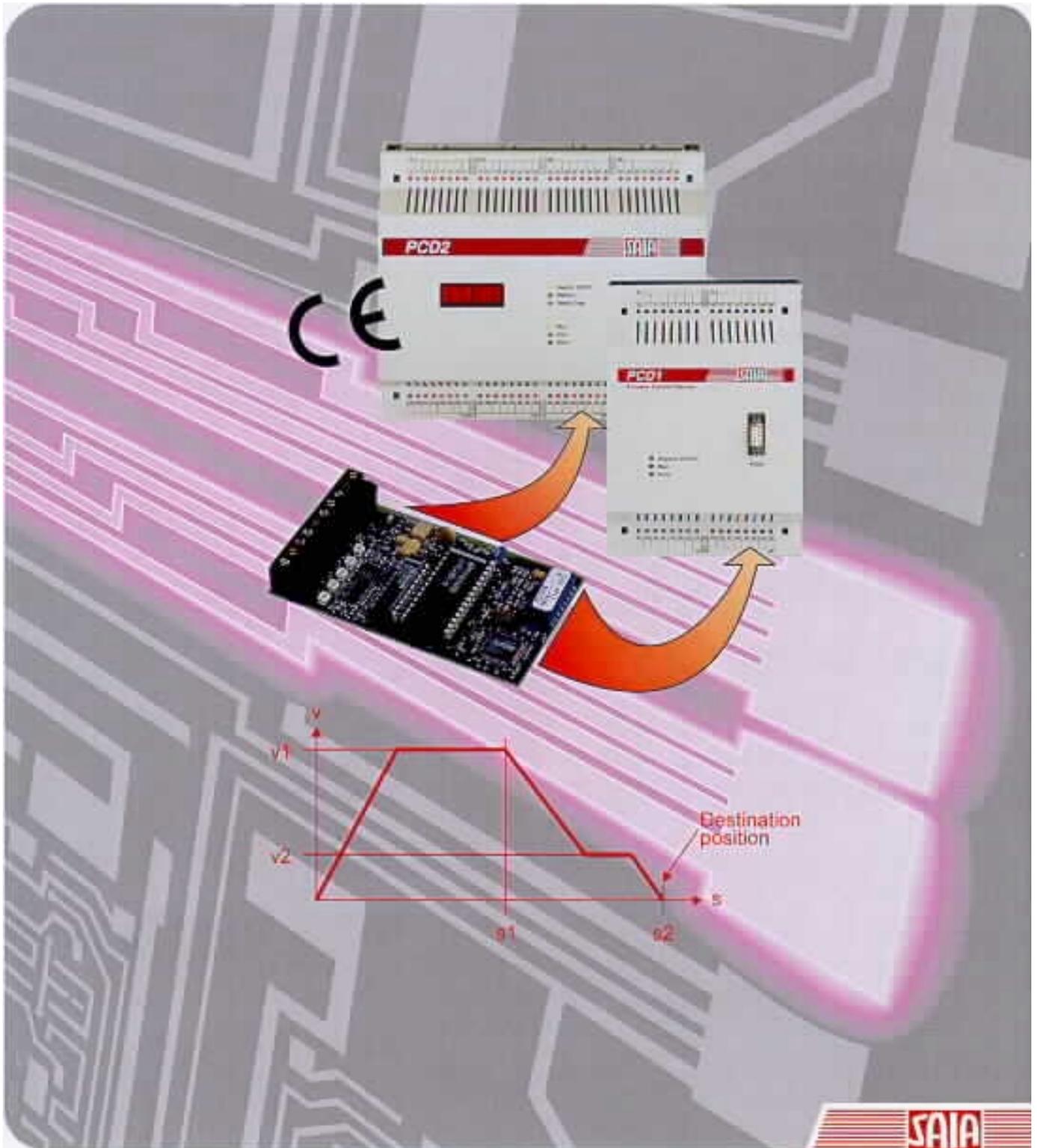


SAIA® PCD
Process Control Devices

PCD2.H31x
Positioniermodule
für Servoantriebe



SAIA-Burgess Gesellschaften

Schweiz	SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83	Frankreich	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
Deutschland	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	Niederlande	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
Österreich	SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11	Belgien	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
Italien	SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92	Ungarn	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

Vertretungen

Gross-britannien	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	Portugal	INFOCONTROL Electronica e Automatismo LDA. Praceta Cesário Verde, No 10 s/cv, Massamá P-2745 Queluz ☎ 21 430 08 24, Fax 21 430 08 04
Dänemark	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	Spanien	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
Norwegen	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	Tschechische Republik	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
Schweden	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	Polen	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
Suomi/ Finnland	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
Australien	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	Argentinien	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

Kundendienst

USA	SAIA-Burgess Electronics Inc. 1335 Barclay Boulevard Buffalo Grove, IL 60089, USA ☎ 847 215 96 00, Fax 847 215 96 06
------------	---

SAIA® Process Control Devices

Positioniermodule für Servoantriebe

PCD2.H31x

SAIA-Burgess Electronics AG 1999. Alle Rechte vorbehalten
Ausgabe 26/762 D2 - 09.99

Technische Änderungen vorbehalten

Anpassungen

Handbuch: PCD2.H31x - Positioniermodule für Servoantriebe - Ausgabe D2

Datum	Abschnitt	Seite	Beschreibung
12.05.2000	Anhang A	A 36	div. Korrekturen

Inhalt

	Seite
1. Einführung	
1.1 Allgemeines	1-1
1.2 Funktion und Anwendung	1-2
1.3 Die wichtigsten Eigenschaften	1-3
1.4 Typische Einsatzgebiete	1-4
1.5 Programmierung	1-5
1.6 Betriebsarten	1-6
1.7 Inbetriebnahme	1-7
2. Technische Daten	
2.1 Technische Daten der Hardware	2-1
2.2 Elektrische Spezifikationen	2-3
2.3 Funktionsspezifische Daten	2-4
3. Präsentation	
4. Anschlüsse und Bedeutung der LEDs	
5. Funktionsbeschreibung	
5.1 Betriebsarten	5-1
5.2 Generator für das Geschwindigkeitsprofil	5-2
5.3 PID-Regler	5-4
5.4 Positionsdecoder und Eingangsschaltung	5-5
5.5 D/A-Wandler	5-8
5.6 Ergänzende Information: Referenzfahren (FB Home)	5-9
6. Der schnelle Einstieg	
6.1 Einstieg mit Programmierung in IL	6-2
6.1.1 Einsteiger-Beispiel in IL mit Warteschleifen	6-3
6.1.2 Einsteiger-Beispiel in GRAFTEC	6-6
6.1.3 Einfaches Inbetriebnahme-Programm	6-11
6.2 Einstieg mit Programmierung in FUPLA	6-13

Anhang A: Zusammenfassung aller Funktions-Blocks (FB)

Init:	Initialisierungs-FB	A-1
Home:	Referenzpositions-FB	A-3
Exec:	Ausführungs-FB	A-5
StartMot	Start motion	A-7
StopUrg	Stop motion in urgency	A-8
Stop	Stop motion	A-9
MotOff	Motor regulation off	A-10
RdActPos	Read Actual Position	A-11
RdActVel	Read Actual Velocity	A-12
RdIntSum	Read Integration Sum	A-13
RdIndexRg	Read Index Register	A-14
RdStatRg	Read Status Register	A-15
RdTargPos	Read Target Position	A-16
RdTargVel	Read Target Velocity	A-17
GoForw	Go Forwards	A-18
GoBackw	Go Backwards	A-19
SgStpFor	Single Step Forwards	A-20
SgStpBak	Single Step Backwards	A-21
LdDestAbs	Load Destination Absolute	A-22
LdDestRel	Load Destination Relative	A-23
LdVelAbs	Load Velocity Absolute	A-24
LdVelRel	Load Velocity Relative	A-25
LdAccAbs	Load Acceleration Absolute	A-26
LdAccRel	Load Acceleration Relative	A-27
LdPropG	Load Proportional Gain	A-28
LdIntG	Load Integrative Gain	A-29
LdDerG	Load Derivative Gain	A-30
LdSampInt	Load derivative Sampling interval	A-31
LdIntLim	Load Integrative Limit	A-32
ActRegFact	Activate Regulation Factors	A-33
LdBrkPtAbs	Load Brakpoint Absolute	A-34
LdBrkPtRel	Load Breakpoint Relative	A-35
ResStatRg	Reset Status Register	A-36
SetIdxPos	Set Index Position	A-37
SetZero	Set Home position	A-38
MotConf	Motion Configuration	A-39
SetPosTol	Set Position Tolerance	A-40

Anhang B: Zusammenfassung aller Function-Boxes (FBox)



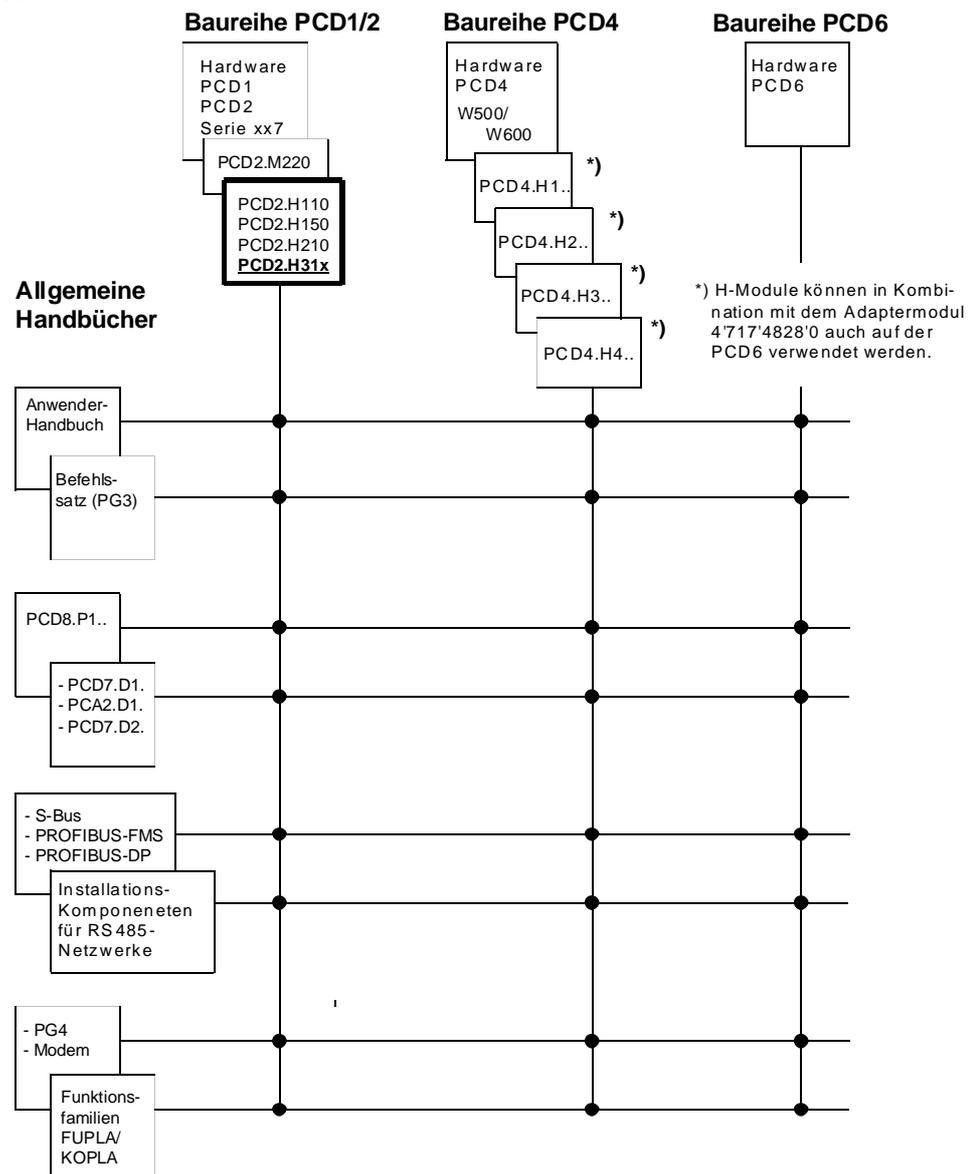
Wichtiger Hinweis:

Um den einwandfreien Betrieb von SAIA® PCD sicherstellen zu können, wurde eine Vielzahl detaillierter Handbücher geschaffen. Diese wenden sich an technisch qualifiziertes Personal, das nach Möglichkeit auch unsere Workshops erfolgreich absolviert hat.

Die vielfältigen Leistungen der SAIA® PCD treten nur dann optimal in Erscheinung, wenn alle in diesen Handbüchern aufgeführten Angaben und Richtlinien bezüglich Montage, Verkabelung, Programmierung und Inbetriebnahme genau befolgt werden.

Damit allerdings werden Sie zum grossen Kreis der begeisterten SAIA® PCD Anwendern gehören.

Übersicht



Zuverlässigkeit und Sicherheit elektronischer Steuerungen

Die Firma SAIA-Burgess Electronics AG konzipiert, entwickelt und stellt ihre Produkte mit aller Sorgfalt her:

- Neuster Stand der Technik
- Einhaltung der Normen
- Zertifiziert nach ISO 9001
- Internationale Approbationen: z.B. Germanischer Lloyd, UL, Det Norske Veritas, CE-Zeichen ...
- Auswahl qualitativ hochwertiger Bauelemente
- Kontrollen in verschiedenen Stufen der Fertigung
- In-Circuit-Tests
- Run-in (Wärmelauf bei 85°C während 48h)

Die daraus resultierende hochstehende Qualität zeigt trotz aller Sorgfalt Grenzen. So ist z.B. mit natürlichen Ausfällen von Bauelementen zu rechnen. Für diese gibt die Firma SAIA-Burgess Electronics AG Garantie gemäss den "Allgemeinen Lieferbedingungen".

Der Anlagebauer seinerseits muss auch seinen Teil für das zuverlässige Arbeiten einer Anlage beitragen. So ist er dafür verantwortlich, dass die Steuerung datenkonform eingesetzt wird und keine Überbeanspruchungen, z.B. auf Temperaturbereiche, Überspannungen und Störfelder oder mechanischen Beanspruchungen auftreten.

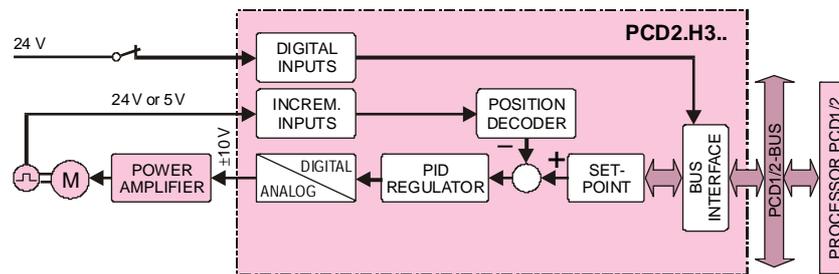
Darüber hinaus ist der Anlagebauer auch dafür verantwortlich, dass ein fehlerhaftes Produkt in keinem Fall zu Verletzungen oder gar zum Tod von Personen bzw. zur Beschädigung oder Zerstörung von Sachen führen kann. Die einschlägigen Sicherheitsvorschriften sind in jedem Fall einzuhalten. Gefährliche Fehler müssen durch zusätzliche Massnahmen erkannt und hinsichtlich ihrer Auswirkung blockiert werden. So sind z.B. für die Sicherheit wichtige Ausgänge auf Eingänge zurückzuführen und softwaremässig zu überwachen. Es sind die Diagnoseelemente der PCD wie Watch-Dog, Ausnahme-Organisations-Blocks (XOB) sowie Test- und Diagnose-Befehle konsequent anzuwenden.

