

103 I_PRGSTART

Wenn der Eingang für I_PRGSTART $\neq 0$ ist, wird zuerst das Autostart-Programm ausgeführt und anschließend gewartet, bis Eingang I_PRGSTART kommt. Dieser wird entsprechend der Programmwahl I_PRGCHOICE (104) ausgewertet um die Programmnummer zu bestimmen, die ausgeführt werden soll.

Wenn kein Eingang für Programmstart gesetzt ist, wird wieder das mit Autokennung versehene Programm gestartet.

Inhalt

Eingang für Programm Start

Parametergruppe

Globale Parameter GLI

Bereich

0 ... 8 ★ 0

104 I_PRGCHOICE

Wenn I_PRGCHOICE > 0 gesetzt wird, gibt dieser Parameter die Eingangsnummer an, ab der die Eingänge für die **Programmwahl** verwendet werden. Dazu gehören alle bis zu I_PRGSTART (103).

Beispiel:

Wenn I_PRGCHOICE = 3 und I_PRGSTART = 7, werden bei der Aktivierung von Eingang 7 die Eingänge 3, 4, 5, 6 binär ausgewertet und das Ergebnis als Programmnummer verwendet.

Eingang	Pegel	Binärwert
3	low	0
4	high	2
5	high	2 ²
6	low	0

=> zu startendes Programm: 6

Maximal kann somit zwischen 128 Programmen, die mit den Nummern 0 bis 127 gekennzeichnet sind, ausgewählt werden.

Inhalt

Eingang für Programmwahl Anfang

Parametergruppe

Globale Parameter GLI

Bereich

0 ... 8 ★ 0

105 I_BREAK

Wenn dieser Eingang definiert ist, wird bei der Aktivierung des Eingangs sofort das laufende Programm abgebrochen. Mit CONTINUE kann ein solches Programm wieder fortgesetzt werden.

Dabei kann man mit einer positiven oder negativen Zahl festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.

Inhalt

Eingang für Abbruch

Parametergruppe

Globale Parameter GLI

Bereich

1 ... 8 = Reaktion auf positive Flanke an Eingang 1 ... 8
 0 = keine Funktion ★ 0
 -1 ... -8 = Reaktion auf negative Flanke an Eingang 1 ... 8

106 I_CONTINUE

I_CONTINUE legt fest, welcher Eingang zur Fortsetzung von abgebrochenen Programmen verwendet wird. Dabei kann man mit einer positiven oder negativen Zahl festlegen ob auf eine positive oder negative Flanke reagiert werden soll.

Inhalt

Programm fortsetzen

Parametergruppe

Globale Parameter GLI

Bereich

1 ... 8 = Reaktion auf positive Flanke an Eingang 1 ... 8
 0 = keine Funktion ★ 0
 -1 ... -8 = Reaktion auf negative Flanke an Eingang 1 ... 8

107 I_ERRCLR

I_ERRCLR legt fest, welcher Eingang zum Fehler löschen verwendet wird. Dabei kann man mit einer positiven oder negativen Zahl festlegen, ob auf eine positive oder negative Flanke reagiert werden soll.

Inhalt

Fehler löschen

Parametergruppe

Globale Parameter GLI

Bereich

- 1 ... 8 = Reaktion auf positive Flanke
an Eingang 1 ... 8
- 0 = keine Funktion ★ 0
- 1 ... -8 = Reaktion auf negative Flanke
an Eingang 1 ... 8

108 O_ERROR

Der mit O_ERROR definierte Ausgang wird gesetzt, wenn ein Fehler der Option aufgetreten ist. Wenn der Fehler gelöscht wurde, wird dieser Ausgang zurück gesetzt.

Eine positive Zahl bedeutet, dass der Ausgang high (24 Volt) ist, wenn die Funktion aktiv ist. Eine negative Zahl bedeutet, dass der Ausgang low (0 Volt) ist, wenn die Funktion aktiv ist.

Inhalt

Ausgang für Fehler

Parametergruppe

Globale Parameter GLI

Bereich

- 8 ... 8 ★ 0


ACHTUNG!

Die Einstellung des Parameters O_ERROR haben keinen Einfluss auf die Verwendung der Befehle OUT und OUTB. Mit diesen Befehlen kann auch der Zustand der Ausgänge verändert werden, die vordefinierte Funktionen besitzen.

Kapitel 7

■ **Meldungen und Fehler-Referenz**

Meldungen vom VLT und vom SyncPos
Motion Controller 214
Tabelle der Meldungen 214
Meldungen vom VLT und vom SyncPos
Motion Controller im Detail 215
Meldungen von der SyncPos PC Software) 220

■ Meldungen vom VLT und vom SyncPos Motion Controller

Alle Meldungen werden im LCP-Display des VLT in Kurzform und in der SyncPos-Software im Klartext angezeigt.

Informieren Sie sich in der Tabelle in Kürze oder im darauf folgenden Abschnitt im Detail über die Fehlermeldungen.

■ Tabelle der Meldungen

Die Tabelle enthält die Meldungen nach Nummern geordnet. Buchstaben hinter einem %-Zeichen stehen für variable Werte, die im Klartext an den entsprechenden Stellen eingesetzt werden.

O.ERR_	LCP-Anzeige	SyncPos Klartext
3	ACHSE NICHT VORH.	Achse %bu nicht vorhanden
5	FEHLER NICHT BESEIT	Achse: 1 Fehler nicht beseitigt
6	HOME NICHT ANGEF.	Achse: 1 HOME noch nicht angefahren
7	HOME_VEL Null	Achse: 1 HOME Geschwindigkeit 0
8	SCHLEPPFEHLER	Achse: 1 Schleppabstand ueberschritten
9	INDEX NICHT GEFUND.	Achse: 1 Indeximpuls nicht gefunden
10	UNBEKANNTER BEFEHL	Unbekannter Befehl
11	SW ENDSCHALTER	Achse: 1 Software-Endschalter ueberschritten
12	FALSCHER PARAMETERNR	Falsche Parameternummer
13	Genereller VLT Fehler	Genereller VLT Fehler
14	ZUVIELE VERS.CH LOOP	Zu viele verschachtelte LOOP Befehle
16	PARAM. DEF.I.EEPROM	Parameter im EEPROM defekt
17	PROGR. DEF.I.EEPROM	Programme im EEPROM defekt
18	RESET DURCH CPU	Reset durch CPU ausgelöst
19	BENUTZER ABBRUCH	Benutzer Abbruch
49	ZU VIELE INTERRUPTS	Achse: 1 Zu viele Interruptfunktionen
25	ENDSCHALTER ERR.	Achse: 1 Endschalter erreicht
51	ZUVIELE VERS. GOSUB	zu viele verschachtelte Unterprogramme
52	ZUVIELE RETURN	zu viele Return
62	FEHLER EEPROM VERIF	Fehler beim Verifizieren
70	FEHLER IN DIM	Fehler in DIM Anweisung
71	FELDGR. UEBERSCHR.	Feldgrenzen über- oder unterschritten
79	TIMEOUT WAITNDX	Timeout bei Warten auf Index
84	ZUVIELE ONTIME AUFR	Zu viele Time Interrupts
87	VAR.SPEICH.ZU KLEIN	Kein Platz mehr für Variablen
90	MEMORY locked	Programmspeicher ist schreibgeschützt
91	KURVEN ARRAY DEFEKT	Kurven-Array in DIM-Anweisung falsch
92	ENCODER FEHLER	Fehler aus Encoder Überwachung
xx	INTERNER FEHLER	Interner Fehler ##

■ Meldungen vom VLT und vom SyncPos Motion Controller im Detail

Die Meldungen sind wie die vorhergehende Tabelle numerisch geordnet: Hier finden Sie zusätzlich Hinweise zu den möglichen Fehlerursachen und Tips zur Fehlerbehebung.

O.ERR_3

SyncPos

Achse %bu nicht vorhanden

Ursache

Es wurde versucht eine Achse anzusprechen, die nicht in der Steuerung vorhanden ist. (Das Programm erlaubt die Eingabe von Achsnummern größer 1, da es bereits für Mehrachsen-Anwendungen vorbereitet ist.)

Hinweise

Kontrollieren Sie, ob Ihr Programm Achsbefehle mit einer ungültigen Achsnummer oder allgemeine Achsbefehle (...X(*)) enthält.

O.ERR_5

SyncPos

Achse: 1 Fehler nicht beseitigt

Ursache

Es wurde versucht einen Bewegungsbefehl auszuführen, obwohl eine aktuell bestehende Fehlermeldung noch nicht gelöscht wurde.

O.ERR_6

SyncPos

Achse: 1 HOME noch nicht angefahren

Ursache

Gemäß dem Achsparameter HOME_FORCE (3) wird zwingend eine Fahrt zum Maschinennullpunkt gefordert, bevor andere Bewegungsbefehle ausgeführt werden können. Diese Fahrt zum Maschinennullpunkt wurde noch nicht vorgenommen.

O.ERR_7

SyncPos

Achse: 1 HOME Geschwindigkeit 0

Ursache

Es wurde versucht, eine Homefahrt mit Geschwindigkeit 0 auszuführen.

O.ERR_8

SyncPos

Achse: %bu – Schleppabstand überschritten

Bedeutung

Der Abstand zwischen der Soll- und Istposition war größer als im Parameter **Tolerierter Positionsfehler** POSERR (15) definiert.

Ursachen

Mechanisch blockierter oder überlasteter Antrieb, zu kleiner **Tolerierter Positionsfehler** POSERR (15), Sollgeschwindigkeit größer als VLT Parameter 202 und 205, zu große Sollbeschleunigung, zu geringer **Proportionalfaktor** KPROP (11) oder VLT nicht freigegeben.

O.ERR_9

SyncPos

Achse: 1 Indeximpuls nicht gefunden

Bedeutung

Bei einer Referenz- bzw. Indexsuche konnte der Indeximpuls des Encoders nicht innerhalb einer Motorumdrehung gefunden werden.