Werden alle diese Punkte berücksichtigt, verfügen Sie mit der SAIA® PCD über eine moderne und sichere programmierbare Steuerung, die Ihre Anlage über viele Jahre zuverlässig steuern, regeln und überwachen wird.

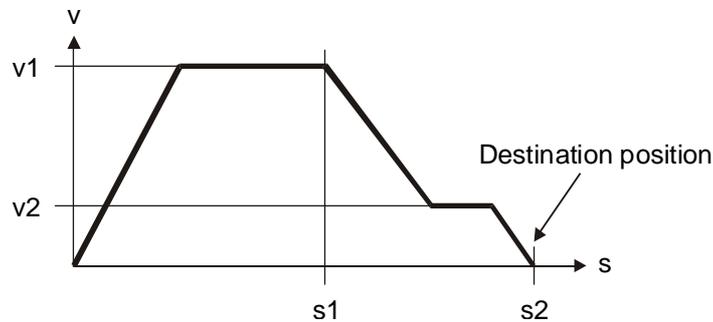
1. Einführung

1.1 Allgemeines

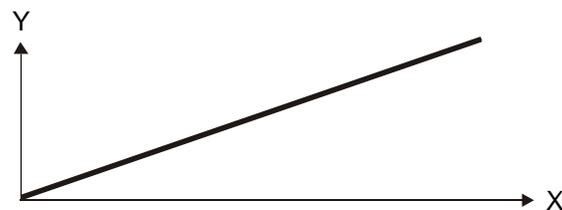
Das Positioniermodul PCD2.H3.. ist ein intelligentes E/A-Modul der Baureihe PCD2. Das Modul dient der Positionierung einer unabhängigen Achse mit drehzahlregelbarem Antrieb (Servomotor). Ein solcher Servomotor kann ein regelbarer DC- oder AC-Motor sein, welcher über eine Leistungsstufe und einen Inkrementaldrehgeber zur Positions- und Drehzahlfassung verfügt.



Blockschema eines Servoantriebes für 1 Achse



Positionsgeregelter Betrieb mit konstanter Beschleunigung und Verzögerung sowie Anfahren der Zielposition mit Langsamvorschub



Koordinierter Quasi-Synchronbetrieb von 2 Achsen

1.2 Funktion und Anwendung

Das ..H3..-Modul dient zur Positionierung einer einzelnen Achse mit drehzahlregelbaren DC- oder AC-Servomotoren. Dabei muss die Antriebseinheit über eine Leistungsstufe sowie einen Inkrementaldrehgeber zur Positions- oder Drehzahlerfassung verfügen. In einem PCD1-System können bis zu 4 Positioniermodule (4 Achsen) und in einer PCD2 bis zu 8 Module ..H3.. (8 Achsen) an beliebiger Stelle auf den I/O-Bus gesteckt und zusammen mit allen Basisgeräten betrieben werden.

Auf dem Positioniermodul befindet sich ein Singlechip-Prozessor. Dieser steuert selbständig und PID-geregt jede Bewegung entsprechend den vom Anwenderprogramm übergebenen Parametern (Geschwindigkeit, Beschleunigung und Zielposition). Dadurch lässt sich jede Achse unabhängig steuern. Eine Verkettung mehrerer Achsen (Punkt-Punkt) im koordinierten Quasi-Synchronbetrieb ist programmierbar. Damit wird durch kartesische Achsen eine lineare Bewegung erreicht.

Die Positioniermodule PCD2.H31x dürfen nur im Basisgerät PCD2.M1xx und nicht im Erweiterungsgerät PCD2.C1xx gesteckt werden. Dies, um dem relativ hohen Stromverbrauch Rechnung zu tragen.

1.3 Die wichtigsten Eigenschaften

- Ideal für kompakte und preisgünstige Maschinen
- PID-geregelte Steuerung der Position und Drehzahl von Servomotoren
- Geschwindigkeit, Zielposition und PID-Parameter können selbst während der Bewegung verändert werden
- Analoges $\pm 10V$ -Ausgang mit 12 Bit inkl. Vorzeichen zur Ansteuerung der Leistungsstufe für den Motor
- Digitaler Eingang für Referenzschalter beim .. H310
- Encodersignal-Eingänge von 24 VDC (Quellbetrieb) oder 5 V/RS 422 (Antivalent Line Driver)
- Andere achsenpezifische Ein- und Ausgänge, wie z.B. Limit Switches, der Referenzschalter für das Modul PCD2.H311, Enable für den Driver, müssen mit einem Standard-I/O-Modul (z.B. PCD2.B100, PCD2.E110/E111 bzw. PCD2.A410) überwacht resp. gesteuert werden.

1.4 Typische Einsatzgebiete

- Handlingroboter
- Bestückungs- und Montageautomaten
- Palettierautomaten
- Verpackungsmaschinen
- Blechbearbeitungsmaschinen
- Bohrautomaten
- und anderes mehr

1.5 Programmierung

Dank praxisbezogener Funktionsbausteine (FBs und FBoxen) müssen für die verschiedenen Bewegungs- und Fahrfunktionen lediglich die gewünschten Parameter eingegeben werden. Das vorliegende detaillierte Handbuch enthält die Beschreibung jedes Funktionbausteines und wird ergänzt durch praxisgerechte Anwendungsbeispiele. Die Programmierung erfolgt mit dem Standard-Programmierwerkzeug PG4 ab Version V2.0.70 entweder in IL (Instruction List) oder grafisch (FUPLA).

Initialisierungsbefehl

INIT FB, welchem beim Aufruf 11 Parameter mitgegeben werden

siehe 7.1.2 und Anhang A, Seiten A1/2

Ausführungsbefehl

EXEC FB mit jeweils 3 Parametern. Es können über 30 verschiedene Befehle ausgeführt werden.

siehe 7.1.2 und Anhang A, Seiten A5/40

Einrichtungsbefehl

HOME FB zur automatischen Suche des Referenzschalters. FB mit 7 Parametern

siehe 7.1.2 und Anhang A, Seiten A3/4

Diagnose und Fehlerbehandlung

Erkennung von falschen FB-Parametern (Diagnose-Register)
Timeout-Überwachung für FB 'Home'.

Siehe Kapitel 8.

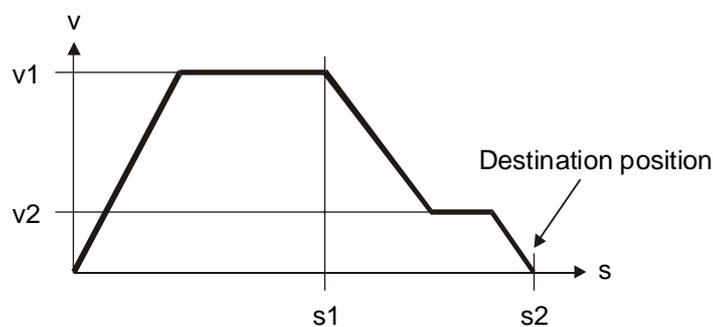
1.6 Betriebsarten

Bei der Lösung von Positionierungsaufgaben mit Servoantrieb wird grundsätzlich zwischen zwei Betriebsarten unterschieden:

- positionsgeregelter Betrieb
- drehzahl geregelter Betrieb

Im positionsgeregelten Betrieb wird eine vorgegebene Zielposition nach Eingabe verschiedener Parameter (PID-Faktoren, Beschleunigung, Geschwindigkeit usw.) geregelt angefahren.

Im drehzahl geregelten Betrieb wird mit einer vorgegebenen Beschleunigung bis zur Sollgeschwindigkeit hochgefahren. Mit dieser Geschwindigkeit wird solange geregelt gefahren, bis ein Stopbefehl erfolgt. Die Sollgeschwindigkeit lässt sich auch während der Bewegung verändern.



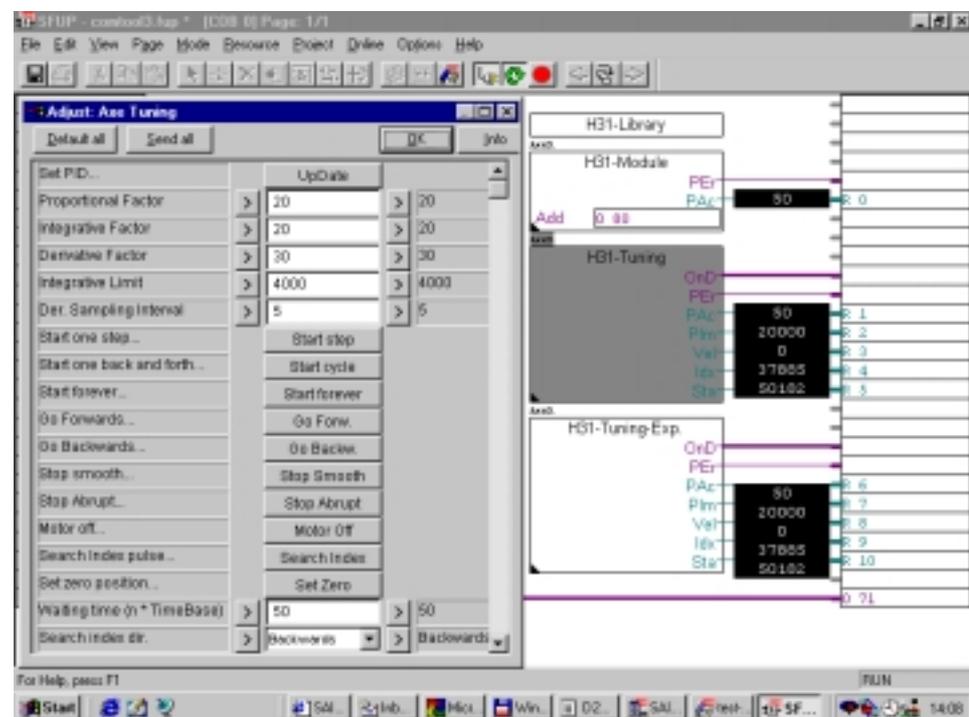
Positionsgeregelter Betrieb mit konstanter Beschleunigung und Verzögerung sowie Anfahren der Zielposition mit Langsamvorschub

1.7 Inbetriebnahme

Für die Inbetriebnahme steht eine komfortable Software in der Form eines FUPLA-Programms zur Verfügung. Mit diesem Werkzeug können alle während der Inbetriebnahme notwendigen Tests und Einstellungen auf einfache Art und Weise online vorgenommen oder überprüft werden.

Dieses Commissioning Tool ist auf Diskette erhältlich, Bestellnummer: PCD8.H31.

Näheres siehe Kapitel 10.



2. Technische Daten

2.1 Technische Daten der Hardware

Digitale Eingänge des Moduls PCD2.H310

Anzahl Eingänge:	1 Encoder A, B, IN 1 Referenz-Eingang
Eingangsspannung:	24V typisch
Bereich "Low":	0 ... +4V
Bereich "High":	+15 ... +30V
Nur Quellbetrieb (pos. Logik)	
Eingangsstrom bei 24 VDC:	6 mA (typisch)
Schaltungsart	galvanisch verbunden
Reaktionszeit	30 μ s

Digitale Eingänge des Moduls PCD2.H311

Anzahl Eingänge:	1 Encoder A, /A, B, /B, IN, /IN (kein Referenz-Eingang)
Eingangsspannung:	5V typisch
Signalpegel:	Antivalent-Eingänge nach RS422
Hysterese:	max. 200 mV
Abschlusswiderstand:	150 Ω

Analog Ausgang für die Module PCD2.H310/311

Analoger Regler-Ausgang	Auflösung 12 Bit (mit Vorzeichen)
Kurzschluss-Schutz	ja
Galvanische Trennung	nein
Ausgangsspannung *)	\pm 10V, Abgleichtoleranz \pm 5 mV
Logik	positiv (plus-schaltend)
Minimale Lastimpedanz	3 k Ω

5V-Speisung für 5V-Encoder für das Modul PCD2.H311

5V-Ausgang	5V-Speisung des Encoders
Kurzschluss-Schutz	ja
Galvanische Trennung	nein
Ausgangsspannung	5V
Max. Laststrom	300 mA
Kurzschlussstrom	400 mA
(Dieser Strom belastet zusätzlich den 5V-Bus der PCD1/2)	

*) Die Abgleichung der Ausgangsspannung wird im Werk vorgenommen.
Es wird daher dringend davon abgeraten, das Abstimmpotentiometer zu verstellen.

Betriebsbedingungen

Umgebungstemperatur	Betrieb: 0 ...+50°C ohne Zwangsbelüftung Lagerung: -20 ... +85°C
Störimmunität	CE-Zeichen gemäss EN 50081-1 und EN 50082-2

LED-Anzeigen und softwarmässige Abfragemöglichkeiten

Anzahl	5
LED "A"	Status des Encoder-Eingangs "A"
LED "B"	Status des Encoder-Eingangs "B"
LED "IN"	Status des Index-Eingangs
LED "Ref"	Status des Referenzschalters (H310)
LED "Pw 5V"	Speisung (5V) des Encoders (H311)
LED "Power"	Speisung $\pm 15V$
Softwaremässige Abfragen	
Input Power (Adr. 08)	Erlaubt die softwaremässige Überwachung der Speisungen
Input Ref (Adr. 11)	Erlaubt das Abfragen des logischen Zustands des Referenz-Schalters (H310)
Input Pw5V (Adr. 11)	Erlaubt die softwaremässige Überwachung der 5V-Speisung (H311)
Input Version (Adr. 12)	Erlaubt die Abfrage des Modultyps H310 oder H311 (H = H310, L = H311)

Allgemeines

Prozessor	LM 628
Programmierung	Auf PCD-Anwenderprogramm (PG4) basierend und unterstützt von FB- und FBox-Bibliothek.

Bestellangaben

PCD2.H310	1 Achse für Encoder 24 VDC
PCD2.H311	1 Achse für Encoder 5V/RS422
PCD9.H31E	Software-Bibliothek mit Funktionsbausteinen (FBs) für die Programmierung in IL.
PCD8.H31	Commissioning Tool in FUPLA

2.2 Elektrische Spezifikationen

Interne Leistungsaufnahme (ohne Encoder)

+5V 125 mA typisch, 150 mA max.
Uext 10 mA typisch, 15 mA max

Externe Speisung

Klemmen +/-: 24V (19 ... 32 V) geglättet,
zulässige Welligkeit max. 10%

Digitale Eingänge

4 bzw. 6 (siehe Abschnitt 2.1)

Analoger Ausgang

1 (siehe Abschnitt 2.1)

Wichtig :

Der maximale Strom, welcher auf den 5V geliefert werden kann, ist 1600 mA bei einer PCD2 bzw. 750 mA bei einer PCD1.

Der Anwender der Module PCD2.H310 und/oder PCD2.H311 ist angehalten, den Gesamtstromverbrauch **aller** Module einer PCD2/1 **und** in einem eventuellen Erweiterungsgerät C100 oder C150 zu überprüfen und dafür zu sorgen, dass das Maximum nicht überschritten wird. (Der Strom, welcher zur Speisung der 5V-Encoder geliefert wird, entstammt auch dieser Quelle und muss ebenfalls berücksichtigt werden).

Falls mit einem Erweiterungsgerät gearbeitet wird, ist darauf zu achten, dass die PCD2.H31x-Module im Basisgerät plaziert und im Erweiterungsgerät nur "normale" E/A-Module eingesteckt werden, da sonst der Spannungsabfall auf dem Verbindungskabel zu grosse Werte annehmen könnte.

2.3 Funktionsspezifische Daten

Anzahl Systeme 1

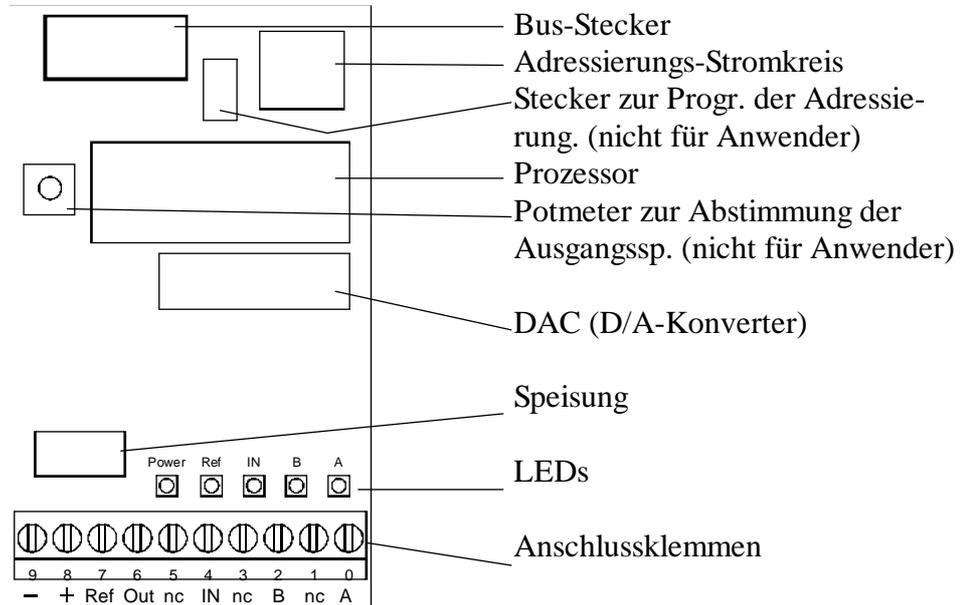
Bewegungsparameter

(für die Zielposition, Geschwindigkeit und Beschleunigung werden Register zu 31 Bit verwendet, Zahlenbereich $\pm 2^{30}$)

Position	Auflösung wählbar (vom Maschinen-Faktor abhängig)
Geschwindigkeit	Auflösung wählbar (vom Maschinen-Faktor abhängig)
Beschleunigung	Auflösung wählbar (vom Maschinen-Faktor abhängig)
PID-Regler	Abtastzeit 341 μ s, programmierbare Proportional-, Integral- und Differentialfaktoren, Abtastzeit für Differentialteil getrennt programmierbar
Zählfrequenz	max. 50 kHz

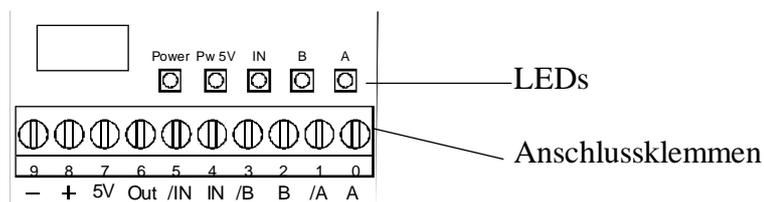
3. Präsentation

Bestücktes Modul mit Anschlussklemmen PCD2.H310 (24V-Encoder)



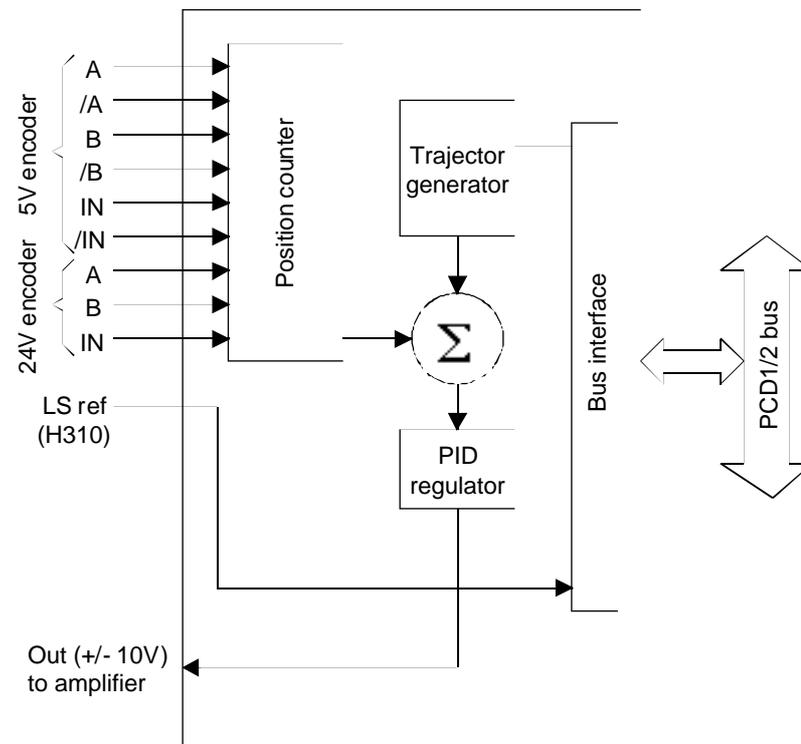
- und + sind die Klemmen für die externe Speisung: V_{ext}
- Ref** ist der digitale Eingang für den Referenz-Schalter
- Out** ist der analoge Regler-Ausgang
- A, B, IN** sind die 3 Signale des Encoders
- nc** sind nicht verwendete Klemmen (not connected)

Bestücktes Modul mit Anschlussklemmen PCD2.H311 (5V-Encoder)



- und + sind die Klemmen für die externe Speisung: V_{ext}
- 5V** ist der Ausgang für die 5V-Speisung des Encoders (300 mA max.)
- Out** ist der analoge Regler-Ausgang
- A, B, IN** sind die 3 nicht-invertierten Signale des Encoders
- /A, /B, /IN** sind die 3 invertierten Signale des Encoders

Einfaches Blockschaltbild

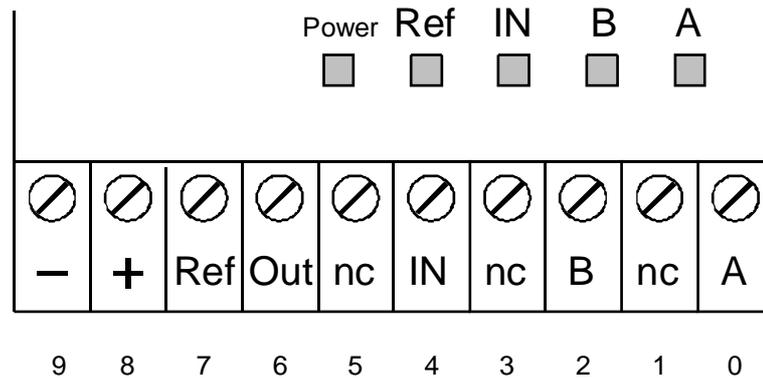


Andere achsenpezifische Ein- und Ausgänge, wie z.B. Limit Switches, der Referenzschalter für das Modul PCD2.H311, Enable für den Driver, müssen mit einem Standard-I/O-Modul (z.B. PCD2.B100, PCD2.E110/E111 bzw. PCD2.A410) überwacht resp. gesteuert werden.

4. Anschlüsse und Bedeutung der LEDs

Anschlussklemmen und LEDs des PCD2.H310 (24V Encoder)

Das Bild zeigt die Beschriftung der Leiterplatte. Der E/A-Steckerblock ist standardmässig von 0 .. 9 nummeriert (von rechts nach links)



Eingänge:

Anzahl		4
Klemme 0 =	"A":	Encoder-Signal "A"
Klemme 1 =	nc:	nicht verwendet *)
Klemme 2 =	"B":	Encoder-Signal "B"
Klemme 3 =	nc:	nicht verwendet *)
Klemme 4 =	"IN":	Encoder-Signal "IN"
Klemme 5 =	nc:	nicht verwendet *)
Klemme 7 =	"Ref":	digitaler Eingang für den Referenzschalter

Ausgang:

Klemme 6 =	"Out":	analoger Regler-Ausgang
------------	--------	-------------------------

Speisung:

Klemme 8 =	+	+ 24 VDC, geglättet
Klemme 9 =	-	GND

*) nicht verdrahten - darf nicht als Stützpunkt verwendet werden

Bedeutung der LEDs

LED **Ref** zeigt den logischen Zustand des Referenz-Schalters.
Die LED ist "ein", wenn eine Referenz verlangt wird oder wenn der Schalter nicht verdrahtet ist. (Nur beim PCD2.H310)

LED **Pw5V** ist das Abbild der Speisung des angeschlossenen Encoders.
Die LED ist "ein", wenn die 5V in Ordnung sind (kein Kurzschluss). (Nur beim PCD2.H311)

LED **Power** zeigt die Anwesenheit der $\pm 15V$.
Die LED ist "ein", wenn beide 15V-Speisungen in Ordnung sind.

LEDs **A, B** sind das Abbild der Encoder-Eingänge.
Die LEDs sind "ein" beim logischen Zustand "H".

LED **IN** ist das Abbild des Index-Eingangs des Encoders.
Die LED ist "ein" beim aktiven Zustand (negative Logik beim H310).



Achtung: Auf diesem Modul befinden sich Bauteile, welche bezüglich elektrostatischen Entladungen empfindlich sind!

5. Funktionsbeschreibung

5.1 Betriebsarten

Grundsätzlich werden zwei verschiedene Betriebsarten unterschieden :

- **positionsgeregelter Betrieb**
- **drehzahlgeregelter Betrieb**

Positionsgeregelter Betrieb (Einleitung mit dem Befehl: 'StartMot')

Die Positionierung erfolgt nach folgendem Befehlsschema:

1. Position- und Parametereingabe für das Geschwindigkeitsprofil
2. Positionierung starten
3. Signal abwarten für "**Zielposition erreicht**".

Im Positionierbetrieb wird nach Eingabe verschiedener Parameter (PID-Faktoren, Beschleunigung, Geschwindigkeit usw.) eine vorgegebene Zielposition geregelt angefahren, wobei Geschwindigkeit, PID-Faktoren und Zielposition während der Bewegung verändert werden können.

Drehzahlgeregelter Betrieb

(Einleitung mit den Befehlen 'GoForw' oder 'GoBackw')

Befehlsschema :

1. Parametereingabe für das Geschwindigkeitsprofil
2. Bewegung starten
3. Bewegung abbrechen durch eingeben eines **Stopbefehls**

Im drehzahlgeregelten Betrieb wird mit der definierten Beschleunigung bis zur Sollgeschwindigkeit beschleunigt. Es wird geregelt mit dieser Geschwindigkeit gefahren bis ein Stopbefehl erfolgt. Die Sollgeschwindigkeit kann während der Bewegung verändert werden.

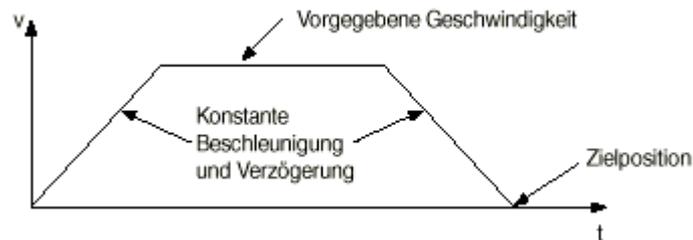
Funktionseinheiten

Aus dem Blockschaltbild (Seite 3-2) ist ersichtlich, dass das Positioniermodul im Wesentlichen aus den folgenden Funktionseinheiten besteht:

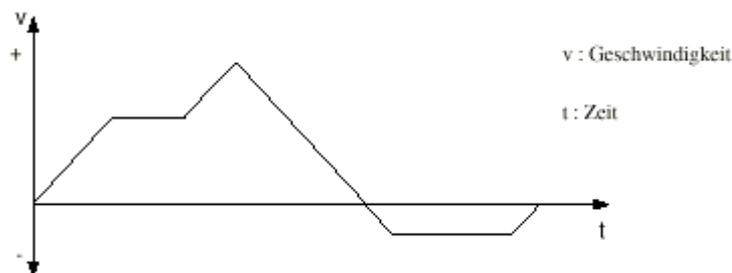
- Generator für Geschwindigkeitsprofil
- PID-Regler
- Positionsdecoder und Eingangsschaltung
- Businterface (CPLD) zum PCD-Bus
- D/A-Wandler für die analoge Stellgröße

5.2 Generator für das Geschwindigkeitsprofil

Entsprechend der vorgegebenen Beschleunigung und Geschwindigkeit berechnet der Profildgenerator die Sollgeschwindigkeit in Funktion der Zeit im Positionier- und Drehzahlmodus. Im Positionierbetrieb wird während der Bewegung die Differenz von Soll- und Ist-Position dem PID-Regler zugeführt. Damit wird eine sehr genaue Positionierung der Bewegung erreicht.



Standard Geschwindigkeitsprofil



Geschwindigkeitsprofil mit geänderter Sollgeschwindigkeit und Zielposition während der Bewegung.

Die Geschwindigkeit und die Zielposition können an einer beliebigen Position während der Bewegung verändert werden und der Controller wird entsprechend mit der definierten Beschleunigung beschleunigen oder bremsen. Die Beschleunigungs- und die Bremsrampe sind symmetrisch.

Im drehzahlgeregelten Betrieb beschleunigt der Controller bis zur vom Anwender definierten Sollgeschwindigkeit und fährt mit konstanter Geschwindigkeit, bis ein Stopbefehl erfolgt.

Funktionsprinzip der Drehzahlregelung:

Die Sollposition wird kontinuierlich (entsprechend der verlangten Geschwindigkeit) vergrößert. Die Differenz aus Soll- und Ist-Position (welche mit dem Encoder erfasst wird) wird wiederum dem PID-Regler zugeführt. Dieser kompensiert sofort Geschwindigkeitsschwankungen, verursacht durch irgendwelche Störeinflüsse, durch vergrößern oder verkleinern der Stellgröße.

Falls der Motor die Sollgeschwindigkeit nicht erreicht (z.B. durch blockierten Rotor), so wird die Differenz zwischen Soll- und Ist-Position sehr gross. Dies erzeugt eine Positionsfehlermeldung, welche einen automatischen Stop des Motors auslösen kann. Der Wert für den maximal zulässigen Positionsfehler ist programmierbar.

5.3 PID-Regler

Mit Hilfe des PID-Reglers kann der Motor die Zielposition genau anfahren und wird in dieser Position gehalten, da der Regler solange aktiv ist bis ein Stopbefehl erfolgt.

Der Controller benutzt folgenden Regelalgorithmus:

$$U(n) = k_p * e(n) + k_i * \sum_{N=0}^n e(n) + k_d * [e(n') - e(n'-1)]$$

wobei: $U(n)$ → Stellgrösse für den Motor
 $e(n)$ → Positionsfehler bei der n'ten Abtastung
 n → Abtastung für den Integralteil
 n' → Abtastung für den Differentialteil
 k_p → Proportionalfaktor
 k_i → Integralfaktor
 k_d → Differentialfaktor

Parameter, welche vom Anwender programmiert werden können:

- Regelfaktoren k_p , k_i , k_d
- Differentialabtastzeit
- Integrationsgrenze (IL) für den Integralanteil

Die **Regelfaktoren k_p , k_i und k_d** dürfen während einer Bewegung verändert werden.

Die **Abtastzeit für den Proportional- und Integralteil** beträgt $341\mu\text{s}$. Das bedeutet, dass die Stellgrösse in einem Intervall von $341\mu\text{s}$ aufgefrischt wird.