Ursachen

Es wird ein Drehgeber ohne Indeximpuls verwendet, der Indeximpuls ist nicht korrekt angeschlossen, nicht korrekter Indeximpuls oder der Parameter **Geberstrichzahl** ENCODER (2) ist zu niedrig angegeben.

O.ERR_10

SyncPos

Unbekannter Befehl

Ursache

Es muss ein Kommunikations- oder Programmfehler vorliegen.

Hinweis

Das Programm sollte neu übersetzt und neu geladen werden.

O.ERR_11

SyncPos

Achse: 1 Software-Endschalter überschritten

Ursache

Durch einen Bewegungsbefehl wurden bzw. würden die Software-Endschalter überschritten.

Hinweis

Bei einer Bewegung im Drehzahlmodus wird das Überschreiten der Wegbegrenzung erst erkannt, nachdem die aktuelle Position mit dem Software-Endschalter identisch ist.

In diesem Fall wird die Lageregelung abgeschaltet und der Antrieb muss manuell wieder innerhalb des zulässigen Bereichs bewegt werden. Oder die Überwachung des Software-Endschalters muss kurzzeitig mit Hilfe der Achsparameter SWPOSLIMACT (20) bzw. SWNEGLIMACT (19) deaktiviert werden. Erst danach kann die Fehlermeldung gelöscht werden. Bei einer Positionierbewegung wird vor dem Bewegungsstart bereits erkannt, dass die Zielposition außerhalb der Wegbegrenzung liegt. In diesem Fall wird die Bewegung nicht ausgeführt und die Fehlermeldung kann gelöscht werden.

O.ERR_12

SyncPos

Falsche Parameternummer

Ursache

Es wurde versucht, mit einem SET oder SETVLT Befehl einen Parameter zu verändern, den es nicht gibt.

O.ERR_13

SyncPos

VLT NICHT BEREIT

Ursache

Der VLT ist nicht breit, aber der PID-Regler ist aktiv. Das VLT-Statuswort (Bit 09 and Bit 11) wird alle 20 ms überprüft, wenn der PID-Regler aktiv ist. Der VLT ist im „NICHT BEREIT“-Status:

- bei Alarm,
- wenn er im lokalen Modus ist (Parameter 002 = local),
- wenn der lokale LCP-Stop aktiviert ist.

O.ERR 13 kann durch Toggeln des Eingangs 27 zurückgesetzt werden.

Hinweis

Wenn lokaler Stopp auf dem LCP gedrückt ist, kann Error 13 auf dem Display angezeigt werden. Um dies zu vermeiden, kann man den Parameter 014 auf „disable“ setzen; aber das schaltet auch die Stoppfunktion auf dem lokalen Kontrollpanel aus.

O.ERR_14

SyncPos

Zu viele verschachtelte LOOP Befehle

Ursache

Im ausgeführten Programm sind zu viele ineinander geschachtelte LOOP Befehle.

O_ERR_16

SyncPos

Parameter im EEPROM defekt!

Bedeutung

Der im Speicher abgelegte Parametersatz ist nicht mehr korrekt vorhanden.

Ursachen

EEPROM defekt oder Spannungsausfall während des Speicherns.

Hinweis

Sie müssen die Parameter mit "STEUERUNG" → "PARAMETER" → "RESET" neu initialisieren und diese anschließend mit Ihren Benutzerparametern überschreiben. Fahrprogramme, die Benutzerparameter voraussetzen, würden sonst nicht mehr korrekt funktionieren.

O.ERR_17

 SyncPos

Programme im EEPROM defekt!

Bedeutung

Die im Speicher abgelegten Programmdateien sind nicht mehr vorhanden bzw. nicht mehr korrekt.

Ursachen

EEPROM defekt oder Spannungsausfall während des Speicherns.

Hinweis

Löschen Sie das EEPROM mit "STEUERUNG" → "SPEICHER" → "EEPROM LÖSCHEN" und laden Sie anschließend die Programme und Parameter neu.

O.ERR_18

 SyncPos

Reset durch CPU ausgelöst!

Bedeutung

Der Prozessor wurde angehalten, was wiederum durch eine interne Überwachungsschaltung (Watchdog) erkannt wurde und automatisch einen Reset auslöste.

Ursachen

Kurzzeitiger Spannungsabfall, Spannungsspitze oder Kurzschluss.

O.ERR_19

 SyncPos

Benutzer Abbruch

Ursache

Das Autokennungs-Programm (Autostart) wurde durch den Benutzer abgebrochen. Oder es wurde während des Einschaltens die Taste [CANCEL] gedrückt und ein Master Reset ausgelöst.

O.ERR_25

 SyncPos

Achse 1 Endschalter erreicht

Ursache

Durch einen Bewegungsbefehl wurde der Endschalter einer Achse überfahren.

Hinweis

Durch das Überfahren eines Endschalters wird die Lageregelung (abhängig vom Parameter ENDSWMOD (44)) automatisch abgeschaltet und der Antrieb muss manuell aus der Endschalterstellung bewegt werden, bevor die Fehlermeldung zurückgesetzt werden kann.

O.ERR_49

 SyncPos

Zu viele Interruptfunktionen

Ursache

Es wurden mehr als die maximal möglichen Interruptfunktionen benutzt ...

Hinweis

...erlaubt sind:

- 32 ON INT
- 32 ON STATBIT
- 32 ON COMBIT
- 10 ON PARAM
- 20 ON APOS, ON MAPOS, ON MCPOS

O.ERR_51

 SyncPos

Zu viele verschachtelte Unterprogramme

Ursache

Im ausgeführten Programm wurde zu häufig von einem Unterprogramm direkt in das nächste Unterprogramm gesprungen.

Der Fehler tritt meist dann auf, wenn man rekursiv im Unterprogramm auf eines der Unterprogramme verweist.

Hinweis

Vermeiden Sie zu viele (maximal 10!) gegenseitige Unterprogrammaufrufe und Unterprogramme, die sich selbst aufrufen (Rekursive Unterprogrammaufrufe).

O.ERR_52

 SyncPos

Zu viele Return

Ursache

Im ausgeführten Programm sind entweder mehr RETURN als entsprechende GOSUB Befehle vorhanden oder es wurde mit einem GOTO Befehl direkt in ein Unterprogramm gesprungen.

Pro Unterprogramm ist nur ein RETURN erlaubt. Es ist immer besser, an den Anfang des Unterprogrammes zu springen und dann mit IF... nach einem vorher definierten Label zu springen.

O.ERR_62

 SyncPos

Fehler beim Verifizieren

Ursache

Nach einem Speichervorgang ins EEPROM (Programm oder Parameter) wurde beim Verifizieren ein Fehler festgestellt.

Hinweis

Löschen Sie das EEPROM mit "STEUERUNG" → "SPEICHER" → "EEPROM LÖSCHEN" und versuchen Sie noch einmal das Programm oder die Parameter zu speichern. Wenn es nicht gelingt, wenden Sie sich bitte an den Service.

O.ERR_70

 SyncPos

Fehler in DIM Anweisung

Bedeutung

Eine Array-Definition in einer DIM Anweisung stimmt nicht mit den bereits existierenden Arrays in der SyncPos-Option überein.

Ursache

Die Felder stammen von älteren SyncPos-Programmen. Das aktuelle Programm hat andere Definitionen.

Hinweis

Passen Sie entweder das SyncPos-Programm an die richtige Array-Größe an, löschen Sie die alten Arrays mit "STEUERUNG" → "SPEICHER" → "EEPROM LÖSCHEN", oder benutzen Sie den neuen Befehl "STEUERUNG" → "RESET" → "ARRAYS".


ACHTUNG!

Beachten Sie aber die Ratschläge zur Sicherung der Programme und Parameter, bevor Sie das EEPROM löschen.

O.ERR_71

 SyncPos

Feldgrenzen über- oder unterschritten

Bedeutung

Es wurde versucht ein Array-Element zu beschreiben, welches außerhalb der definierten Array-Grenzen liegt.

Ursache

Fehler im SyncPos-Programm. Die Array-Dimensionierung stimmt mit dem benötigtem Platz nicht überein (z.B. durch eine falsch programmierte Schleife). Oder das Array ist für die Anzahl der mit TESTSTART ausgelösten Testfahrten zu klein.

O.ERR_79

 SyncPos

Timeout bei Warten auf Index

Bedeutung

Der Befehl WAITNDX wurde ausgeführt und der dort angegebene Timeout überschritten.

Ursache

Vermutlich ist der Timeout zu kurz gesetzt oder der Indeximpuls konnte nicht gefunden werden (siehe auch O.ERR_9).

O.ERR_84

 SyncPos

Zu viele Time Interrupts

Bedeutung

Innerhalb des Programmes sind zu viele ON TIME oder ON PERIOD Befehle benutzt worden.

Hinweis

Es sind max. 12 dieser ON TIME und/oder ON PERIOD Befehle innerhalb eines Programmes erlaubt.

O.ERR_87

 SyncPos

Kein Platz mehr für Variablen

Bedeutung

Beim Start eines SyncPos-Programmes wird dynamisch der Platz für die benötigten Variablen reserviert. Dieser Platz ist jetzt nicht mehr vorhanden.

Ursache und Hinweise

Evtl. haben Sie die maximale Anzahl der Variablen zu groß gewählt. Reduzieren Sie die maximale Anzahl in "EINSTELLUNGEN" → "COMPILER" (Standard = 92).

Oder der verfügbare Speicher ist mit Programmen oder Arrays belegt. Löschen Sie die Programme mit "STEUERUNG" → "PROGRAMME" → "ALLE LÖSCHEN".

Oder löschen Sie auch die Arrays, d.h. den gesamten Speicher mit "STEUERUNG" → "SPEICHER" → "EEPROM LÖSCHEN".


ACHTUNG!

Beachten Sie aber die Ratschläge zur Sicherung der Programme und Parameter, bevor Sie das EEPROM löschen.