Die **Abtastzeit vom Differentialteil** kann in Schritten von $341\mu\text{s}$ eingestellt werden (max. $256 * 341\mu\text{s}$). Für den Betrieb mit langsamen Geschwindigkeiten sollte eine grössere Abtastzeit gewählt werden.

Integrationsgrenze IL : begrenzt wird der Betrag vom Ausdruck

$$k_i * \sum_{N=0}^n e(n)$$

5.4 Positionsdecoder und Eingangsschaltung

Positions- und Geschwindigkeitserfassung

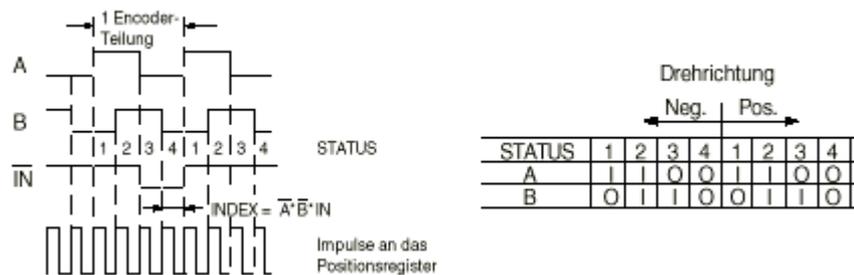
Die genaue Position und/oder Drehzahl des Motors wird mit einem Inkremental-Drehencoder erfasst. Folgende Encodersignale können angeschlossen werden:

PCD2.H310: A, B, IN (Klemmenanschlüsse)
24V-Signale im Quellbetrieb

PCD2.H311: A, /A; B, /B; IN, /IN (Klemmenanschlüsse)
5V RS422 Eingänge (Antivalent Line Driver)

Eingänge A, B, /IN :

Zustandsdiagramm der Signale A, B, /IN am Positionsdecoder :



Eingänge A, B:

Auf jeden Zustandswechsel ($0 \rightarrow 1$ und $1 \rightarrow 0$) der Signale A und B wird das interne Positionsregister um 1 erhöht oder erniedrigt. Dadurch wird die vierfache Auflösung der Encoderteilung erreicht. Die Eingabe für die Zielposition muss dementsprechend auch mit vier multipliziert werden, wenn mit Encoderimpulsen gearbeitet wird.

Für den Positionsdecoder müssen die Signale genau die Folge aufweisen wie im Bild oben dargestellt.

Eingang IN:

Bei den Modulen für 24V-Encoder (Typ H310) kann der Eingang IN als Indeximpuls- (Nullsignal vom Encoder) oder als Referenzpunkt-Eingang verwendet werden.

- Verwendung als Indeximpulseingang:
Jedesmal, wenn alle 3 Encodersignale den Zustand Null haben und vorher der Funktionsblock "SetIdxPos" (Set Index Position) aufgerufen wurde, wird die absolute Bewegungsposition in das Index-Positionsregister geschrieben .
- Verwendung als Referenzpunkteingang:
Es kann ein Referenzschalter angeschlossen werden, um z.B. die Nullposition zu definieren.

Module für den Anschluss von 5V Encodersignalen

Es kommt das Modul PCD2.H311 zur Anwendung. Für die Verbindung zum 5V-Encoder muss ein abgeschirmtes Kabel verwendet werden:

- max. Kabellänge: 20m
 - min. Leiterquerschnitt: 0.25 mm²
- z.B. Kabel PCD2.K271 oder PCD2.K273

Referenzschalter

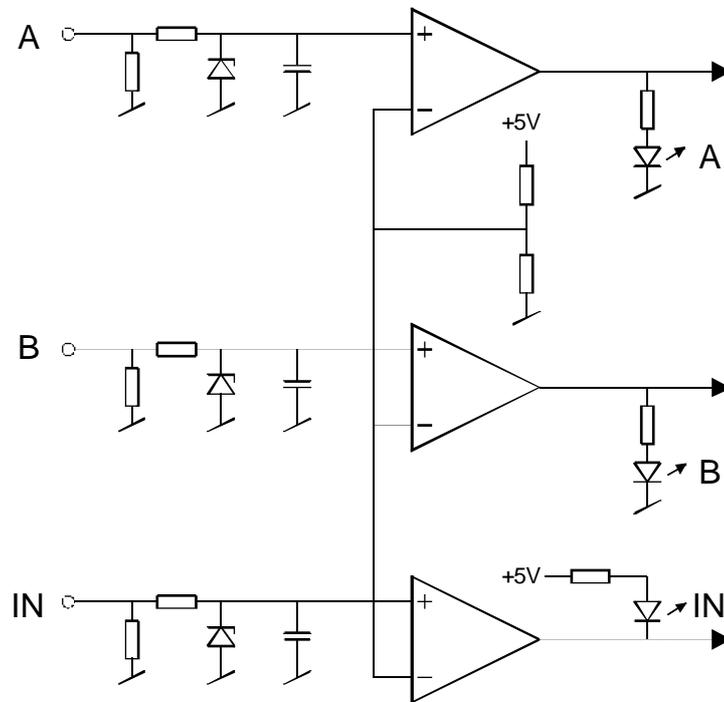
Nur das PCD2.H310 hat den Eingang 'Ref' (24 VDC, Quellbetrieb). Das Signal dieses Schalters wird direkt auf den PCD2-Bus geführt. Das bedeutet, dass der Eingang vom Anwenderprogramm aus zu überwachen ist, um die notwendigen Aktionen auszulösen.

Beim PCD2.H311 ist ein eventueller Referenzschalter auf ein "normales" digitales E/A-Modul zu verdrahten.

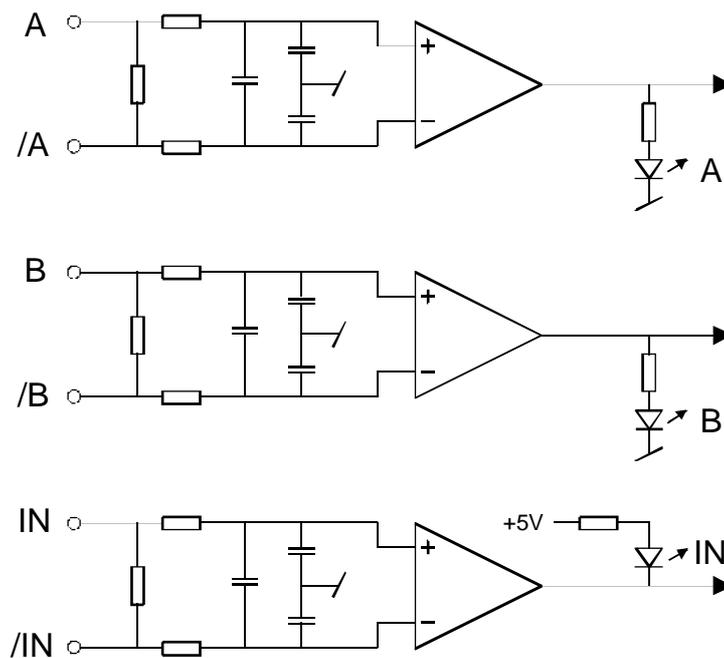
Der 'Referenzschalter' wird in Kombination mit den 'Limit Switches' beim FB 'Home' verwendet. Die Limit Switches sind bei beiden Modulen (H310 und H311) auf ein digitales E/A-Modul zu führen.

Eingangsschaltbilder und Anschlüsse

PCD2.H310



PCD2.H311

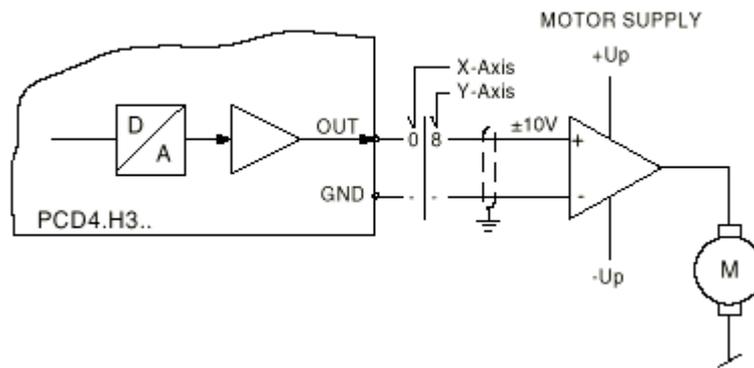


5.5 D/A-Wandler

Beide Module PCD2.H310 und H311 verfügen über einen analogen Ausgang für die Motorstellgröße.

Es ist ein 12 bit D/A-Wandler eingesetzt.

Analogausgang-Anschluss:

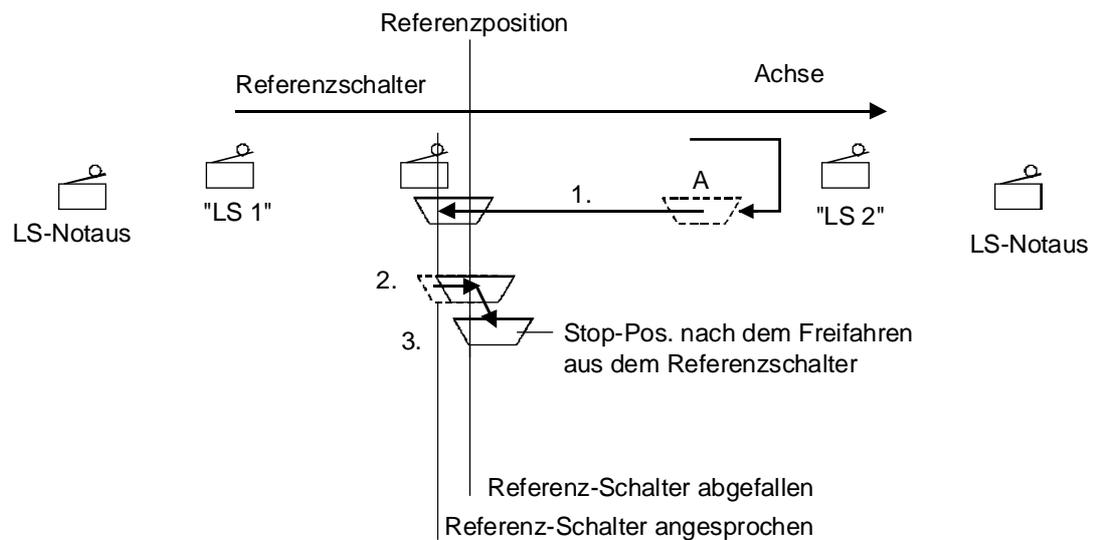


5.6 Ergänzende Information: Referenzfahren (FB Home)

Mit dem FB 'Home' kann die Referenzfahrt eigenständig ausgeführt werden. Zur Definition der Referenzfahrt werden 7 Parameter benötigt. Die Verwendung des FB 'Home' ist im Abschnitt 12.1. beschrieben.

Die zu referenzierende Achse muss initialisiert sein (FB Init). (Die nachfolgende Beschreibung bezieht sich auf Bild auf der folgenden Seite). Für eine erfolgreiche Anwendung des FB 'Home' müssen sich die Achse und der Referenz-Schalter zwischen dem 'LS1' und dem 'LS2' befinden.

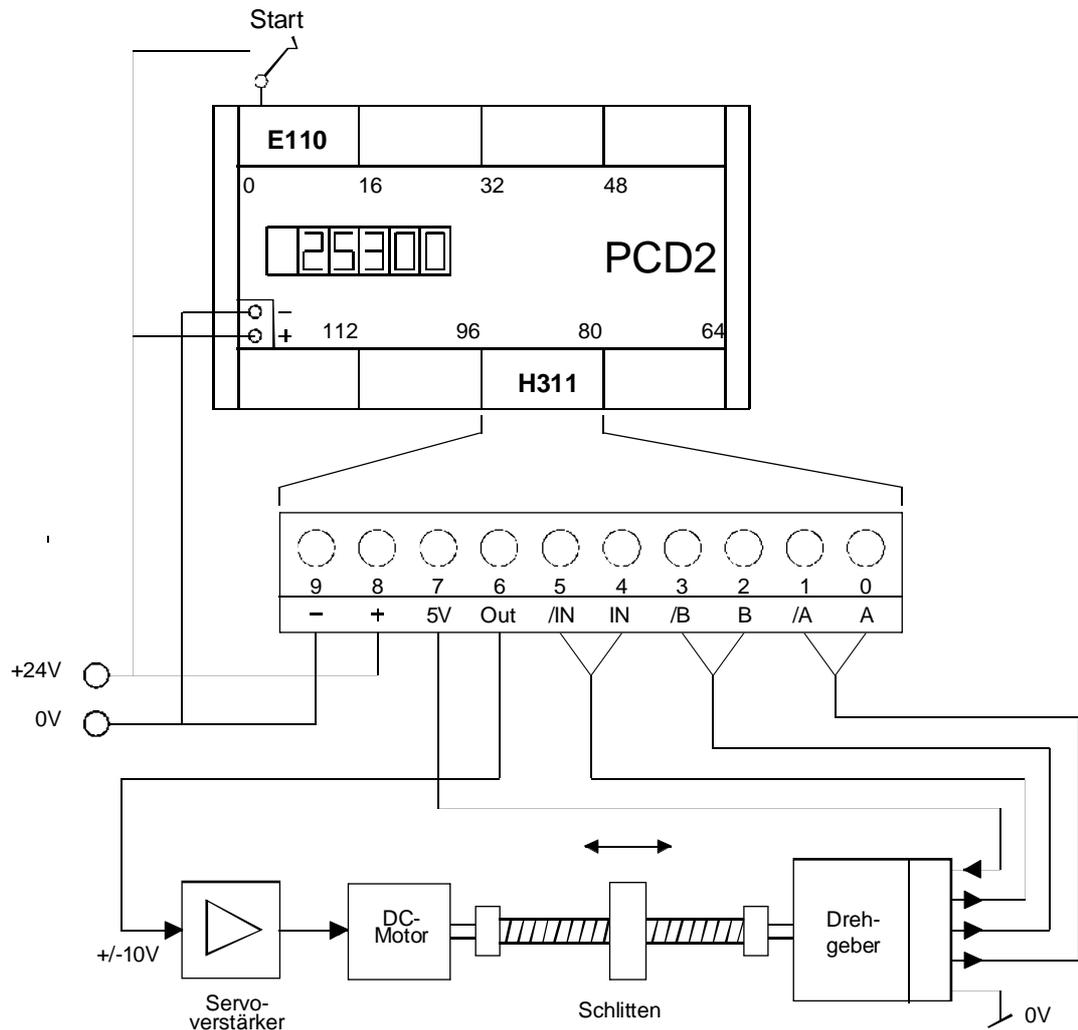
1. Die Suche des Referenzschalters wird mit der im Parameter 5 (Vmax) definierten Geschwindigkeit durchgeführt. Im Parameter 2 wird die Suchrichtung definiert. Wird der Referenzschalter nicht gefunden und trifft die Achse auf einen Endschalter, wird die Suchrichtung invertiert. Trifft die Achse auf den 2. Endschalter ohne den Referenzschalter gefunden zu haben, wird die Referenzfahrt abgebrochen und das Fehlerflag 'fHomeErr_x' gesetzt.
2. Der digitale Eingang, an welchem der Referenzschalter verdrahtet ist, wird im Parameter 7 definiert. Für ein PCD2.H310.Modul ist diese Adresse die Basisadresse des Moduls plus 11. Für ein PCD2.H311-Modul kann die Adresse frei gewählt werden ("normales" digitales Eingangsmodul).
3. Ist der Referenzschalter gefunden, beginnt das Freifahren. Die Richtung des Freifahrens ist in Parameter 3 definiert, die Geschwindigkeit für das Freifahren ist Vmin (Parameter 4).
4. Nachdem der Referenzschalter abgefallen ist, fährt die Achse bis zum Erreichen des nächsten Indexsignals des Encoders weiter und stoppt. Diese Position ist nun als Nullposition definiert.
5. Das Modul ist mit den ursprünglichen Einstellungen (aus dem FB Init) konfiguriert und der FB Home wird verlassen.



Hinweise:

- Es können mehrere Module gleichzeitig auf der gleichen PCD referenziert werden.
- Das Flag 'fEndHome_x' muss für den Start der Homingprozedur = L sein.
- Die Flags 'fLS1_x' und 'fLS2_x' sind das Abbild der Limit Switches. Der logische Zustand der beiden Flags ist das Resultat der Abfrage der beiden digitalen Eingänge, an welche die LS angeschlossen sind.
- Es sind grundsätzlich Öffnerkontakte für alle Schalter zu verwenden.
- Wird kein Referenzschalter gefunden, wird das Fehlerflag 'fHomeErr_x' (x = Modul Nr.) gesetzt und der FB 'Home' verlassen.
- Ist der FB 'Home' beendet (erfolgreich oder wegen eines Fehlers abgebrochen), wird das Flag 'fEndHome_x' automatisch gesetzt.
- Der FB 'Home' kann mit einem Timeout (Parameter 6) abgebrochen werden. Der Wert entspricht der Zeit in Sekunden, nach welcher der Abbruch erfolgt. Zusätzlich zum Fehlerflag 'fHomeErr_x' wird in diesem Fall das Diagnoseregister 'rDiag' mit dem Code 6 (für Parameter 6) im 3. Byte (für FB Home) geladen. Der Parameter Check erfolgt wie in Kapitel 8 beschrieben.

6. Der schnelle Einstieg



Minimal-Anordnung für eine Positionssteuerung mittels einem PCD2.H310 bzw. einem PCD2 H311, ohne Einbezug von Referenz- und Endschaltern.

Die einzelnen Elemente sind:

- PCD1 oder PCD2, mindestens bestückt mit

1 PCD2.H310/311
mit Display F510/530
1 PCD2.E110
- Bewegungsmodell mit DC-Motor, Spindel und Inkremental-Drehgeber

DC-Motor mit Getriebe:	ca. 500 U/min bei 10 VDC
Steigung der Spindel:	1 mm/Umdrehung
Inkrementaldrehgeber:	1000 Signale pro Umdrehung
- 4-Quadrant-Servo-Verstärker
z.B. mit Op-Amp LM 12 (National Semiconductor)
Details siehe: <http://www.national.com/pf/LM/LM12.html>
- PG4 ab Version V2.0.70 und FBs PCD9.H31E

6.1 Einstieg mit Programmierung in IL

Um auf einfachste Weise eine geregelte Positionssteuerung in Betrieb zu nehmen, sei das nachfolgend gezeigte Minimal-Programm vorgeschlagen:



Ein sauberes Anwenderprogramm soll keine Warteschleifen enthalten. Das vorliegende 1. Beispiel wurde, zum Aufzeigen der wesentlichen Befehle zur Ansteuerung eines PCD2.H31x, trotzdem mit Warteschleifen gestaltet. In der Praxis ist für diese Art von Programmen immer eine GRAFTEC- oder in Zukunft eine FUPLA-Struktur zu wählen. (Siehe 2. Beispiel und Kapitel 10).

Aufgabe: Nach dem Einschalten des PCD-Eingangs 'Start' soll der Schlitten in die eine und nach einer Pause wieder zurück in die Ausgangsposition gefahren werden. Es wird davon ausgegangen, dass beim Start der Schlitten etwa in der Mitte der Achse steht.

Es geht bei diesem Beispiel darum, das Prinzip und mögliche Strukturen des Anwenderprogramms aufzuzeigen. Es ist vor allem der Programmumfang mit den Assemblerdirektiven '\$include ' und '\$group' zu beachten.

Die Bedeutung und die Bestimmung der einzelnen Parameter verlangt einige Kenntnisse in Regeltechnik und wird im Kapitel 9: 'Installation und Inbetriebnahme' erläutert.

Das 1. Beispiel (mit Warteschleifen) hat den Namen 'Intro-1.src'. Das nachfolgend gezeigte gleiche Beispiel in GRAFTEC hat den Namen 'Intro-2.sfc'

Die FBs (IL für PG4 ab Version V2.0.70) sind auf der Diskette PCD9.H31E. Um die FBs auf den PC zu installieren, ist den Anweisungen im nachfolgenden Kapitel 7 und der Datei README.TXT zu folgen. Diese Datei liegt auch auf dieser Diskette.

Die Anzahl Module (1) und die Adresse des PCD2.H311-Modules (Adresse 80) muss in die Datei D2H310_B.MBA eingetragen werden:

```
NbrModules EQU 1 ; No. of H31x modules used (0...16)
```

```
BA_1 EQU 80 ; Base address of module 1
```

Diese Datei (D2H310_B.MBA) muss im Projektverzeichnis dieses Beispiels liegen, d.h. diese Datei ist ab der Diskette von Hand ins aktuelle Projektverzeichnis zu kopieren.

6.1.1 Einsteiger-Beispiel in IL mit Warteschleifen: INTRO-1.SRC

```

$include D2H310_B.equ
$group H310

    xob    16

    LD     R 1000
           4.0      ; Mecanic factor
    LD     R 1001
           8000     ; Initial absolute speed
    LD     R 1002
           10000    ; Initial absolute acceleration

    CFB    Init      ; Initialization FB
    K 1    1         ; Module number
           250      ; Proportional factor (regulator)
           0        ; Intergrative factor (regulator)
           0        ; Derivative factor (regulator)
           4000     ; Integrative limit value
           5        ; Derivative term sampling interval
           500     ; Position tolerance
           0        ; Behavior in case of position error
    R 1000 1000     ; Mecanic factor register
    R 1001 1001     ; Initial velocity register
    R 1002 1002     ; Initial acceleration register

    exob

; -----

    cob    0
           0

start:  sth    i 0

    CFB    Exec      ; Executable FB
    K 1    1         ; Module number
           RdActPos ; Command: Read Actual Position
    R 90   90        ; Actual Position register
    DSP   R 90      ; Display register

    jr     l start   ; If Start is not done, wait

    LD     R 100     ; Target Position
           20000    ; Absolute value

    CFB    Exec      ; Executable FB
    K 1    1         ; Module number
           LdDestAbs ; Command: Load Absolute Destination
    R 100  100      ; Absolute Destination register

    CFB    Exec      ; Executable FB
    K 1    1         ; Module number
           StartMot  ; Command: Start motion
           rNotUsed  ; Dummy register

```

```

pos1:  CFB    Exec    ; Executable FB
        K 1      ; Module number
        RdActPos ; Command: Read Actual Position
        R 90    ; Actual Position register
        DSP    R 90 ; Display register

        CFB    Exec    ; Executable FB
        K 1      ; Module number
        RdStatRg ; Command: Read Status Register
        R 0      ; Value of Status Register

        STH    fOnDest_1 ; Position reached?
        jr    l pos1     ; if no - wait (loop)

        ld    t 0      ; load timer for pause
                50      ; 5 sec
pause1: sth    t 0      ; pause elapsed?
        jr    h pause1 ; if no - wait (loop)

        LD    R 100    ; Target Position
                0      ; Absolute value

        CFB    Exec    ; Executable FB
        K 1      ; Module number
        LdDestAbs ; Command: Load Absolute Destination
        R 100    ; Absolute Destination register

        CFB    Exec    ; Executable FB
        K 1      ; Module number
        StartMot ; Command: Start motion
        rNotUsed ; Dummy register

pos2:  CFB    Exec    ; Executable FB
        K 1      ; Module number
        RdActPos ; Command: Read Actual Position
        R 90    ; Actual Position register
        DSP    R 90 ; Display register

        CFB    Exec    ; Executable FB
        K 1      ; Module number
        RdStatRg ; Command: Read Status Register
        R 0      ; Value of Status Register

        STH    fOnDest_1 ; Position reached?
        jr    l pos2     ; if no - wait (loop)

        ld    t 0      ; load timer for pause
                50      ; 5 sec
pause2: sth    t 0      ; pause elapsed?
        jr    h pause2 ; if no - wait (loop)

        ecob

$endgroup

```

Beschreibung des Programms:

Mit der Direktive '\$include' wird die Datei 'D2H310_B.equ' eingebunden. (Diese Datei bindet wiederum die Datei 'D2H310_B.mba' mit den Angaben zur Anzahl H31x-Modulen und deren Basisadressen ein. Dies geht automatisch, der Anwender hat damit nichts zu tun.)

Die Direktive '\$group H310' deklariert den Programmcode bis zu '\$endgroup' als zum Modul PCD2.H31x gehörend.

Im XOB 16 (Kaltstart-Block) wird die Initialisierung des Moduls durchgeführt. Es sind vor dem FB INIT die 3 Register für den Maschinenfaktor, die Geschwindigkeit und die Beschleunigung zu laden. Die Wahl der einzelnen Werte wird im Kapitel 9: 'Installation und Inbetriebnahme' erläutert.

Es folgt der Aufruf des FB INIT. Auch die Wahl der 11 Parameter wird im Kapitel 9 beschrieben.

Im COB 0 erfolgt das eigentliche Bewegungsprogramm.

Es wird der Start mit dem PCD-Eingang 0 abgewartet. Damit die aktuelle Position auch in dieser Phase erfasst wird und auf den Display erscheint, wird der Befehl 'RdActPos' in diese Warteschleife eingebaut, so dass bei nicht erfüllter Startbedingung die Position dauernd gelesen wird.

Es wird die absolute Ziel-Position ins PCD-Register R 100 geladen. Mit 'LdDestAbs' wird dieser Wert ins Modul übernommen. Mit dem Befehl 'StartMot' wird die Bewegung gestartet. (20 mm Verschiebung)

In einer Programmschleife wird mit 'RdActPos' laufend die Position gelesen und via das PCD-Register R 90 zum Display gebracht. Die Bewegung wird gemäss den im FB INIT gewählten Parametern ausgeführt, d.h. die Bewegung wird optimal, gesteuert durch das Modul selbst ohne Einfluss des Anwenderprogramms, bis zum Zielpunkt gefahren. Damit der Programmablauf korrekt weiter geführt werden kann, ist zu ermitteln, wann die Bewegung abgeschlossen ist. Dies wird durch die Abfrage des Flags 'fOnDest_x' (fOnDest_1 für unsern Fall des Moduls Nr. 1) gemacht. Bevor jedoch dieses Flag abgefragt werden kann, muss dieses mit dem Befehl 'RdStatRg' aktiviert werden.

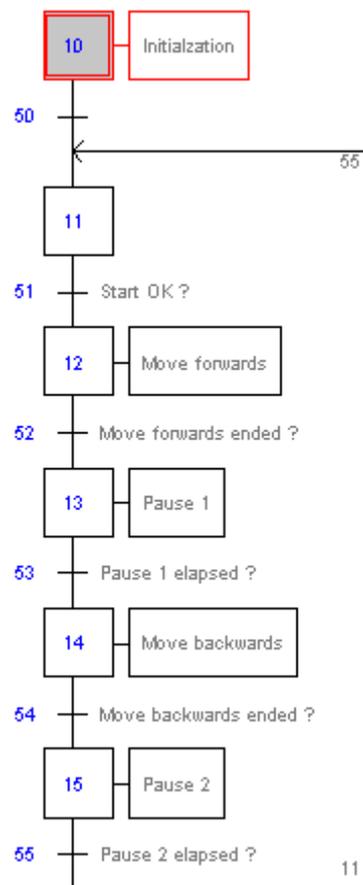
Die Programmschleife zum Lesen und zum Anzeigen der aktuellen Position, zur Aktivierung des Positionsflags sowie die Abfrage desselben, wird bis zum Erreichen der Ziel-Position ständig durchlaufen.

Ist die Ziel-Position erreicht, wird eine Pause von z.B. 5 Sekunden geladen und abgewartet. Danach wird die neue Ziel-Position (Null) geladen und die Bewegung zum Ausgangspunkt zurückgefahren.

Um auch in den Pausen die wirklich aktuelle Position zu kennen (Ausregelung der Endposition), müsste auch in die Warteschleifen für die Pausen der Befehl 'RdActPos' und 'DSP R 90' eingebaut werden.

6.1.2 Einsteiger-Beispiel in GRAFTEC: INTRO-2.SFC

Gleiches Beispiel wie unter 6.1.1, jedoch in sauberer GRAFTEC-Struktur ohne jegliche Programmsprünge und Warteschleifen. Die einzelnen Steps und Transitionen sind in IL editiert.



Der Code des Programms "intro-2.sfc"

(Um diese Darstellung zu erhalten, ist die Datei "intro-2.sfc in "intro-2.src" umzubenennen).

```

SB      0
;-----
IST     10          ;Initialization
        O 50

$include D2H310_B.equ
$group H310

LD      R 1000
        4.00        ; Mecanic factor
LD      R 1001
        8000        ; Initial absolute speed
LD      R 1002
        10000       ; Initial absolute acceleration

CFB     Init        ; Initialization FB
        K 1          ; Module number
        250         ; Proportional factor (regulator)
        0           ; Intergrative factor (regulator)
        0           ; Derivative factor (regulator)
        4000        ; Integrative limit value
        5           ; Derivative term sampling interval
        500        ; Position tolerance
        0           ; Behavior in case of position error
        R 1000      ; Mecanic factor register
        R 1001      ; Initial velocity register
        R 1002      ; Initial acceleration register
EST     ;10

;-----
ST      11
        I 50
        I 55        ;Pause 2 elapsed ?
        O 51        ;Start OK ?
EST     ;11

;-----
ST      12          ;Move forwards
        I 51        ;Start OK ?
        O 52        ;Move forwards ended ?

LD      R 100      ; Target Position
        20000      ; Absolute value

CFB     Exec        ; Executable FB
        K 1          ; Module number
        LdDestAbs   ; Command: Load Absolute Destination
        R 100       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1          ; Module number
        StartMot    ; Command: Start motion
        rNotUsed    ; Dummy register
EST     ;12

```

```

;-----
ST      13          ;Pause 1
        I 52          ;Move forwards ended ?
        O 53          ;Pause 1 elapsed ?

ld      t 0
        50
EST     ;13

;-----
ST      14          ;Movebackwards
        I 53          ;Pause 1 elapsed ?
        O 54          ;Move backwards ended ?

LD      R 100       ; Target Position
        0            ; Absolute value

CFB     Exec        ; Executable FB
        K 1          ; Module number
        LdDestAbs   ; Command: Load Absolute Destination
        R 100       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1          ; Module number
        StartMot    ; Command: Start motion
        rNotUsed    ; Dummy register
EST     ;14

;-----
ST      15          ;Pause 2
        I 54          ;Move backwards ended ?
        O 55          ;Pause 2 elapsed ?

ld      t 0
        50
EST     ;15

;-----
TR      50
        I 10          ;Initialization
        O 11
ETR     ;50

;-----
TR      51          ;Start OK ?
        I 11
        O 12          ;Move forwards

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdActPos    ; Command: Read Actual Position
        R 90        ; Actual Position register
DSP     R 90        ; Display register

sth     i 0

ETR     ;51

```

```

;-----
TR      52          ;Move forwards ended ?
        I 12        ;Move forwards
        O 13        ;Pause 1

CFB     Exec       ; Executable FB
        K 1         ; Module number
        RdActPos   ; Command: Read Actual Position
        R 90        ; Actual Position register
DSP     R 90        ; Display register

CFB     Exec       ; Executable FB
        K 1         ; Module number
        RdStatRg   ; Command: Read Status Register
        R 0         ; Value of Status Register

STH     fOnDest_1  ; Position reached?
ETR     ;52

;-----
TR      53          ;Pause 1 elapsed ?
        I 13        ;Pause 1
        O 14        ;Move backwards

stl     t 0
ETR     ;53

;-----
TR      54          ;Move backwards ended ?
        I 14        ;Move backwards
        O 15        ;Pause 2

CFB     Exec       ; Executable FB
        K 1         ; Module number
        RdActPos   ; Command: Read Actual Position
        R 90        ; Actual Position register
DSP     R 90        ; Display register

CFB     Exec       ; Executable FB
        K 1         ; Module number
        RdStatRg   ; Command: Read Status Register
        R 0         ; Value of Status Register

STH     fOnDest_1  ; Position reached?
ETR     ;54

;-----
TR      55          ;Pause 2 elapsed ?
        I 15        ;Pause 2
        O 11

stl     t 0

$endgroup
ETR     ;55

ESB     ;0

```

Erläuterungen zum Programm

Kenntnisse des PG4 im allgemeinen und des GRAFTEC im speziellen werden vorausgesetzt.