O.ERR_90

 SyncPos

MEMORY locked

Bedeutung

Der Programmspeicher ist schreibgeschützt und kann nicht verändert werden.

Hinweise

Sie können also Autokennung weder setzen noch löschen und keine Programme sichern oder löschen. Ebenso werden → "RAM SPEICHERN" und → "EEPROM LÖSCHEN" nicht ausgeführt.

O.ERR_91

 SyncPos

Kurven-Array in DIM-Anweisung falsch

Bedeutung

In der DIM-Anweisung für SETCURVE wird ein falsches oder altes Array definiert.

Hinweise

Prüfen Sie, ob die Reihenfolge der Arrays in der DIM-Anweisung mit der Reihenfolge in der cnf-Datei übereinstimmt. Sehen Sie dazu auch die Nummer des Arrays in der Titelleiste im CAM-Editor.

Ein altes Array kann existieren, wenn man die cnf-Datei mit allen Parametern und Arrays nicht in den CAM-Editor geladen hat.

Ursachen eines falschen Arrays können sein:

- nicht vom Kurvendedor erstellt
- frühere Version eines Kurveditors. Ein solches Array muss erst durch den aktuellen Kurvendedor konvertiert werden (→ laden und neu speichern).

O.ERR_92

 SyncPos

Fehler aus Encoder Überwachung

Bedeutung

Die mit Parameter ENCODERTYPE und/oder MENCODERTYPE eingestellte Drehgeber-Überwachung meldet einen Fehler, wenn die Encoder-Leitungen unterbrochen sind und bei Kurzschluss. Welche Leitung betroffen ist, erkennen Sie an der entsprechenden LED.


ACHTUNG!

Es wird auch ein Fehler gemeldet, wenn kein Encoder angeschlossen ist. Schalten Sie dann besser die Encoder-Überwachung aus.

O.ERR_xx

SyncPos

Interner Fehler ##

Bedeutung

Sollte solch ein Fehler auftreten, setzen Sie sich bitte mit Ihrem Händler in Verbindung und nennen dem Service die dazu angezeigte Fehlernummer.

■ Meldungen von der SyncPos PC-Software

Die Meldungen von der SyncPos PC-Software sind alphabetisch geordnet. Buchstaben hinter einem %-Zeichen stehen für variable Werte, die im Klartext an den entsprechenden Stellen eingesetzt werden.

■ Fehler beim Compilieren ...

SyncPos

Fehler beim Compilieren: Programm nicht gespeichert!

Bedeutung und Hinweis

Eine Datei wird immer erst compiliert und dann gespeichert. Wenn Sie das Programm speichern wollen, zum Beispiel im Menü "STEUERUNG" → "PROGRAMM SICHERN" und beim Compilieren ein Syntaxfehler festgestellt wird, erhalten Sie diese Meldung.

Starten Sie die "SYNTAXPRÜFUNG" im Menü "ENTWICKLUNG", beheben Sie den Syntaxfehler und speichern Sie dann das Programm.

■ Fehler in Datei: Achsparameter

SyncPos

Fehler in Datei: Achsparameter

Bedeutung

Beim Zurücksichern einer Konfiguration ("STEUERUNG" → "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI") wurde erkannt, dass die Daten im Bereich der Achsparameter ein nicht korrektes Format aufweisen. Die Anzahl und Reihenfolge der Parameter muss korrekt sein und die Nummerierung darf keine Lücken aufweisen.

Ursachen

Um eine Datei zurückspeichern zu können, müssen folgende Bedingungen erfüllt sein:

- Identische Softwareversionen und damit gleiche Anzahl und Reihenfolge der Parameter
- Gleiche Konfiguration (z.B. gleiche Anzahl der Achsen)

■ Fehler in Datei: Array ...

SyncPos

Fehler in Datei: Array-Daten

Bedeutung

Beim Zurücksichern einer Konfiguration ("STEUERUNG" → "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI") wurde erkannt, dass die Daten im Bereich der Arrays ein nicht korrektes Format aufweisen.

Ursachen

Um eine Datei zurückspeichern zu können, müssen folgende Bedingungen erfüllt sein:

- Identische Software-Versionen
- Gleiche Konfiguration (z.B. gleiche Anzahl der Achsen)
- Falls bereits Arrays angelegt sind, müssen diese in Art und Größe zu denen passen, die zurückgespeichert werden sollen.

■ Fehler in Datei: Globale Parameter

SyncPos

Fehler in Datei: Globale Parameter.

Bedeutung

Beim Zurücksichern einer Konfiguration ("STEUERUNG" → "PARAMETER" → "WIEDERHERSTELLEN AUS DATEI") wurde erkannt, dass die Daten im Bereich globale Parameter ein nicht korrektes Format aufweisen.

Ursachen

Um eine Datei zurückspeichern zu können, müssen folgende Bedingungen erfüllt sein:

- Identische Software-Versionen und damit gleiche Anzahl und Reihenfolge der Parameter
- Gleiche Konfiguration (z.B. gleiche Anzahl der Achsen)

■ Steuerung führt ein Programm ... aus

SyncPos

Steuerung führt ein Programm oder Kommando aus!

Bedeutung und Hinweis

Während die Steuerung ein Kommando oder Programm ausführt, steht sie nicht für weitere Kommandos zur Verfügung. Sie müssen den neuen Befehl → "ABBRECHEN" und erneut starten, wenn das vorhergehende Kommando zu Ende ausgeführt ist.

Kapitel 8**■ Programmbeispiele**

Einleitung und Sicherheitshinweise	224
COM_OPT	225
Marker count.....	225
Master Simulation per Software	226

■ Einleitung**SICHERHEITSHINWEISE**

Die gesamte Einheit muss gemäß den Installationsanweisungen installiert und die Funktionsfähigkeit der einzelnen Komponenten und Verbindungen muss getestet sein.

Ein NOT-STOP muss gemäß den länderspezifischen Richtlinien installiert sein.

Die Programmbeispiele dürfen nur mit einem Antrieb, der ohne Wegbegrenzung frei drehen kann, benutzt werden.

Der Antrieb sollte in einer stabilen Halterung fixiert sein.

Bei der Verwendung einer Lineareinheit müssen Endschalter zur Wegbegrenzung montiert und angeschlossen sein.

Bei der Verwendung einer Lineareinheit sollte der Verfahrbereich der Programmbeispiele unbedingt überschlägig berechnet und mit den mechanisch möglichen Verfahrwegen verglichen werden.

Verweise auf die Programmbeispiele in der Online-Hilfe

In der Online-Hilfe finden Sie alle anderen Programmbeispiele, auf die in der Software-Referenz verwiesen wird. Sie können diese Programme oder Programmteile kopieren und in Ihr Programm einfügen.

■ COM_OPT

```

/* Programm zum Senden und Empfangen von */
/* 8-Byte-Datenworten via */
/* Kommunikationsoption und PPO type 2 */

/* Arrays definieren */
DIM send [4]
DIM receive [4]

/* Benutzerparameter definieren */
LINKGPARG 133 710 "DATA WORD 1" 0 255 0
LINKGPARG 134 711 "DATA WORD 2" 0 255 0
LINKGPARG 135 712 "DATA WORD 3" 0 255 0
LINKGPARG 136 713 "DATA WORD 4" 0 255 0

/* Array initialisieren (alle Elemente = 0) */
i = 1
WHILE (i<=4) DO
    receive [i] = 0
    i = i+1
ENDWHILE
j = 1
WHILE (j<=4) DO
    send [j] = 0
    j = j+1
ENDWHILE

/* Hauptprogrammschleife */
main:
send [1] = GET 133
    /* send array, element 1 = Parameterwert 710 */
send [2] = GET 134
    /* send array, element 2 = Parameterwert 711 */
send [3] = GET 135
    /* send array, element 3 = Parameterwert 712 */
send [4] = GET 136
    /* send array, element 4 = Parameterwert 713 */

COMOPTGET 4 receive
    /* Kopiere 4 Datenwörter von der */
    /* Kommunikationsoption in das */
    /* Empfänger-Array */
COMOPTSEND 4 send
    /* Kopiere 4 Datenwörter vom Sende-Array */
    /* in die Kommunikationsoption */

/* Empfangene Daten */
print "RECEIVED(4 WORDS)," "receive [1],"
"receive [2]," "receive [3]," "receive [4]
GOTO main
    /* Programmende */

```

■ Marker count

```

/* Programm zum Messen des Markerabstandes */
/* in Verbindung mit dem Befehl SYNCM */
/* Master und Slave müssen laufen, wenn dieses */
/* Programm ausgeführt wird. */

DIM test[4]
    /* Arrays mit 4 Elementen definieren */

    /* Variablen definieren */
dones = 0
donem = 0

    /* Letzte Markerposition des Slave und */
    /* Masters speichern */
test [1] = IPOS
test [2] = MIPOS

    /* Hauptprogrammschleife */
main:
    /* Nächsten Slavemarkers erfassen und seine */
    /* Position speichern */
IF (IPOS != test [1] AND dones == 0) THEN
    test [3] = IPOS
    dones = 1
    res = test [3] - test [1]
    /* Abstand zwischen 2 aufeinanderfolgenden */
    /* Slavemarkern berechnen */
    PRINT "Slave Abstand ", res /* Print result */
ENDIF

    /* Nächsten Mastermarker erfassen und seine */
    /* Position speichern */
IF (MIPOS != test [2] AND donem == 0) THEN
    test [4] = MIPOS
    donem = 1
    res = test [4] - test [2]
    /* Abstand zwischen 2 aufeinanderfolgenden */
    /* Mastermarkern berechnen */
    PRINT "Master Abstand ", res
    /* Print result */
ENDIF
GOTO main

```

■ syncc_msim.m

```
// Simulation eines Masters per Software-Befehl
DIM curve[112], test [1000]
SET MENCODERTYPE 6
  // auf interne Software Simulation umschalten
SETCURVE curve
simpos = 0
SYNCC 0
SYNCCSTART 0
anf:
  SYSVAR[4105] = simpos
  // setzt Master-Position (in qc)
  // anschließend wird Getriebe angewendet
  simpos = simpos + 100
  // ACHTUNG, MAPOS liefert nach wie vor
  // die Encoder-Werte
GOTO anf
```


Kapitel 9

■ Anhang	
■ Glossar	228
■ Neues in der PC-Software ...	
2.1x / Optionskarten Software 3.1x	229
2.2x und in der SP Motion Controller Software	231
Neue Hardware	232
Neues in der PC-Software ...	
2.3x und in der SP Motion Controller Software	233
Neues in der PC-Software ...	
2.5x und in der Optionskarte Version 5.00	233
Neues in der PC-Software ...	
6.5x und in der Optionskarte Version 5.04	234
■ Technische Referenz	235
■ Stichwortverzeichnis	239

■ Glossar
Benutzereinheit [BE]

Wegangaben in Fahrbefehlen erfolgen immer in Benutzereinheiten und werden intern in Quadcounts umgerechnet. Diese wirken sich auf alle Befehle für das Positionieren aus: z.B. APOS, POS.

So ist es durch eine entsprechende Wahl dieser Normierungsgröße möglich, mit beliebigen technischen Maßangaben (zum Beispiel in mm) zu arbeiten.

Bei der Kurvenscheibensteuerung wird der maximale Fahrabstand des Slaves bzw. die Slave-Zykluslänge in Benutzereinheiten BE (qc) angegeben.

Der Faktor ist ein Bruch, der sich aus Zähler und Nenner zusammensetzt:

$$1 \text{ BE} = \text{POSFAC}_Z (23) / \text{POSFAC}_N (26)$$

FPGA

Field Programmable Gate Array

LED

Light Emitting Diode (Leuchtdiode)

LSB und MSB

LSB = least significant Bit = niederwertigstes signifikante Bit;

MSB = most significant Bit = höchstwertigstes signifikante Bit

Master Units [MU]

Die Kurvenlänge bzw. Master-Zykluslänge und andere Angaben (zum Beispiel der Markerabstand) für die Kurvenscheibensteuerung werden in Master Units MU angegeben.

Für die Umrechnung in qc wird wie bei der Benutzereinheit ein Faktor (Bruch) benutzt:

$$1 \text{ MU} = \text{SYNCFAC}_M (49) / \text{SYNCFAC}_S (50)$$

MLONG

Für eine versions- und hardware-unabhängige Beschreibung wurden für einige Grenzwerte Variablen benutzt: so beträgt der Wert von

$$\text{MLONG} = 1.073.741.824$$

MTC10

MotionControlTool

Mt

Mastertaktlänge

negierter Wert

nichtA = /A = A quer = \bar{A}

Nullimpuls

= Indeximpuls

Quadcounts

Die durch den Parameter Benutzerfaktor in Parameter POSFACT_Z (23) und POSFACT_N (26) definierte Einheit BE wird nicht bei allen Weg-/Positions-Parametern verwendet.

Für verschiedene Parameter, zum Beispiel I_POSLIMITSW und **Tolerierter Positionsfehler** POSERR (15) gilt die Einheit Quadcount (qc).

4 Quadcounts entsprechen (bei Inkrementalgebern) einer Gebereinheit. Bei Absolut-Drehgebern werden die absoluten Werte 1 : 1 zurückgegeben.

Schleppabstand

Der "Schleppabstand Achse" ist der Positionsfehler. Im Parameter POSERR (15) wird der maximal **tolerierete Positionsfehler** festgelegt. Wird dieser überschritten, erfolgt die Meldung "Schleppabstand überschritten" oder "Schleppfehler".

st

sample time = 1 ms

■ Neues in der PC Software 2.1x und Optionskarten-Software 3.1x
■ Erweiterung der Benutzeroberfläche
Mehr Komfort im Editierfenster

Verschiedene Farben erleichtern Ihnen die Unterscheidung zwischen Kommentaren, Programmteilen, Operatoren, Ziffern usw. Sie können die Farbuordnung mit "EINSTELLUNGEN" → "FARBEN EDITOR" ändern.

Außerdem können Sie das Fenster für die Darstellung von mehreren Dateien nun auch vertikal oder horizontal teilen.

Weitere Eingabehilfen und Shortcuts

Es gibt neue Editierkommandos, Tastenkombinationen und Funktionstasten. Darüber hinaus sind folgende Editierhilfen hinzu gekommen: Unbegrenzt rückgängig und Wiederherstellen, die Funktion Suchen und Ersetzen und die Möglichkeit Makros aufzuzeichnen.

Wenn Sie während des Einschaltens die Taste CANCEL am VLT drücken, wird die SyncPos-Optionskarte kein Programm starten.

■ Erweiterung des Compilers, der Befehle und Parameter
Erweiterung des Compilers

Kennzeichnung des Kommentars zusätzlich auch mit // analog der Programmiersprache C.

Weitere Operatoren, wie XOR und Modulo und weitere Bit-Operatoren.

Zuweisung (Assignment Operation)

Zusätzliche Interrupts

ON COMBIT	Interrupt wenn Bit n gesetzt wird
ON STATBIT	Interrupt wenn Bit n gesetzt wird
ON PARAM	Interrupt wenn sich ein Parameter n ändert

■ Neue Befehle

DEFMORIGIN	Aktuelle Master-Position als Nullpunkt für den Master setzen
DELETE ARRAYS	Alle Arrays im RAM löschen.
DISABLE .. inttyp	Interrupt ... ausschalten
DISABLE ALL	Alle Interrupts (außer ON..ERROR) ausschalten.
ENABLE ...	Interrupt ... einschalten
ENABLE ALL	Alle Interrupts einschalten
LINKSYSVAR	Systemvariable mit LCP-Display verknüpfen
MOVESYNCORIGIN	Synchronisationsursprung relativ verschieben
ON COMBIT .. GOSUB	Unterprogramm aufrufen, wenn Bit n des Kommunikationspuffers gesetzt ist
ON PARAM .. GOSUB	Unterprogramm aufrufen, wenn sich ein Parameter ändert
ON STATBIT	Unterprogramm aufrufen, wenn Bit n des VLT-Status gesetzt ist.
PCD	Pseudo-Array für den direkten Zugriff auf den Profibus-Datenbereich
SAVE ARRAYS	Arrays im EEPROM sichern
SAVE AXPARS	Aktuelle Achsenparameter im EEPROM sichern
SAVE GLBPARS	Aktuelle globale Parameter im EEPROM sichern
SETMORIGIN	Beliebige Position als Nullpunkt für den Master setzen
SYNCSTATCLR	Zurücksetzen der Flags MERR und MHIT
SYSVAR	Systemvariable (Pseudo-Array) liest Systemwerte
TESTSETP	Aufzeichnungsdaten für Testfahrt festlegen
TESTSTART	Aufzeichnung der Testfahrt starten

■ Befehle, die erweitert wurdenSYNCERR

Aus dem Rückgabewert lässt sich jetzt auch ablesen, ob die Synchronisation voraus- oder nachläuft (negatives bzw. positives Ergebnis), wenn SYNCACCURACY mit negativem Vorzeichen definiert ist.

ERRCLR

ERRCLR setzt nun auch Bit 7 des Steuerwortes vom VLT zurück. Die VLT-Meldungen müssen dann nicht mehr gelöscht werden.

LINKGPAR

Der Befehl LINKGPAR prüft, ob der Wert des Benutzerparameters innerhalb des vorgegebenen Bereiches liegt. Falls nicht, wird das entsprechende Limit verwendet und dieser Wert gespeichert. Auf diese Weise wird sichergestellt, dass im Display eine Anzeige erscheint.

SYNCM

Wenn in SYNCMSTART (62) definiert, wird beim Start von SYNCM auf die erste Auswertung der Markerpulse gewartet und erst dann der Offset SYNCPOSOFFS (54) angewandt.

#DEBUG

Zusätzlich zu ON und OFF können Sie NOSTOP definieren. Dann wird nicht angehalten, aber interne Break-Kommandos aktualisieren die Zeilennummer wenn man das Programm mit "ENTWICKLUNG" → "EINZELSCHRITT" F9 schrittweise ausführt.

Die so erzeugten Zeilennummern können dann mit LINKSYSVAR auf einen VLT-Parameter 795-799 gelegt und am LCP-Display beobachtet werden.

■ Neue ParameterESCCOND (70)

Mit diesem Parameter legen Sie fest, wie der VLT bei Programmabbruch durch ESC reagieren soll.

SYNCMWINM (68) / SYNCMWINS (69)

Die Marker Window Master und Slave geben an, wie groß die erlaubte Toleranz für das Auftreten der Marker ist.

■ Parameter, die erweitert wurdenPOSDRCT (28)

Durch die Eingabe von „2“ und „-2“ kann man die Motorleitungen bzw. die A- und B-Spur beim Drehgeber per Definition „tauschen“, also ohne die Verkabelung tatsächlich zu ändern.

O_BRAKE (48)

Wenn ein Ausgang für die Bremse definiert ist, bleibt dieser auch dann aktiviert, wenn das Programm mit ESC abgebrochen wird.

SYNCPOSOFS (54)

Beim Start von SYNCM – im Gegensatz zu SYNCP – wird auf die erste Auswertung der Markerpulse gewartet. Erst dann wird der Offset angewandt.

SYNCCACCURY (55)

Erweiterter Eingabebereich: Negatives Vorzeichen liefert den Synchronisationsfehler an SYNCERR mit Vorzeichen. Daraus lässt sich dann erkennen, ob die Synchronisation voraus- oder nachläuft.

SYNCMSTART (62)

Erweiterung der Möglichkeiten, um das Startverhalten für die Markersynchronisation festzulegen.