Der Sequentialblock SB 0 wird beim Assemblieren automatisch aus einem COB aufgerufen.

Der Ablauf des GRAFTEC-Programms kann online verfolgt werden.

Im IST 10 wird die Initialisierung des H310-Moduls vorgenommen. Der IST wird in der gewählten Anordnung, ähnlich dem XOB 16, nur beim ersten Aufruf des SB abgearbeitet. Es ist sinnvoll, die Initialisierung des H310-Moduls im IST desjenigen SB auszuführen, welcher das Modul behandelt, damit der ganze Programmteil beieinander bleibt. Im XOB 16 sollen vorzugsweise Initialisierungen ausgeführt werden, welche für die ganze PCD Gültigkeit haben.

Im ST 12 und ST 14 werden die absolute Ziel-Position via das PCD-Register R 100 ins H310-Modul geladen und jeweils der Start der Bewegung veranlasst.

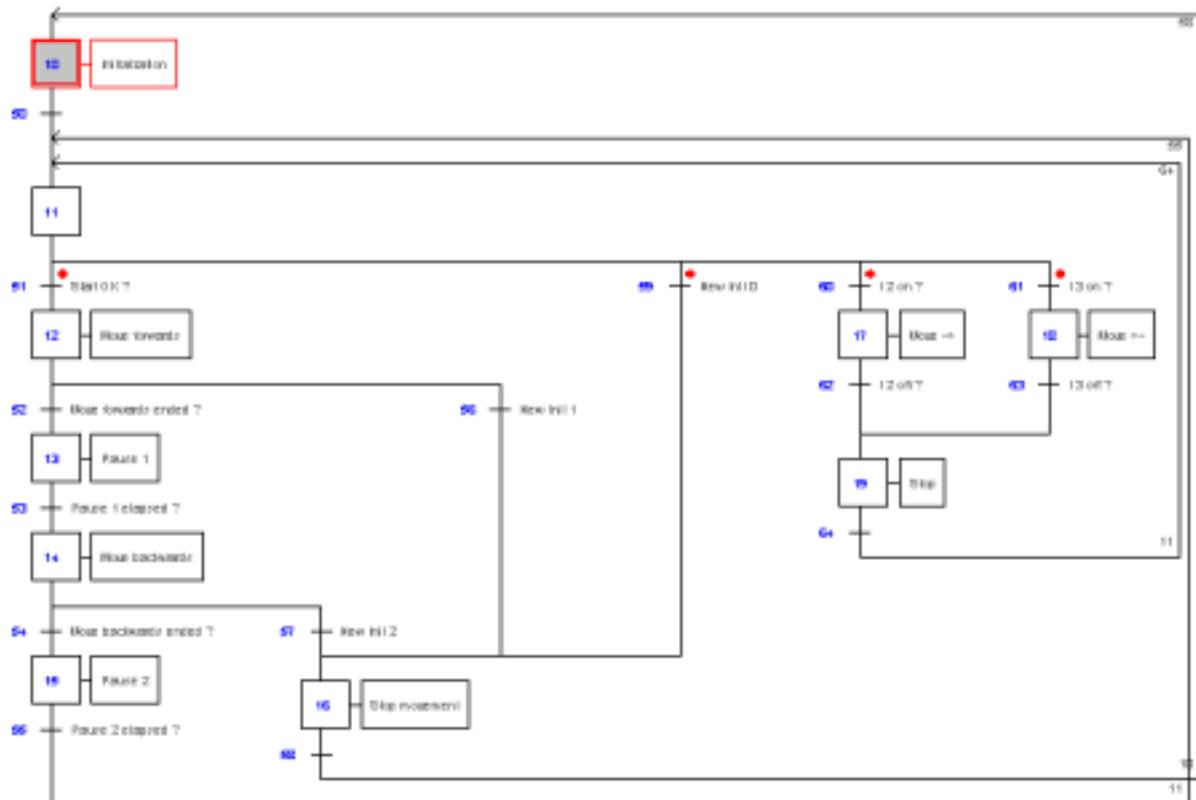
In den TR 52 und TR 54 wird softwaremässig das Ende der Bewegung abgefragt, um den weiteren Programmablauf frei zu geben. Die Bewegung selbst wird durch das Modul direkt gesteuert. Vor der Abfrage der Weiterschaltbedingung (sth fOnDest_1) wird die aktuelle Position gelesen und zum Display gebracht, sowie mit 'RdStatRg' das 'fOnDest_1'-Flag aufgefrischt. Gemäss den Regeln des GRAFTEC, wird das Programm bei jeder nicht erfüllten TR (z.B. Ziel-Position noch nicht erreicht) zum aufrufenden COB zurückkehren und dort weiter arbeiten. Beim nächsten Programmzyklus wird die nicht erfüllte TR wieder **ganz** abgearbeitet. Es ist so sichergestellt, dass die Position jedesmal automatisch gelesen und angezeigt wird, sowie das 'fOnDest_1'-Flag aufgefrischt wird.

Korrekterweise sollte in Versuchsanordnungen die Position auch während den Pausen gelesen und angezeigt werden, um so auch den Stand des Schlittens während dem Ausregeln verfolgen zu können.

6.1.3 Einfaches Inbetriebnahme-Programm *)

Zum Ausprobieren der verschiedenen Parameter zur Erreichung einer optimalen Bewegung, sei, abgeleitet vom vorangehenden Beispiel "intro-2.sfc", das nachfolgend gezeigte Programm "test-par.sfc" vorgeschlagen.

Funktion:



- I 0: Start der Hin- und Herbewegung (wie "intro-2.sfc")
- I 1: Übernahme von neuen Parametern, welche online im Debugger verändert wurden. Durch das Betätigen von I 1 kann eine Bewegung auch abrupt abgebrochen werden.
- I 2: Aus der Ruheposition kann der Schlitten mit den gewählten Parametern so lange vorwärts bewegt werden, wie I 2 eingeschaltet bleibt.
- I 3: Aus der Ruheposition kann der Schlitten mit den gewählten Parametern so lange rückwärts bewegt werden, wie I 3 eingeschaltet bleibt.

Das Ändern der Parameter kann online im Debugger erfolgen. Es ist mit <Display> <Program> <Step> <10> <CR> der IST anzuzeigen und ev. auszudrucken, um die absoluten Adressen der Parameter zu kennen.

- *) Ein komfortableres Inbetriebnahmeprogramm (Commissioning Tool) in FUPLA ist im Kapitel 10 vorgestellt.

```

SAIA PCD Online Debug
File Tools Options Help
Stn: None CPU: 0 Type: D2M1 007 Status: RUN
>Display Program St 10
000009 IST 10
000010 I 58
000011 O 50
000012 LD R 1000
000013 -2147483581
000015 LD R 1001
000016 8000
000018 LD R 1002
000019 10000
000021 CFB 0
000022 K 1
000023 250
000024 0
000025 0
000026 4000
000027 5
000028 500
000029 0
000030 R 1000
000031 R 1001
>
Run Stop Display Write Batch Clear rEstArt Locate Print File Help
cOnnect broAdcast Quit

```

```

LD R 1000
4.0 ; Mecanic factor
LD R 1001
8000 ; Initial absolute speed
LD R 1002
10000 ; Initial absolute acceleration

CFB Init ; Initialization FB
K 1 ; Module number
250 ; Proportional factor (regulator)
0 ; Intergrative factor (regulator)
0 ; Derivative factor (regulator)
4000 ; Integrative limit value
5 ; Derivative term sampling interval
500 ; Position tolerance
0 ; Behavior in case of position error
R 1000 ; Mecanic factor register
R 1001 ; Initial velocity register
R 1002 ; Initial acceleration register
EST ;10

```

Soll z.B. die Initialbeschleunigung (Initial absolute acceleration) von 10000 auf 30000 erhöht werden, ist die absolute Programmzeile 19 zu bearbeiten:

<Write> <Program> <19> <CR> <30000> <CR> <Esc>

Wenn Bewegung in Ruhe: PCD-Eingang I 1 ein- und ausschalten. Damit sind die geänderten Parameter übernommen worden. Durch das Einschalten von I 0 kann das neue Verhalten ausprobiert werden, u.s.w.

6.2 Einstieg mit Programmierung in FUPLA

In Vorbereitung

Notizen

7. Programmierung

Die Programmierung der PCD für den Einsatz der Zähl- und Positioniermodule PCD2.H... erfolgt über das PCD-Anwenderprogramm mittels der Standardprogrammierwerkzeuge PG4 ab Version V2.0.70. (Für die Verwendung des älteren Programmierwerkzeuges PG3 sind die FBs des Moduls PCD4.H3.. zu einzusetzen).

Die Programmierung erfolgt entweder in IL (Instruction List) mit FBs (Funktions Blocks) oder im FUPLA mit FBoxen (in Vorbereitung). Die FB sind auf Diskette unter der Bezeichnung PCD9.H31E erhältlich.

Da es sich bei Positionieraufgaben immer um sequentielle Abläufe handelt, werden Anwenderprogramme vorzugsweise in GRAFTEC programmiert, wobei die einzelnen Steps und Transitionen in IL mit FBs oder im FUPLA mit FBoxen editiert werden können. Anwenderprogramme können jedoch auch in reinem BLOCTEC oder in reinem FUPLA geschrieben werden.

7.1 Programmierung in IL mit FBs

7.1.1 Das IL-Paket (Installation der FB)

Die Bestellnummer der Diskette lautet PCD9.H31E. Die Diskette enthält die folgenden Verzeichnisse:

- APPSDIR : enthält alle Helps
- FB : enthält die .SRC- und .EQU-Dateien des H31x
- FBOX : enthält die FBoxen zum H31x
- PG3_FB : enthält alle Dateien der FB des PG3
- PG4_FB : enthält Beispiele und die .MBA-Datei
- Readme : enthält allgemeine Informationen

Das Paket ist für die Verwendung mit dem SAIA PG4 ab Version V2.0.70 vorgesehen sind. Für alle andern PG4-Versionen ist die Datei 'Readme' zu konsultieren. (Das Paket enthält auch FBs für die Verwendung mit dem älteren PG3, siehe 'Readme').

Die FBoxen für den FUPLA sind noch nicht verfügbar.

Installation des Pakets für das PG4

Die Installation wird am einfachsten mit dem PG4-Programm 'Setup Extra Files' durchgeführt:

Einfügen der Diskette PCD9.H31E ins Laufwerk A:

<Start> <Programs> <SAIA PG4> <Setup Extra Files>. Die FBs und die 'Help'-Datei werden auf der Harddisk ins Verzeichnis 'PG4' installiert.

Es werden die folgenden Dateien installiert:

D2H310_B.SRC	FB-Quell-Code	read-only Datei
D2H310_B.EQU	FB-Definitionen	read-only Datei

Diese 2 Dateien werden ab der Diskette ins PG4-Verzeichnis ...\\PG4\\FB kopiert.

FB_LIB.HLP	FB-Bibliotheks-Daten
D2H310_B.HLP	FB-Help-Datei

Diese Datei liegt im Verzeichnis A:\\APPSDIR und wird im PG4 ins Verzeichnis ...\\PG4 kopiert.

Die Datei **D2H310_B.MBA** (Modul-Basisadressen) muss ab der Diskette, aus dem Verzeichnis PG4_FB, **von Hand** in das jeweilige Projektverzeichnis kopiert werden.

Die für den Anwender wichtige Datei '**D2H310_B.MBA**' ist nachfolgend gezeigt:

Datei: **D2H310_B.MBA** (MBA = Modul Basis Adresse)

```

;
; This file can be modified by the user
;
; Basis addresses defined by the user
; -----
$group H310
NbrModules      EQU      1          ; No. of H310 modules used (0...16)
;
; Module base addresses (only the used modules must be defined)

BA_1            EQU      32         ;Base address of module 1
BA_2            EQU      0          ;Base address of module 2
BA_3            EQU      0          ;Base address of module 3
BA_4            EQU      0          ;Base address of module 4
BA_5            EQU      0          ;Base address of module 5
BA_6            EQU      0          ;Base address of module 6
BA_7            EQU      0          ;Base address of module 7
BA_8            EQU      0          ;Base address of module 8
BA_9            EQU      0          ;Base address of module 9
BA_10           EQU      0          ;Base address of module 10
BA_11           EQU      0          ;Base address of module 11
BA_12           EQU      0          ;Base address of module 12
BA_13           EQU      0          ;Base address of module 13
BA_14           EQU      0          ;Base address of module 14
BA_15           EQU      0          ;Base address of module 15
BA_16           EQU      0          ;Base address of module 16
$endgroup

```

Es sind die Anzahl PCD2.H31x-Module anzugeben. danach sind die Hardware-Basisadressen der verwendeten PCD2.H31x-Module einzutragen.

Da die '.mba'-Datei nicht im Projekt-Manager erscheint, muss zum Anpassen ein Texteditor, z.B. der SEDIT32, verwendet werden.

Die Module sind, beginnend mit 'BA_1', aufeinanderfolgend zu nummerieren. Werden z.B. 3 Stk. H310-Module in einem Projekt eingesetzt, so sind 'BA_1', 'BA_2' und 'BA_3' zu verwenden. Die Steckplätze der Module können beliebig zugewiesen werden. Beispiel:

```

NbrModules      EQU      3          ; No. of H310 modules used (0...16)
;
; Module base addresses (only the used modules must be defined)

BA_1            EQU      64         ;Base address of module 1
BA_2            EQU      112        ;Base address of module 2
BA_3            EQU      16         ;Base address of module 3
BA_4            EQU      0          ;Base address of module 4
BA_5            EQU      0          ;Base address of module 5

```

Die Basisadressen der Register, Flags und der FB werden automatisch vergeben und können in der Ressourcen Liste unter 'View' - 'Resource List' eingesehen werden..