SYNCFVTIME (65)

Masterfilter für SYNCP und SYNCM

■ **Neues in der PC-Software 2.2x und in der entsprechenden SyncPos Motion Controller Software:**

VLT5000/SyncPos Software Version 3.xx/4.1x
 VLT5000FluxSyncPos Software Version 5.xx/4.1x

■ **Kurveneditor**

Diese PC-Software-Version enthält einen Kurveneditor für die Erstellung der Kurven für Kurvenscheibensteuerungen mit und ohne Markerkorrektur des Masters und des Slaves. Im Kurveneditor werden auch alle für diese Anwendung notwendigen Parameter gesetzt.

Die Kurven können sowohl mit alphanumerischen Eingaben wie auch interaktiv mit der Maus definiert werden. Kurvenverlauf und Parameter können im CAM Profil visualisiert werden.

■ **Erweiterung der Befehle und Parameter**

Neue Interrupts

Die neuen Positionierungs-Interrupts POSINT wurde besonders für die Nockenschaltwerk-Funktionen entwickelt: ON APOS, ON MAPOS und ON MCPOS.

■ **Neue Befehle**

Die neuen Befehle wurden für die Realisierung von Kurvenscheibensteuerungen und Nockenschaltwerke entwickelt:

CURVEPOS	Slave-Position, die der aktuellen Master-Position der Kurve entspricht, abfragen.
DEFMCPOS	Anfangsposition des Masters definieren
ON APOS .. GOSUB	Unterprogramm aufrufen, wenn die Slave-Position xxx passiert wurde
ON MAPOS .. GOSUB	Unterprogramm aufrufen, wenn die Master-Position xxx (qc) passiert wurde
ON MCPOS .. GOSUB	Unterprogramm aufrufen, wenn die Master-Position xxx (MU) passiert wurde
POSA CURVEPOS	Slave auf die, der Master-Position entsprechenden Kurvenposition fahren.
SETCURVE	CAM-Kurve setzen
SYNCC	Synchronisation im CAM-Mode
SYNCCMM	Synchronisation im CAM-Mode mit Markerkorrektur des Masters

SYNCCMS	Synchronisation im CAM-Mode mit Markerkorrektur des Slaves
SYNCCSTART	Slave zur Synchronisation im CAM-Mode starten
SYNCCSTOP	Slave nach der CAM-Synchronisation anhalten
SWAPMENC	Master- und Slave-Drehgeber intern tauschen

■ **Modifizierte und erweiterte Befehle**

SET ORIGIN	In Verbindung mit dem Befehl CURVEPOS kann man festlegen, dass die aktuelle Slave-Position mit dem entsprechenden Wert der Kurve übereinstimmt.
SYNCSTAT	Die Flags SYNCCMERR und SYNCSMERR werden jetzt automatisch zurückgesetzt: Bei der nächsten erfolgreichen Markerkorrektur und bei erneutem Start von SYNCM oder durch den Befehl SYNCSTATCLR.
SYSVAR	Ergänzung mit den Achsprozessdaten für das Kurvenprofil.

■ **Neue Parameter**

PROTIME - 29
 Abtastzeit für Profilgenerator

■ **Modifizierte und erweiterte Parameter**

Parameter 700
 Als dritte Einstellung kann man "ENABLE SP W/O MONI" wählen, also SyncPOS ohne VLT-Überwachung aktivieren und so den Fehler 13 (VLT nicht bereit) vermeiden.

ENCODERTYPE (27)
 Bei absoluten Drehgebern ohne Überlauf (linearen Drehgebern) kann die Störung korrigiert werden. Zusätzlich kann man für alle Encodertypen die Encoder-Überwachung ein- und ausschalten.

POSFAC_Z (23) und POSFAC_N (26)
 Wird im CAM-Mode zur Umrechnung der Einheiten qc in BE benutzt.

SYNCFAC_M (49) und SYNCFAC_M (50)
 Wird im CAM-Mode zur Umrechnung der Einheiten qc in MU benutzt.

SYNCPULSM (58)

Im CAM-Mode Abstand zwischen Sensor und Arbeitsposition in MU.

SYNCPULSS (59)

Im CAM-Mode Abstand zwischen Sensor und Arbeitsposition in BE.

SYNCTYPEM (60) und SYNCTYPES (61)

Die Einstellung des Signal- bzw. Markertyps gilt auch für den Befehl MIPOS.

SYNCMSTART (62)

Weiterer Parameter für das Startverhalten:
2000 = im CAM-Mode: Das Zählen (Beobachten) der Masterpulse in CM beginnt mit dem Mastermarker.

MENCODERTYPE (67)

Bei absoluten Drehgebern ohne Überlauf (linearen Drehgebern) kann die Störung korrigiert werden. Mit dem Encodertyp „6“ lässt sich ein Master per SyncPos-Befehl simulieren. Und zusätzlich kann man für alle Encodertypen die Encoder-Überwachung ein- und ausschalten.

SYNCMWINM (68) und SYNCMWINS (69)

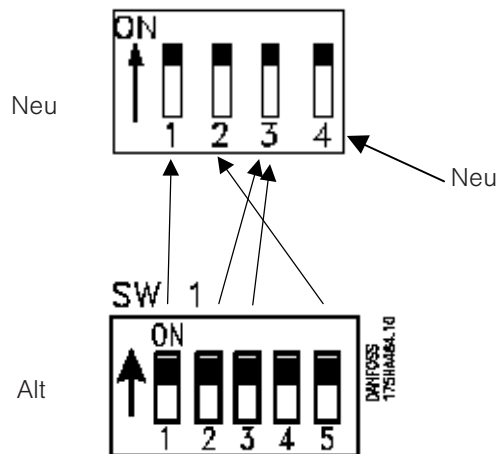
Änderungen des Parameters werden sofort aktiv – nicht erst nach dem nächsten SYNCM-Befehl.

■ Neue Hardware

Die Hardware des Programmierbaren SyncPos Motion Controllers wurde aktualisiert:

- Der Programmspeicher wurde von 38 kByte auf 62 kByte erweitert.
- Hardware Drehgeberüberwachung.
- Neue LED's.

Der DIP-Schalter ist verändert: Die Schalter 2, 3 und 4 wurden in der neuen Hardware zu einem Schalter (3) zusammengefasst:



SyncPos für VLT5000Flux basiert grundsätzlich auf der neuen Hardware.

Ob SyncPos für VLT5000 die neue Hardware unterstützt, erkennt man an der Seriennummer auf dem Barcode-Label. Seriennummer xxxxxxGxxx kennzeichnet die neue Hardware und Seriennummer xxxxxxNxxx die alte Hardware.

Software Kompatibilität

Die neue Hardware wird von der Optionskarten-Software ab 4.xx unterstützt.

Die alte Hardware wird von der Optionskarten-Software bis 3.12 unterstützt.

**■ Neues in der PC-Software 2.3x
und in der SyncPos Motion Controller Software**

VLT5000/SyncPos Software Version 3.xx/4.2x
VLT5000FluxSyncPos Software Version 5.xx/4.2x

■ Debug Modus

Komfortabler Debug-Modus und online Statusinformationen sowie die Möglichkeit, während der Programmausführung die Variablen zu ändern:

"VORBEREITEN EINZELSCHRITT"

Alle ausführbaren Programmzeilen werden markiert.

"EINZELSCHRITT"

Im Debug-Modus: Programmausführung stoppt vor jeder Programmzeile

Haltepunkte

Im Debug-Modus: Programmausführung stoppt vor jedem Haltepunkt.

Variablen online ändern

Im Debug-Modus Variablen setzen

Variablen auslesen

Im Debug-Modus nach der Programmausführung den aktuellen Wert der Variablen auslesen.

"Überwachung anzeigen"

ermöglicht die Online-Überwachung der Variablen, Arrays, System- und Achsprozessdaten (gemäß der SYSVAR Indizes) und Achsenparameter.

■ Suchen und Ersetzen

Realisierung gemäß den Windows-Konventionen plus

"ALLE MARKIEREN"

Alle Fundstellen markieren

F2

Zwischen den markierten Fundstellen hin- und herspringen

F3

Gleichen Begriff Weitersuchen

**■ Neues in der PC-Software 2.5x
und in der Optionskarten-Software Version 5.00**

Diese Version enthält vor allem die erforderlichen Erweiterungen und eine modifizierte Benutzeroberfläche für den Einsatz des SyncPos-Programmes im MCT10-Modus.

Darüber hinaus gibt es mit dem modifizierten CAM-Editor eine zusätzliche Möglichkeit, die Achsparameter einzustellen oder zu ändern.

■ Neue und modifizierte BefehleDEF SYNCORIGIN

Definiert das Verhältnis von Master zu Slave für den nächsten SYNCP oder SYNCM Befehl

SYNCERR

SYNCERR (und auch die interne Master-Sollposition G_Mpcmd) wird nun auch aktualisiert, wenn SYNCP oder SYNCM nicht aktiv sind, z.B. nach einem MOTOR OFF.

■ Modifizierter ParameterPRGPAR - 102

Erweitert mit der Möglichkeit, die aktivierte Programmnummer zu starten, aber beim Hochfahren den Motor ausgeschaltet zu lassen (MOTOR OFF).

■ Neues in PC Software Version 6.5x und in der Optionskarten-Software Version 5.04

VLT5000/SyncPos Software Version 3.7X/5.0X
 VLT5000Flux/SyncPos Software Version 5.XX/5.0X

■ Benutzeroberfläche

Menü "ENTWICKLUNG"

Mit → "SCHNITTSTELLE SCHLIESSEN" kann eine zurzeit offene Motorcontroller-Schnittstelle geschlossen werden. Für den schnellen Zugriff wurde die Funktion auch in der Symbolleiste ergänzt.

Mit → "IN DATEI KOMPILIEREN" kann ein aktuelles File kompiliert und als binäres File gesichert werden.

Die Funktion "SYNTAXPRÜFUNG" erzeugt nun zusätzlich zum Check ein Debug-File und speichert es als „temp.ad\$“.