Anordnung der Dateien und Vorgehen bei der Erstellung eines Anwenderprogramms. Das zu erstellende Projekt habe den Projektnamen "TEST-H3" und das eigentliche Anwenderprogramm den Namen "move-01.sfc".

```
C:\pg4 \FB          \D2H310_b.equ
          \D2H310_b.src
          \...
          \FBOX          \...
          \GALEP3        \...
          \PROJECTS     \FUP_E          (Demo Beispiel PG4)
          \GRAF_E        (Demo Beispiel PG4)
          \TEST-H3      \D2H310_b.mba
          \move-01.sfc
          \...
          \D2H310_b.hlp
```

Das Anwenderprogramm für den H310-Teil präsentiert sich folgendermassen :

```
$include D2H310_b.equ
$group H310

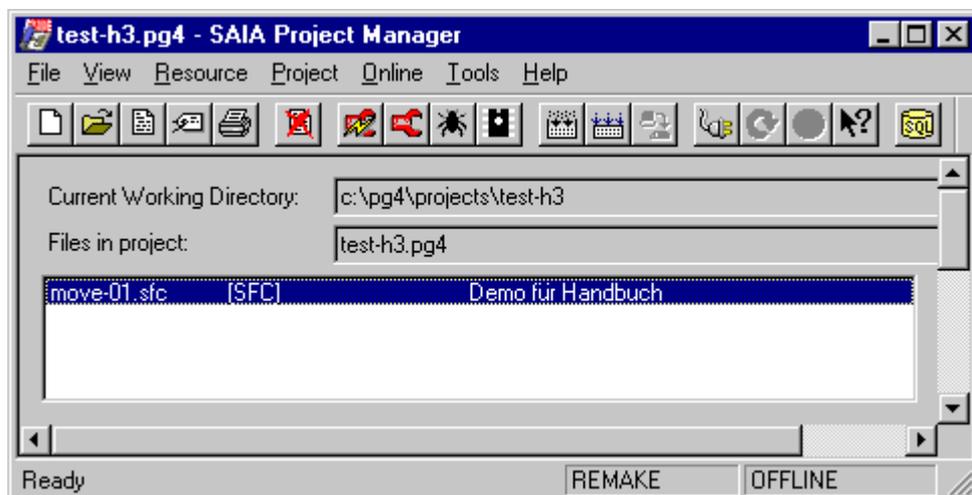
XOB      16

PCD-Code

ecob
$endgroup
```

Ist das Programm in GRAFTEC geschrieben, kommen die Assemblerdirektiven "\$include" und "\$group" in den 1. Step (ST), normalerweise den Initialstep (IST) zu liegen. "\$endgroup" kommt ans Ende der letzten Transition (TR).

Wurde alles korrekt installiert, das Anwenderprogramm editiert und alle Parameter definiert, kann mit 'Project' - 'Build' das Programm verarbeitet und in die PCD geladen werden.



7.1.2 Die einzelnen FBs

Das ganze Paket besteht grundsätzlich aus 2 (3) FBs mit Parametern:

- INIT Initialisierung FB mit 11 Parametern
- EXEC Ausführen FB mit 3 Parametern
- HOME Grundstellung FB mit 7 Parametern

Der Aufruf des FB "INIT" präsentiert sich immer wie folgt:
(die eingetragenen Werte sind nur als Beispiele gedacht)

```
CFB      init      ; Intitialisierung eines PCD2.H31x-Moduls
k 1      ; Par. 1: Modul-Nummer (k 1 - k 16)
250     ; Par. 2: Propotionlafaltor
150     ; Par. 3: Integralfaktor
10      ; Par. 4: Differentialfaktor
4000    ; Par. 5: Integrationsgrenze
5       ; Par. 6: Abtastzeit des D-Faktors
500     ; Par. 7: Positionstoleranz
0       ; Par. 8: Verhalten bei Positionsfehler
r 1000  ; Par. 9: Maschinenfaktor *)
r 1001  ; Par. 10: Geschwindigkeit *)
r 1002  ; Par. 11: Beschleunigung *)
```

*) Die Register der Paramter 9, 10 und 11 sind vor der Abarbeitung des FB INIT mit den korrekten Werten zu laden.

Der Aufruf des FB 'EXEC' präsentiert sich für einige typische Beispiele wie folgt:

```
CFB      exec
k 1      ; Par. 1: Modul-Nummer (k 1 - k 16)
LdDestRel ; Par. 2: Funktion (Befehl)
r 777    ; Par. 3: Wert (aus Quell-Register)
```

```
CFB      exec
k 1      ; Par. 1: Modul-Nummer (k 1 - k 16 )
StartMot ; Par. 2: Funktion (Befehl)
rNotUsed ; Par. 3: nicht verwendet
```

```
CFB      exec
k 1      ; Par. 1: Modul-Nummer (k 1 - k 16)
RdActPos ; Par. 2: Funktion (Befehl)
r 1000   ; Par. 3: Wert (in Ziel-Register)
```

Es müssen immer 3 Parameter angegeben werden, auch wenn für eine Funktion nur 2 verlangt sind. Als 3. Parameter kann 'rNotUsed' oder irgend ein Register angegeben werden.

Eine Liste mit allen Befehlen folgt auf der nächsten Seite.

Befehle (Funktionen) für den FB 'Exec' (Parameter 2):

No.	Symbol	Befehl (English)	Befehl (deutsch)	Seite
01	StartMot	Start Motion	Start der Bewegung (Positions-Modus)	A-7
02	StopUrg	Stop motion in Urgency	Notstop	A-8
03	Stop	Stop motion	Stoppen der Bewegung	A-9
04	MotOff	Motor regulation Off	Motorregelung ausschalten	A-10
05	RdActPos	Read Actual Position	Lesen der aktuellen Position	A-11
06	RdActVel	Read Actual Velocity	Lesen der aktuellen Geschwindigkeit	A-12
07	RdIntSum	Read Integration Sum	Lesen der Integrations-Summe	A-13
08	RdIndexRg	Read Index Register	Lesen des Index-Registers	A-14
09	RdStatRg	Read Status Register	Lesen des Status-Registers	A-15
10	RdTargPos	Read Target Position	Lesen der Zielposition	A-16
11	RdTargVel	Read Target Velocity	Lesen der Ziel-Geschwindigkeit	A-17
12	GoForw	Go Forwards	Bewegung vorwärts (Geschw.-Modus)	A-18
13	GoBackw	Go Backwards	Bewegung rückwärts (Geschw.-Modus)	A-19
14	SgStpFor	Single Step Forwards	Einzelschritt vorwärts	A-20
15	SgStpBak	Single Step Backwards	Einzelschritt rückwärts	A-21
16	LdDestAbs	Load Dest. Absolute	Laden der absoluten Ziel-Position	A-22
17	LdDestRel	Load Dest. Relative	Laden der relativen Ziel-Position	A-23
18	LdVelAbs	Load Velocity Absolute	Laden der absoluten Geschwindigkeit	A-24
19	LdVelRel	Load Velocity Relative	Laden der relativen Geschwindigkeit	A-25
20	LdAccAbs	Load Acc. Absolute	Laden der absoluten Beschleunigung	A-26
21	LdAccRel	Load Acc. Relative	Laden der relativen Beschleunigung	A-27
22	LdPropG	Load Proportional Gain	Laden des Proportionalfaktors	A-28
23	LdIntG	Load Integrative Gain	Laden des Integralfaktors	A-29
24	LdDerG	Load Derivative Gain	Laden des Differentialfaktors	A-30
25	LdSamplnt	Load derivative Sampling Interval	Laden der Abtastzeit	A-31
26	LdIntLim	Load Integrative Limit	Laden der Integrationsgrenze	A-32
27	ActRegFact	Activate Regulation Factors	Aktivieren der Regelfaktoren	A-33
28	LdBrkPtAbs	Load Breakp. Absolute	Laden eines absoluten Breakpoints	A-34
29	LdBrkPtRel	Load Breakp. Relative	Laden eines relativen Breakpoints	A-35
30	ResStatReg	Reset Status Register	Rücksetzen des Statusregisters	A-36
31	SetIdxPos	Set Index Position	Setzen der Indexposition	A-37
32	SetZero	Set Zero position	Setzen der Null-Position	A-38
33	MotConf	Motion Configuration	Konfigurieren Drehzahl- bzw. Positionsgeregelter Betrieb	A-39
34	SetPosTol	Set Position Tolerance	Setzen der Positionstoleranz	A-40

Die Zahl in der ersten Kolonne (0 - 34) ist der absolute Wert des Parameters Nr. 2 im FB 'Exec'. Mit Hilfe dieser Zahl kann, wenn das Anwenderprogramm im Debugger verfolgt wird, die Funktion des FB 'Exec' ge- deutet werden.

Der Aufruf des FB "HOME" präsentiert sich immer wie folgt:
(die eingetragenen Werte sind nur als Beispiele gedacht)

```
CFB      home      ; Initialisierung der Referenzposition
k 1      ; Par. 1: Modul-Nummer (k 1 - k 16)
1        ; Par. 2: Suchrichtung
0        ; Par. 3: Freifahrtrichtung
r 990    ; Par. 4: Minimale Geschwindigkeit
r 991    ; Par. 5: Maximale Geschwindigkeit
1000    ; Par. 6: Timeout
i 7      ; Par. 7: Referenzeingang
```

Für den Anwender abfragbare Elemente:

Element	Beschreibung
fHomeErr_x	Wenn Fehler mit 'Home', Element = H. (Timeout, Home-Position nicht gefunden)
fLS1_x	H, wenn bei Endschalter 1 eingetroffen
fLS2_x	H, wenn bei Endschalter 2 eingetroffen
fEndHome_x	Immer = H, ausser während Home-Prozedur
fBrkPt_x	H, wenn Breakpoint erreicht
fOnDest_x	H, wenn Zielposition erreicht
fPosErr_x	H, bei grossem Positionsfehler
fPar_Err	Parameter Error (ausserhalb Bereich)
fTimeout	Lesen/Schreiben (bei Hardwareproblem)
Ref_1	Abbild des Referenz-Eingangs

'_x' entspricht der Modul-Nummer

Die effektiven Adressen der Elemente sind der Ressourcenliste zu entnehmen: Aus Project Manager 'View' - 'Resource List'. Es erscheint eine ausführliche, umfangreiche Liste.

Symbol	Type	Address/Value	Scope	Auto	Comment	Module
H310.BA_3	K	0	Local	No	Base address of module 3	D2H3...
H310.BA_16	K	0	Local	No	Base address of module 16	D2H3...
H310.BA_15	K	0	Local	No	Base address of module 15	D2H3...
H310.BA_14	K	0	Local	No	Base address of module 14	D2H3...
H310.NbrModules	K	1	Local	No	No. of H310 modules used (0..16)	D2H3...
H310.NbrModules	K	1	Local	No	No. of H310 modules used (0..16)	2-spe...
H310.StartMot	K	1	Local	No	Start motion	D2H3...
__TIME_DIVFACT__	K	1	Public	No		~prop...
__TIME_MULFACT__	K	1	Public	No		~prop...
H310.StartMot	K	1	Local	No	Start motion	2-spe...
H310.StopUp	K	2	Local	No	Stop motion abruptly	D2H3...
H310.Stop	K	3	Local	No	Stop motion smoothly	D2H3...
H310.MotOff	K	4	Local	No	Motor Off	D2H3...
H310.NbrR	K	4	Local	No	No of reg. used by each module	D2H3...
__DYNAMICS__ FIR...	K	5	Public	No		~prop...

Eine übersichtliche Liste mit den absoluten Adressen, welche für das Debugging genügen sollte, liefert die .map-Datei des Projects:

SAIA PCD LINKER SP 2.0.83 FILE: test-2h3.pcd
LINKED: 06/30/99 10:57 PAGE 3

FOR SAIA'S INTERNAL USE ONLY

SYMBOL	TYPE	VALUE	DEFINED	REFERENCED
H310.Exec	FB	1	D2H310_B	2-speed
H310.fBrkPt_1	F	7517	D2H310_B	
H310.fEndHome_1	F	7516	D2H310_B	
H310.fHomeErr_1	F	7513	D2H310_B	
H310.Flag_base	F	7502..7528	D2H310_B	2-speed
H310.fLS1_1	F	7514	D2H310_B	
H310.fLS2_1	F	7515	D2H310_B	
H310.fOnDest_1	F	7518	D2H310_B	
H310.fPar_Err	F	7502	D2H310_B	
H310.fPosErr_1	F	7528	D2H310_B	
H310.fTimeout	F	7512	D2H310_B	
H310.Home	FB	2	D2H310_B	
H310.Init	FB	0	D2H310_B	2-speed
H310.rDiag	R	3500	D2H310_B	
H310.Reg_base	R	3500..3524	D2H310_B	2-speed
H310.rIniVel_1	R	3524	D2H310_B	
H310.rMecFac_1	R	3522	D2H310_B	
H310.rSampInt_1	R	3523	D2H310_B	

Linkage complete. 0 errors, 0 warnings.

Belegung der Adressen auf dem Bus

Jedes Modul belegt 16 Adressen als Eingänge (lesbar) und 16 Adressen als Ausgänge (scheibbar).

Bit-Nr.	DATA In (lesen)	DATA Out (scheiben)
0	Bus-Daten (LSB)	Bus-Daten (LSB)
1	Bus-Daten	Bus-Daten
2	Bus-Daten	Bus-Daten
3	Bus-Daten	Bus-Daten
4	Bus-Daten	Bus-Daten
5	Bus-Daten	Bus-Daten
6	Bus-Daten	Bus-Daten
7	Bus-Daten (MSB)	Bus-Daten (MSB)
8	Speisung *)	Schreiben (WR LM628)
9	-	Lesen (RD LM628)
10	-	Port select (PS LM628)
11	Ref. Switch **)/ Pw5V ***)	-
12	Version ****)	-
13 - 15	-	-

- *) Überwachung ±15V (H: Speisung OK; L: Speisung nicht OK).
- **) Ref Switch. Nur für H310. (H: RS aktiv; L: RS nicht aktiv).
- ***) Überwachung 5V. Nur H311. (H: 5V OK; L: 5V nicht OK).
- ****) Version des Moduls (H: H310; L: H311).

7.2 Programmierung in FUPLA mit FBoxen

in Vorbereitung

7.3 Programmierung in GRAFTEC mit FBoxen

in Vorbereitung

8. Fehlerbehandlung und Diagnose

8.1 Definitionsfehler durch Assembler überprüft

Folgende Definitionsfehler in der Datei D2H310_b.MBA werden während dem Assemblieren überprüft:

- Falls die Anzahl Module (NbrModules) < 1 ist, wird kein Code assembliert und die Warnung:

"Remark: No H310 used (NbrModules = 0 in D2H310_B.MBA)"

in das 'Make'-Fenster geschrieben.

- Falls die Anzahl Module (NbrModules) > 16 ist, wird kein Code assembliert und die Fehlermeldung:

"Error : more than 16 Modules H310 defined (NbrModules = 0...16)"

in das 'Make'-Fenster geschrieben.

- Falls ein falscher Befehlscode für den FB 'Exec' verwendet wird (z.B. RdIdenti anstatt RdIdent), meldet der Assembler einen Fehler:

"Symbol not defined 'H310.RdPosAct'"

(wobei der Ausdruck 'H310' vom \$group h310 generiert wird)

- Falls die Definition \$group H310 fehlt, meldet der Assembler:

"Symbol not defined"

für jeden Befehl und jedes Register/Flag welches im Programm verwendet wird.

8.2 Fehlerbehandlung in Run

8.2.1 Falsche Parameter

Im FB 'Exec' wird nur der Befehlscode überprüft. Der Parameter 1 (Modul-Nr.) sowie Parameter 3 (Quell-/Zielregister) werden nicht überprüft um die Ausführungszeit nicht zu verlängern.

In den FBs 'Init' und 'Home' werden alle Parameter auf deren Wert überprüft, ob diese im zulässigen Bereich liegen. Falls ein Parameter ausserhalb eines Bereichs liegt, wird dieser auf den minimalen Wert gebracht, das Fehlerflag 'fPar_Err' gesetzt und das Diagnoseregister 'rDiag' mit dem entsprechenden Fehlercode geladen.