Menü "TESTFAHRT"

Es kann kein Testrun gestartet werden, wenn die Motorsteuerung Fehler meldet. Mit "FEHLER LÖSCHEN" im Dialogfeld Testfahrt-Parameter können nun alle aktuell anliegenden Fehlerbedingungen in der Motorsteuerung gelöscht werden.

Menü "STEUERUNG"

Bei → "SICHERN MIT QUELLCODE" wird der Quellcode mit allen dazugehörigen „Included“-Dateien ergänzt und in der Steuerung gespeichert.

Bislang konnte man → "PROGRAMM SICHERN" nicht auswählen, wenn ein Programm bereits aktiv war. Nun wird in einem solchen Fall ein Dialogfenster geöffnet, in dem das aktive Programm unterbrochen werden kann. Das neue Programm wird dann gespeichert.

CAM Editor

Wenn eine ältere Version eines ".cnf"-Files geladen wird (in dem nicht alle neuesten Parameter definiert sind), ergänzt der CAM Editor diese mit geeigneten Default-Werten.

Menü "EINSTELLUNGEN"

Nun kann für das VLT-Interface auch eine Baudrate von 19200 eingestellt werden.

■ Neue Befehle

MOVESYNCORIGIN

Synchronisationsursprung relativ verschieben

ON DELETE .. GOSUB

Löscht einen Positions-Interrupt: ON APOS, ON MAPOS oder ON MCPOS

■ Modifizierte und erweiterte Befehle

CURVEPOS

CMASTERCPOS (SYSVAR) und CURVEPOS werden nun auch aktualisiert, wenn SYNCC nicht mehr aktiv ist.

SETCURVE

Unterschiedliches Verhalten wenn SYNCC aktiv oder nicht aktiv ist.

SYSVAR

SYSVAR liefert wieder weitere Informationen für den Anwender: Ein neuer Systemprozessdaten-Index 35 für den ungefilterten Wert des analogen Eingangs 1 und einige weitere Achsprozessdaten (Indices 4113 ... 4128 und 4240 ... 4253).

■ Neue und modifizierte Parameter

SYNMMAXCORR (6)

Begrenzt die max. Korrektur der Markerkorrektur

SYNCOFFTIME (16)

Geschwindigkeitskompensation eines Offsets (1. Synchronisieren; 2. Neuer Offset)

SYNMFPAR (17)

Markerfilter Konfiguration

SYNMFMTIME (18)

Das Verhalten des Parameters SYNMFMTIME – Filterzeit für die Markerkorrektur – wurde etwas geändert.

SYNCTYPE (51)

Gibt an, ob normale Synchronisation (0) oder Synchronisation mit Look-ahead (1).

■ Technische Referenz

Dieser Abschnitt dokumentiert Datenstrukturen und Compilerdetails, die der Anwender nur in Ausnahmefällen benötigt. Zum Beispiel wenn eine automatisch erzeugte Programmierung, wie ein CAM-Profil verändert werden soll.

Da der Abschnitt nur für erfahrene Programmier vorgesehen ist, wird die Referenz nur in englisch dokumentiert.

■ Array Structure of CAM Profiles
Header

The header contains general information like

- Identification for curve array
- Version number for curve structure
- Type of curve
- Name of curve
- Index to curve information section
- Index to start/stop point section
- Index to fixed point section
- Index to interpolation point section
- Index to start/stop point indices (in interpolation section)
- Index to start/stop velocities (times 100000)
- Index to startpath interpolation points
- Index to stoppath interpolation points

Curve information section

This section of the array contains all information about the type of curve like

- Length of curve (master)
- Length of curve (slave)
- Number of fix points
- Number of Interpolation points (this gives the resolution)
- Type of interpolation
- Slave stop point, point where slave is positioned, when synchronization is stopped
- Correction start point (only valid for marker synchronization)
- Correction end point (only valid for marker synchronization)
- Maximum correction which is allowed (only valid for marker synchronization)
- Maximum start/stop path length (Size of start/stop path area)(min. 2)
- No of start/stop point pairs
- Maximum number of cycles per minute (Application information)

Curve start/stop point section

This section contains the start/stop points. Because the use of this point is up to the user, we just speak of a path, which can be a start or a stop sequence. Every path consists of 2 points. If we are moving forward, the path starts (start or stop) with the a-point and ends with the b-point. If we are moving backward, the path starts with the b-point and ends with the a-point. So the user is able to tell us in the program, which pair of points to use for starting or stopping, when he uses a STARTCURVE or STOPCURVE command.

- Path 1 (a – point)
- Path 1 (b – point)
- Path 2 (a – point)
- Path 2 (b – point)
- ...

These points have to lie on interpolation points, so possibly the PC software has to adjust them according to the interpolation resolution. This should not be a real restriction, because the interpolation points are normally very dense. So for example if we have rotating master which makes one revolution per cycle and we choose a cycle length of 3600 MU (1 MU = 1/10 degree). Let us further assume, that we choose the number of interpolation points as 1200, than you have a resolution of 3 MU = 3/10 degree for defining your start and stop points.

Fixed point section

This section contains the fix points, which were the basis for the interpolation calculation. These points always consist of the following triple

- Master coordinate
- Slave coordinate
- Type of point (tangent, curve)

These points are defined by the user in MU units (see internal description). If you want to avoid, that the real interpolation curve misses your fix points, you have to choose them in such a manner that they lay on an interpolation point (see above). This can be forced through a snap function within the PC software.

Interpolation point section

This section contains a list of slave coordinates. They belong to master coordinates which are of equal distance, given by the interpolation resolution.

Indices of start/stop points

Here we have the indices of the start/stop points (see above) within the interpolation array. These are necessary for the ease of start and stop recognition. We are waiting until start index for example equals the actual index and direction of movement is correct. If both is true, we start synchronization. The same is true for stopping.

Start Stop Velocities

To be able to calculate an appropriate starting or stopping path, we need the velocity we have to reach at end (start) or we will have at the beginning (stop) in UU/MU units (Slave units per Master units).

Start / Stop paths

This is the place for the interpolation points of the actual start and stop path. These points are calculated when a SYNCCSTART or SYNCCSTOP command is executed, but we have to reserve the room right now.

■ CAM Array Definition

Part	Index	Name	Unit	Value	Description
General	1	Identification	(dec)	999.000.001	Number to identify array
	2	VersioNumber	(dec)	100	Version as decimal (1.00 = 100)
	3	CurveType	(dec)	0	0 = symmetrical, 1 = compatible
	4	CurveName 1	(4char)	Nona	Name of curve total 16 char.
	5	CurveName 2	(4char)	meCu	default is:
	6	CurveName 3	(4char)	rve0	NonameCurve00001
	7	CurveName 4	(4char)	0001	
	8	IndexCIF	(dec)	16	Index to Curve Information Part
	9	IndexSTP	(dec)	27	Index to Start/Stop Point Part
	10	IndexFIP	(dec)	IndexSTP + STPno*2	Index to Fix Point Part
	11	IndexINP	(dec)	IndexFIP + FixPointNo * 3	Index to Interpolation Point Part
	12	IndexSTPInd	(dec)	IndexINP + InterpolPointNo	Index to StartStop Interpolation Indices
	13	IndexSTPVel	(dec)	IndexSTPInd + STPno*2	Index to StartStop Velocities
	14	IndexSTIP	(dec)	IndexSTPVel + STPno*2	Index to Startpath interpolation points
	15	IndexSTPIP	(dec)	IndexSTIP + MaxStartStopLen	Index to Stoppath interpolation points
Curve Information	1	MasterCycleLen	MU	-	Length of Curve in Curve Master units
	2	SlaveCycleLen	UU	-	Slave max. travel distance in CurveSlave units
	3	FixPointNo	(dec)	4	Number of fix points (min. 4)
	4	InterpolPointNo	(dec)	-	Number of interpolation points (including first and last, which correspond to the same location)
	5	InterpolType	(dec)	0	0 = cubic spline 1 = periodic cubic spline
	6	SlaveStopPosition	UU	0	Position, where slave stands after stopping
	7	CorrectionStartPoint	MU	0	Position, where Correction may start
	8	CorrectionStopPoint	MU	MasterCycleLen	Position, where Correction has to be finished
	9	MaximumCorrection	UU	-	Maximum Correction which is allowed in one cycle

Programmierbarer SyncPos Motion Controller

Part	Index	Name	Unit	Value	Description
	10	MaxStartStopLen	(dec)	0	Maximum length of start/stop path (no of int. points)
	11	StartStopNo	(dec)	0	Number of start stop point pairs (n) (see below)
Start/Stop Point	1	STPoint_1.a	MU	0	Start (forward) / Stop (backward) Point no. 1
	2	STPoint_1.b	MU	0	Stop (forward) / Start (backward) Point no. 1
	3	STPoint_2.a	MU	0	Start (forward) / Stop (backward) Point no. 2
	4	STPoint_2.b	MU	0	Stop (forward) / Start (backward) Point no. 2
	5	...	MU	0	
	6	...	MU	0	
	2*n-1	STPoint_n.a	MU	0	Start (forward) / Stop (backward) Point no. n
	2*n	STPoint_n.b	MU	0	Stop (forward) / Start (backward) Point no. n
Fix Point	1	FixPoint_1.master	MU	0	Fix Point no. 1 - master coordinate
	2	FixPoint_1.slave	UU	-	Fix Point no. 1 - slave coordinate
	3	FixPoint_1.type	(dec)	C	Fix Point no. 1 - type of point (1 = Curve Point, 2= Tangent Point)
	4	...			
	3*n-2	FixPoint_n.master	MU	MasterCycleLen	Fix Point no. n - master coordinate
	3*n-1	FixPoint_n.slave	UU	-	Fix Point no. n - slave coordinate
	3*n	FixPoint_n.type	(dec)	C	Fix Point no. n - type of point (1 = Curve Point, 2 = Tangent Point)
Interpolation Point	1	IntPoint_1	UU	0	Interpolation Point no. 1 - slave coordinate
	...				
	n	IntPoint_n	UU	-	Interpolation Point no. n - slave coordinate
StartStop Indices	1	STPoint_1.a-index	(dec)	0	Index in Interpolation Array, corresponding to Startpoint
	2	STPoint_1.b-index	(dec)	0	Index in Interpolation Array, corresponding to Sartpoint
	3	..			

Programmierbarer SyncPos Motion Controller

Part	Index	Name	Unit	Value	Description
StartStop Velocities	1	STPoint_1.a-veloc.	(dec)	(*100000)	Velocity (UU/MU * 100000) in startpoint
	2	STPoint_1.b-veloc.	(dec)	(*100000)	Velocity (UU/MU * 100000) in startpoint
	...				
StartPath Interpolation Points	1	StartPoint_1	UU	0	Interpolation Point no. 1 - for start path
	...				
	n				
StopPath Interpolation Points	1	StopPoint_1	UU	0	Interpolation Point no. 1 - for stop path
	...				
	n				

■ Stichwortverzeichnis
Symbole

#DEBUG	170
#INCLUDE	171
_GETVEL	170
4-Setup	176

A

Abbrechen	29
Programmausführung	59
Abbruch alle im Menü Entwicklung	61
Abkürzungen	179
Absolutbewegungen	102
Abtastzeit des gesamten Regelalgorithmus	37
ACC	105
ACCMAXQC	183
Achsparameter	65, 177
einer Konfigurationsdatei ändern	66
online einstellen oder ändern	65
Werkseinstellungen	65
Achsprozessdaten	162
Analogausgänge	10
Analogeingänge	10
Ändern der Parameter	172
Anschlussbeispiele	16
Anzahl Intervalle	80
APOS	105
Arithmetik	99
Array Structure of CAM Profiles	235
Arrays	98
Kurvenscheibensteuerung	42
schreiben und lesen	99
versus Variablen	99
Assignment Operation	101
Aufzeichnung anzeigen	71
Ausführen	
abbrechen	59
Programm	58
Ausrichten an Gitter	75
Autostart	63
AVEL	106
AXEND	106

B

BANDWIDTH (35)	37, 195
Bearbeiten	
Ausschneiden	57
Einfügen	57
Kopieren und Einfügen	57
Rückgängig	57
Suchen und Ersetzen	58
Bedeutung und Einfluss der Reglerparameter	35
Befehlsabarbeitung	95
Befehlsaufbau, grundsätzlicher	95

Befehlsausführungszeiten	95
Befehlshilfe	61
Beispiel eines Nockenschaltwerks	52
Beispiele Kurvenscheibensteuerung	
Abstand Sensor größer als eine Masterzykluslänge	48
Kartons bedrucken mit Markerkorrektur	46
Kartons mit Haltbarkeitsdatum stempeln	43
Problemfälle bei Festlegung des Markerabstands	48
Slave-Synchronisation mit Marker	49
Beispiele zur Regleroptimierung	40
Bestimmung der Maximalgeschwindigkeit	41
Dämpfung der Schwingungen	40
Benutzereinheit	101, 228
Berechnen	75
Beschleunigungs-Feed-forward FFACC(37)	37, 195
Beschleunigungs-Limit	81

C

CAM-Befehle	104
CAM-Editor	73
Ausrichten an Gitter	75
Bearbeiten	75
Berechnen	75
Bevor Sie beginnen, eine Kurve zu editieren	73
Drucken und Drucken SW	74
Ende	74
Export	74
Fenster	73
Import	74
Kurvenprofil	76
Laden CNF	74
Menü Ansicht	76
Menü Datei CNF	74
Menü Fenster	76
Menüleiste	73
Neue CNF	74
Permanent Rechnen	75
Rückgängig	75
Sichern CNF	74
Titelleiste	73
Wiederherstellen	75
CAM-Editor-Fenster ohne Kurvendaten	91
CAM-Mode	42
COMOPTGET	107
COMOPTSEND	108
CONTINUE	108
CPOS	109
CSTART	109
CSTOP	110
CURVEPOS	110
CVEL	111
D	
D.TIME CMP.ACT. (780)	175
Datei	55
neu	55

öffnen	55	ENCODER	180
schließen	55	ENCODERTYPE	192
speichern / speichern als	56	ENDSWMOD	197
suchen	55	Entwicklung	58
Dateityp	55	Abbrechen	59
Datentyp	176	Abbruch alle	61
Debug Modus	59	Ausführen	58
Beenden Debug	60	Befehlshilfe	61
Einzelschritt	60	Fortsetzen	59
Start	60	Steuerung auswählen	61
Überwachung anzeigen	60	Syntaxprüfung	61
Variablen online ändern	60	Vorbereiten Einzelschritt	59
Vorbereiten Einzelschritt	59	ERRCLR	116
Debug Wert setzen	60	ERRCOND	197
DEC	111	ERRNO	117
DEF ORIGIN	112	ESCCOND (70)	209
DEF SYNCORIGIN	112	EXIT	117
DEFMCPPOS	112	Export	74
DEFMORIGIN	112	Externe 24 V DC Stromversorgung	15
DELAY	113	F	
DELETE ARRAYS	113	Fahrdiagramme auswerten	71
DFLTACC	194	Fahrtst durchführen	33
DFLTVEL	194	Fahrtweg für Testfahrt	69
Dialogfelder	25	Farben Editor	83
Dialogsprache ändern	28	Fehler löschen bei Testfahrt	70
Differentialfaktor KDER (12)	184	Fenster	
Digitalausgänge / Digitaleingänge	10, 12, 15	CAM-Editor-Fenster ändern	73
DIM	114	Größe ändern	26
DIM Anweisung	98	Horizontal / Vertikal	26
Dip switch	232	Kaskadieren	84
DISABLE ... Interrupts	115	FFACC (37)	37, 195
Drehgeber		FFVEL (36)	37, 195
Ausgang (Virtueller Master)	16	Fixpunkte	77
Drehgeberanschluss prüfen	32	einfügen	76
Eingang 1 (Master)	15	in der Tabelle ergänzen oder löschen	78
Eingang 2 (Slave)	16	Registerkarte	77
Schnittstelle 1 MK3B	12	Fortsetzen (Programm)	59
Überwachung	13	Funktionstasten	27
Drehzahlregelung	102	G	
Drucken	56	Geschwindigkeits-Feed-forward FFVEL(36)	37, 195
Druckereinstellungen	56	Geschwindigkeits-Limit	81
E		GET	118
EEPROM löschen	68	GETVLT	118
Ein-/Ausgabe-Befehle	102	GETVLTSUB	119
Einfügen		Globale Parameter	64, 177
Fixpunkte in der Tabelle	78	GOSUB	119
Fixpunkte mit der Maus	76	GOTO	120
im Menü Bearbeiten	57	H	
in der Befehlshilfe	62	HOME	120
Eingabebereiche	179	HOME_FORCE	180
Einstellungen		HOME_OFFSET	197
Farben Editor	83	HOME_RAMP	196
Sprache	83	HOME_VEL	182
Einzelschritt	59, 60		
ENABLE ... Interrupts	116		

HOMETYPE	196	Kurvenname	79
I		Kurvenprofil	76
I_BREAK	210	scrollen	77
I_CONTINUE	210	vergrößern und verkleinern	77
I_ERRCLR	211	Kurvenpunkt	78
I_NEGLIMITSW	198	Kurvenscheibensteuerung	42
I_POSLIMITSW	198	Beispiel mit Markerkorrektur	46
I_PRGCHOICE	210	Beispiel ohne Marker	43
I_REFSWITCH	197	Beispiel Slave-Synchronisation mit Marker	49
IF ..THEN .., ELSEIF .. THEN .. ELSE .. ENDIF	121	Genauigkeit	42
Import	74	Interpolation	42
IN	122	Prinzipskizze	42
In Datei kompilieren	61	Problemfälle	48
INAD	122	Schnellkurs für Ungeduldige	43
INB	123	Kurventyp	79
INDEX	123	L	
Indizes	99	Layout der Optionskarte	14
Initialisierung	95	LED	228
SyncPos Motion Controller	102	LINKAXPAR	126
INKEY	124	LINKGPAR	126
Integrialfaktor KINT (13)	36, 184	LOOP	127
Integrationslimit KILIM (21)	37, 189	M	
Interpolation	42	MAPOS	128
Interrupt	96	Marker Window Master / Slave	208
Abarbeitung von Interrupt-Prozeduren	96	Master Länge	80
Benutzung von Variablen	96	Master Marker	
Funktionen	103	Abstand	81
Interrupt Schachtelung	97	Master Reset	29
ON PERIOD	96	Master-Marker	
Prioritäten	97	Position	80
Reaktionszeiten	97	Toleranz	81
Intervall-Dauer (ms)	81	Masterunit [MU]	101
Intervall-Größe	81	Maus	26
IPOS	125	MAVEL	128
J		Max. aktuelle Beschleunigung	81
Jetzt ausführen	62	Max. aktuelle Geschwindigkeit	80
K		Max. Slave Beschleunigung	81
KDER (12)	184	Max. Slave Geschwindigkeit	81
KILIM (21)	37, 189	MCT10 Modus	
KINT (13)	36, 184	SyncPos-Fenster	89
Klemmen	15	MCT10-Modus	
Kommunikation einrichten	29	Menü CAM-Editor	91
beenden	30	Menü Datei	90
Kommunikationsoption	104	Menü Entwicklung	90
Konstanten	98	Meldungen	
Kontextmenü	57	Achse nicht vorhanden	215
Konventionen	5, 7	Benutzer Abbruch	217
Konvertierungsindex	176	Endschalter erreicht	217
Korrektur Start	80	Falsche Parameternummer	216
KPROP (11)	36, 184	Fehler aus Encoder Überwachung	219
Kurve Bearbeiten	75	Fehler beim Compilieren	220
Kurven-Daten	79	Fehler beim Verifizieren	218
Kurven-Info	79, 80	Fehler in Datei: Achsparameter	220
		Fehler in Datei: Array-Daten	221

Fehler in Datei: Globale Parameter	221	Nockenschaltwerk	42, 52
Fehler in DIM Anweisung	218	NOWAIT	132
Fehler nicht beseitigt	215	in Interrupts	97
Feldgrenzen über- oder unterschritten	218	Nullimpuls	228
HOME noch nicht angefahren	215	O	
Indeximpuls nicht gefunden	215	O.ERR_3 bis xx	215
Interner Fehler ##	220	O_AXMOVE	205
Kein Platz mehr für Variablen	219	O_BRAKE	198
Kurven-Array in DIM-Anweisung falsch	219	O_ERROR	211
MEMORY locked	219	ON APOS .. GOSUB	132
Parameter im EEPROM defekt!	216	ON COMBIT .. GOSUB	133
Programme im EEPROM defekt!	217	ON DELETE .. GOSUB	133
Reset durch CPU ausgelöst!	217	ON ERROR GOSUB	134
Schleppabstand überschritten	215	ON INT .. GOSUB	135
Software-Endschalter überschritten	216	ON MAPOS .. GOSUB	136
Steuerung führt ein Programm .. aus	221	ON MCPOS .. GOSUB	137
Timeout bei Warten auf Index	219	ON PARAM .. GOSUB	138
Unbekannter Befehl	215	ON PERIOD	96, 138
Zu viele Interruptfunktionen	217	ON STATBIT .. GOSUB	139
Zu viele Return	218	ON TIME	140
Zu viele Time Interrupts	219	Option card layout	14
Zu viele verschachtelte Unterprogramme	217	Optionskartenklemmen	11
Meldungen des CAM-Editors	77	OUT	140
MENCODER	193	OUTAN	141
MENCODERTYPE	207	OUTB	141
Menü	25, 56, 57	OUTDA	142
Bearbeiten	57	P	
CAM-Editor	73	Parameter	
Datei	55	Allgemeines zu den Parameterwerten	179
Einstellungen	82	Befehlsgruppe (PAR)	103
Entwicklung	58	Name	67
Fenster	84	speichern in Datei	67
Hilfe	85	wiederherstellen aus Datei	67
Steuerung	62	Parameter-Referenz	172
MIPOS	129	PC-Software Benutzeroberfläche	53, 87
MK3A	11	PCD	142
MK3B	12	Permanent Rechnen	75
MK3C	12	Pfeiltasten	26
MK3D	12	PID	143
MLONG	228	PID-Regelung	35
MOTOR OFF	130	optimieren	35
MOTOR ON	130	Popup-Menü	25
MOTOR STOP	131	POSA	143
Motordrehrichtung prüfen	32	POSA CURVEPOS	144
MOVESYNCORIGIN	131	POSDRCT	193
Mt	228	POSERR (15)	185
MTC10		POSFAC_T_N	192
Betriebsmodus: Offline oder Online	89	POSFAC_T_Z	190
Dateien-Handling	89	Position anfahren (Teach)	62
SyncPos starten	89	POSLIMIT	181
MU	101	POSR	144
N		PRGPAR (102)	209
negierter Wert	228	PRINT	145
NEGLIMIT	181		
Neue Hardware	232		

Priorität	Reguläre Ausdrücke	58
bei Interrupt-Ereignissen	REGWMAX	196
der Operatoren und Operationen	REGWMIN	196
Profilgenerator Werte	Relaisausgänge	10
PROFTIME (29)	REPEAT .. UNTIL	146
Programm	Reset	
fortsetzen (nach Abbruch)	Arrays	68
starten	Parameter	68
Programmaufbau	vollständig	68
Definitionen	REVERS	205
Hauptprogramm	Richtungstasten	26
Initialisierung	RS485 Anschluss	30
Unterprogramm	RST ORIGIN	147
Programmbeispiele	Rückgängig	27, 57
COM_OPT	Rücklesen Quellcode	63
Marker count		
Simulation eines Masters per Software-Befehl	S	
Programme	SAVE part	147
Alle löschen	SAVEPROM	147
in mehreren VLTs ausführen	Schleppabstand	228
sichern	Schnittstelle schließen	61
Programmende im Menü Datei	SET	148
Programminfo	SET ORIGIN	150
Programmlesbarkeit, Tips	SETCURVE	148
Programm ID 781	SETMORIGIN	149
Proportionalfaktor KPROP (11)	SETVLT	150
PULSACC	SETVLTSSUB	151
PULSVEL	Shortcut-Liste	27
Punkte einfügen oder löschen	Shortcuts	26
Punktetyp ändern	Sicherheit	4
Q	Sichern	
Quadcounts	mit Quellcode	63
Quellcode	Slave-Marker	
rücklesen	Abstand	81
sichern	Position	80
	Toleranz	81
R	Slave-Stop-Position	80
RAM sichern	Slave-Synchronisation mit Marker	49
RAMPMIN	Sollwert-Generator	35
RAMPTYPE	Speicher im Menü Steuerung	68
Reaktionszeiten	Speichern	
Regelprozess: So funktioniert der	der Datei	56
Registerkarte	der Parameter	172
Fixpunkte	Speicherplatz	56
Geschwindigkeit	Sprache im Menü Einstellungen	83
Kurven-Daten	Sprachelemente	98
Kurven-Info	Start Stop Punkte	78
Start Stop Punkte	STAT	151
Synchronisation	Steuerung	62
Regleroptimierung	alle löschen	64
Beispiele	auswählen	30, 61
Schritt für Schritt ... optimieren	Parameter	64
Zehn Schritte zur optimalen Regelung	Programme	62
Reglerparameter	Reset	68
Bedeutung und Einfluss der Reglerparameter	Speicher	68
	SUBMAINPROG .. ENDPROG	152

SUBPROG name .. RETURN	152	SYNCSMHIT	160
Suchen über Index	85	SYNCSTAT	159
Suchen und Ersetzen	58	SYNCSTATCLR	160
Reguläre Ausdrücke	58	SYNCTYPE (51)	200
SWAPMENC	153	SYNCV	161
SWNEGLIMACT	187, 188	SYNCVELREL	207
SWPOSLIMACT	189	SYNCVFTIME	206
Symbole anordnen	84	Syntaxprüfung	61
Symbolleiste	24	System-Prozessdaten	162
SYNCACCURACY	159, 202	SYSVAR	161
SYNCC	153	T	
SYNCCMM	154	Tangentenpunkt	42, 78
SYNCCMS	155	Tastatur	26
SYNCCSTART	155	Technische Daten	10, 15
SYNCCSTOP	156	Technische Referenz	235
SYNCERR	157	temporäres Programm	62
SYNCFACTM	199	Testfahrt	69
Syncfactor Master und Slave	81	Aufzeichnung anzeigen	71
SYNCFACTS	200	durchführen	70
SYNCFAULT	159, 202	starten	70
Synchronisation (SYN)	104	wiederholen	71
SYNCM	157	Testfahrt-Grafik	
SYNCMARKM	201	Beschleunigung	72
SYNCMARKS	201	Geschwindigkeit	72
SYNCMFPAR (17)	186	Position	72
SYNCMFTIME (18)	187	Strom	72
SYNCMMAXCORR (6)	181	Testfahrt-Parameter	
SYNCMMERR	160	Abtastintervall	69
SYNCMMHIT	160	Anzahl der Messungen	69
SYNCMPULSM	203	Beschleunigung	69
SYNCMPULSS	203	sichern	39
SYNCMSTART	204	Verzögerung (Bremsbeschleunigung)	69
SYNCMTYPM	203	TESTSETP	164
SYNCMTYPS	204	TESTSTART	165
SYNCMWINM	208	TESTTIM	191
SYNCMWINS	208	TESTVAL	191
SYNCOFFTIME (16)	186	TESTWIN	182
SYNCP	158	Text löschen	57
SyncPos		Text markieren	57
beenden	56	TIME	166
Befehle Übersicht	102	TIMER	185
Fenster	24	TRACKERR	166
im MTC10 Modus starten / beenden	89	Typ	78
installieren	28	U	
starten	28	Überwachung anzeigen	60
SyncPos Parameter	172	Weitere Ausdrücke hinzufügen	61
im Detail	179	V	
Übersicht	177	Variablen	98
SyncPos Parameter einstellen	30	Namen	98
Encoder	31	spezielle Variablen	98
Geschwindigkeit	31	Variablen online ändern	60
Home	31	VEL	167
Synchronisieren	31	VELMAX	179
SYNCPOSOFFS	201		
SYNCREADY	159, 202		
SYNCSMERR	160		

VELMAXQC	183
VELRES	189
Verkettung von Autostartprogrammen	64
Versorgungsspannungen	13
Vertikal (nebeneinander)	84
VLT Steuerkartenklemmen	10
VLT-Parameter	172, 173
einstellen	29
Volltextsuche	85
Voraussetzungen	24
Vorbereiten Einzelschritt	59

W

WAITAX	167
WAITI	168
WAITNDX	168
WAITP	169
WAITT	169
Warnung vor unbeabsichtigtem Anlaufen	4
Was tun ... wenn die Kurve nicht dargestellt wird	77
Was tun wenn ...	
Drehgeber funktioniert nicht	33
maximale Beschleunigung nicht erreicht wird	40
Motor bewegt sich nicht	33
Motor fährt unkontrolliert los	33
Motor schwingt stark	33
Neigung zur Instabilität besteht	40
Schleppabstand überschritten wird	40
Schleppfehler gemeldet wird	34
stationäre Genauigkeit gefordert wird	40
Werkseinstellungen	172, 179
Werteingaben	96
WHILE .. DO .. ENDWHILE	170

Z

Zehnertastatur	26
Zeilennummer	57
Zuweisung	101
Zyklen / min Master	80