Das Flag 'fPar_Err' wird innerhalb des FBs nicht zurückgesetzt, dies sollte im XOB 16 resp. im 'Init'-Step erfolgen.

Der Fehlercode setzt sich wie folgt zusammen:

```
rDiag  bit 31 . . . . . 24 23 . . . . . 16 15 . . . . . 8 7 . . . . . 0
        \ Reserve /   \ FB Nr. /   \ Par. Nr. /   \ Mod. Nr./
                               (Init = FB 1)
                               (Exec = FB 2)
                               (Home = FB 3)
```

Beispiel: Wird die Abtastzeit (Parameter 6) im FB 'Init' von Modul 2 falsch definiert (>255), so wird 'rDiag' mit dem Wert 00 01 06 02 hex geladen.

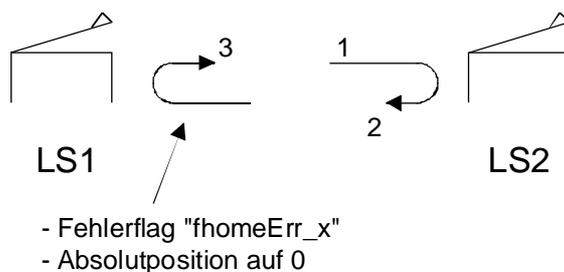
Das Diagnose-Register wird bei jedem falschen Parameter überschrieben und enthält immer den letzten Fehler. Es sollte deshalb ausgewertet werden, sobald das Flag 'fPar_Err' einen Bereichsfehler signalisiert. Die absoluten Adressen von 'rDiag' und 'fPar_Err' sind in der Datei 'project.MAP' ersichtlich (siehe Abschnitt 7.1.2, Seite 7-8). Dies kann bei der Inbetriebnahme mit dem Debugger zur Lokalisierung eines Fehlers nützlich sein:

- Run until flag 'fPar_Err' = H
- Display Register 'rDiag' hex
- Löschen des Flags 'fPar_Err'

8.2.2 Fehler während einer Referenzfahrt

Falls die Referenzposition nicht gefunden werden konnte (z.B. wegen eines defekten Referenzschalters), wird nach der erfolgten Betätigung der beiden LS das Fehlerflag 'fHomeErr_x' gesetzt, die Bewegung gestoppt, die Absolutposition auf 0 gesetzt und der FB 'Home' abgebrochen.

Referenzschalter nicht vorhanden oder falsch verdrahtet:



Falls der FB 'Home' wegen Überschreitung der spezifizierten Timeout-Zeit abgebrochen wurde, wird zusätzlich das Diagnoseregister 'rDiag' mit dem Code 6 als Parameter-Nr. geladen (Timeout ist Parameter 6).

Das Flag 'fHomeErr_x' ist pro Modul definiert (_x ist die Modul-Nr.) und wird am Anfang des FB 'Home' zurückgesetzt. Dieses Flag sollte nach jedem Aufruf des FB 'Home' abgefragt werden, um sicherzustellen, dass die Achse korrekt referenziert ist.

Beispiel:

```
CFB    Home                ; Referenzfahrt Achse 3
      k 3                  ; Module number
      0                    ; serach direction
      r 1010               ; min. speed
      r 1011               ; max. speed
      50                   ; timeout
      i 64                 ; input reference switch

STH    fHomeErr_3         ; Abfrage des Home Fehler-Flags
                        ; der Achse 3
CFB    h Errorhandl       ; Aufruf des (anwenderspezifischen)
                        ; FBs, falls das fHomeErr_3 = H ist

CFB    Exec                ; Bewegung 1
```

Notizen

9. Installation und Inbetriebnahme

9.1 Einleitung

Die folgende Beschreibung zeigt die Vorgehensweise für die Inbetriebnahme eines Servoantriebes mit dem Positioniermodul PCD2.H3... . Um einen einwandfreien Betrieb des H3-Moduls zu gewährleisten, sollten bei der Inbetriebnahme die in der Folge beschriebenen Schritte in der gleichen Reihenfolge ausgeführt werden.

Auswahlkriterien für einen Servoantrieb mit dem PCD2.H3.. - Modul

Eine Positioniereinheit besteht immer aus den folgenden Teilen:

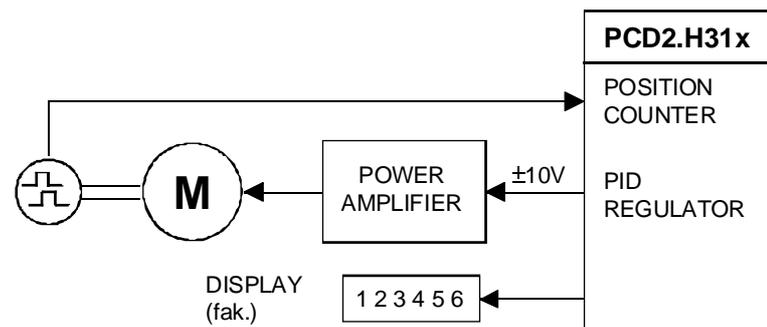
- Einer Positioniersteuerung zur Vorgabe der Positionierparameter (Position, Geschwindigkeit und Beschleunigung) und zur Lageregelung. Diese Aufgabe wird von der PCD1/2 mit dem H3-Modul wahrgenommen.
- Einem Servoverstärker zur Ansteuerung des Servomotors.
- Einem Servomotor zur Umsetzung der elektrischen in mechanische Energie.
- Einem Lagegeber, in der Form eines inkrementalen Drehgebers.
- Einer mechanischen Antriebseinheit.

Auswahl des Servoverstärkers und des Servomotors:

Unabhängig davon, ob ein DC- oder AC-Servoantrieb verwendet wird, sind die folgenden Punkte speziell zu beachten.

- Der Verstärker und der Motor müssen aufeinander abgestimmt sein (Leistung, Spannung und Strom).
- Für eine präzise Lage- und Drehzahl-Regelung ist ein Vier-Quadranten-Leistungssteller mit integriertem Drehzahlregler erforderlich.
- Möglichst grosser Drehzahlregelbereich des Verstärkers und Motors, damit auch bei kleinen Drehzahlen das erforderliche Drehmoment aufgebracht werden kann.
- Als Drehzahlsollwertvorgabe wird vom H3-Modul ein analoges $\pm 10V$ Signal geliefert.
- Das H3-Modul benötigt zur Erfassung der Position einen Inkrementaldrehgeber, welcher mindestens zwei um 90° phasenverschobene Rechtecksignale liefert.

Blockschema eines Positionierantriebes mit dem H3-Modul



9.2. Installation und Verdrahtung

Bei der Installation des PCD2-Systems sollten die nachstehend aufgeführten Punkte besonders beachtet werden. Bevor das System eingeschaltet wird, soll eine Sichtprüfung der Installation und der Verdrahtung wie folgt ausgeführt werden:

- Hat das gesamte PCD1/2-System keine Transport- oder Montageschäden erlitten?
- Ist das H3-Modul am vorgesehenen Platz auf dem PCD1/2-Bus gesteckt und ist die Verdrahtung am entsprechenden Busmodul ausgeführt?
- **Verdrahtung und Anschluss der Anwenderspeisung:**
Das H3-Modul muss an den Klemmen "+" und "-" mit einer geglätteten +24V DC Speisespannung (19...32V, Welligkeit max. 10%) versorgt werden.
- **System-Erdung**
Für einen störungsfreien Betrieb ist eine einwandfreie Erdung zum Ableiten von externen Störspannungen unerlässlich. Das PCD1/2-System soll an der Klemme "GND" (24 VDC -) mit einem möglichst grossen Querschnitt an der Erdschiene des Steuerschranks angeschlossen sein. Weiter ist darauf zu achten, dass alle Erdleitungen ohne Schleifen ausgeführt sind.
- **Verlegung der Kabel**
Vorgeschrieben ist die Verlegung von Starkstromkabeln und Steuerkabeln in getrennten Kabelkanälen.
- **Motorsteuerungsausgang ($\pm 10V$ analog)**
Überprüfen der Anschlüsse, wie im Handbuch Kap. 5.5 beschrieben. Das Kabel muss abgeschirmt sein.
- **Enable** des Drivers wird über einen "normalen" digitalen Ausgang der PCD1/2 angesteuert.
- **Encoderanschlüsse**
Speziell zu beachten für 5V-Encoder:
 - Kabel muss abgeschirmt und die Leitungen paarweise verdrillt sein.
 - Max. Kabellänge 20m, min. Leiterquerschnitt 0.25mm². Weiter sind die richtige Montage des Encoders (kein Schlupf der Kuppelung), der Typ und die technischen Angaben (Impulse pro Umdrehung) zu prüfen.
- **Ref** (bei PCD2.H310) direkt verdrahten
- **LS1 / LS2**, bei PCD2.H311 auch **Ref**, an "normale" digitale Eingänge der PCD1/2 verdrahten.

9.3 Inbetriebnahme des Antriebs ohne Positioniermodul

Zuerst wird nur der Antrieb (Leistungsstufe und Motor), ohne die Positioniersteuerung in Betrieb genommen. Dazu sind folgende Massnahmen notwendig:

- Die Verbindung vom Stellgrössenausgang "Out" ($\pm 10V$) des H3-Moduls zur Leistungsstufe wird an den Klemmen 6 und "-" (Minus) vom H3-Modul gelöst.
- Die PCD1/2 und das H3-Modul sind ausgeschaltet. Sollte dies nicht möglich sein, so ist zuerst Punkt 9.4.1 auszuführen (Einschalten der Speisespannungen).
- Die Notaus-Endschalter des Antriebs sind so einzustellen, dass bei einem unkontrollierten Wegfahren der Achse kein Schaden entstehen kann.

Jetzt kann die Inbetriebnahme der Leistungsstufe und des Motors gemäss der Anleitung der Lieferanten vorgenommen werden.

9.4 Antrieb mit Positioniermodul

9.4.1 Einschalten der Speisespannungen

Dieser Schritt kann auch vor der Inbetriebnahme der Leistungstufe und des Motors erfolgen, jedoch muss gewährleistet sein, dass die Achse keine unkontrollierten Bewegungen machen kann (Leistungsstufe spannungslos, Reglerfreigabe unterbrochen).

Nach dem erstmaligen Einschalten (ohne angeschlossenes Programmiergerät) der Speisespannung für die PCD1 bzw. PCD2 und des H310 bzw. des H311-Moduls sind die Kontroll-LED zu beachten

auf der PCD1:

Bedeutung	LED	Verhalten
24 VDC	gelb	'Speisespannung vorhanden': muss leuchten
Run	gelb	'CPU in Run': leuchtet nicht, da kein Programm geladen
Error	gelb	sollte nicht leuchten

auf der PCD2:

Bedeutung	LED	Verhalten
24 VDC	gelb	'Speisespannung vorhanden': muss leuchten
Battery	rot	'Battery' (fail): sollte nicht leuchten
Watch Dog	gelb	Watch Dog inaktiv: leuchtet nicht (da kein Programm geladen).
Run	gelb	'CPU in Run': leuchtet nicht, da kein Programm geladen
Halt	rot	'CPU in Halt'. leuchtet, da kein Programm vorhanden
Error	gelb	sollte nicht leuchten

auf dem Positioniermodul PCD2.H310:

Bedeutung	LED	Verhalten
Power	gelb	zeigt die Anwesenh. der $\pm 15V$: muss leuchten
Ref (H310)	gelb	Referenzschalter

auf dem Positioniermodul PCD2.H311:

Bedeutung	LED	Verhalten
Power	gelb	zeigt die Anwesenh. der $\pm 15V$: muss leuchten
Pw 5V	gelb	Speisung 5V für Encoder: muss leuchten

9.4.2 Bereitstellen eines minimalen Anwenderprogrammes

Für die Inbetriebnahme des Antriebes mit dem H3-Modul sollte ein minimales Anwenderprogramm in der PCD1/2 geladen sein, dies, um alle Prüfungen und Einstellungen vornehmen zu können. Das "Einfache Inbetriebnahme-Programm" im Abschnitt 6.1.3 kann für erste Versuche verwendet werden. Ein umfassendes Werkzeug (Commissioning Tool), unter Einbezug von End- und Referenzschaltern, ist in Abschnitt 10 beschrieben.

Das Bereitstellen des Anwenderprogrammes beinhaltet folgende Schritte:

- Installation des Programmier-Pakets PCD9.H31E gemäss Abschnitt 7.1.1.
- Eröffnen eines neuen Projekts.
- Einkopieren der Datei D2H310_B.MBA.
- Anpassen der Datei D2H310_B.MBA mit der Anzahl H31x-Modulen und deren Basisadressen.
- Bestimmen der Maschinendaten.
- Anpassen (Parameter im FB INIT und zu fahrender Hin- und Herweg) und Laden des Inbetriebnahme-Programmes

Auswertung des Flags 'fPar_Err'

Besonders für die Inbetriebnahme empfiehlt es sich, dieses Flag auszuwerten. Dieses Flag ist für alle Achsen eines Projekts nur ein einziges Mal vorhanden.

Nach dem Ansprechen des Flags kann im PCD-Register 'rDiag' die Ursache des Fehlers eingesehen werden. Siehe Abschnitt 8.2.1.

Für das Rücksetzen des Flags 'fPar_Err' ist der Anwender verantwortlich.

9.4.3 Bestimmen der Maschinendaten

Für die Erstinbetriebnahme sind für den FB 'Init' verschiedene Parameter zu ermitteln, welche teilweise bereits durch die Maschine vorgegeben, einige jedoch als Testwerte empfohlen werden.

Positionstoleranz → siehe FB 'Init', Par. 7, Handbuch Seite A-1/2

Empfohlener Wert 1 Encoderumdrehung

Bei einem Encoder mit 500 Imp./Umdrehung hat der Parameter den Wert $2000 = 4 * 500$ Imp./Umdrehung (Flankenauswertung der Encoderimpulse).

PID-Faktoren → siehe FB 'Init' Handbuch Seite A-1/2

Empfohlene Startwerte:

Proportionalfaktor (Parameter 2):	10
Integalfaktor (Parameter 3):	0
Differentialfaktor (Parameter 4):	0
Integrationsgrenze (Parameter 5):	30000
Abtastzeit D-Anteil (Parameter 6):	15 (= 5.61 msec.)

Maschinenfaktor FB 'Init', PCD-Register für Parameter 9 (Floatingpoint-Format)

Dieser Faktor ergibt sich durch die Auflösung des verwendeten Encoders, der Spindelsteigung und einem eventuellen Getriebe.

Wichtig: Die hier gewählte Längeneinheit (m, cm, mm, 1/10mm, 1/100mm, µm) ist zwingend auch für die Geschwindigkeit (Par. 10), die Beschleunigung (Par. 11) sowie für alle Weg-, Breakpoint-, Geschwindigkeits- und Beschleunigungswerte im ganzen Anwenderprogramm dieser Achse zu verwenden.

Es gilt: $k = \frac{4 \times In}{s}$ [Impulse/Längeneinheit]

wobei In: Impulse/Umdrehung (Encoderauflösung)
 s: Weg/Umdrehung (Spindelsteigung und Getriebe)

Beispiel: In = 1000 Impulse pro Umdrehung
 s = 2 mm (Steigung der Spindel) (kein Getriebe)

$$k = \frac{4 \times 1000}{2} = 2000 \text{ Impulse/für 1 mm}$$

= 2 Impulse für 1 μm

ins PCD-Register für den Parameter 9 ist 2000.0 (Floatingpoint-Format) bei einer Längeneinheit von 1 mm, bzw. 2.0 (Floatingpoint-Format) bei einer gewählten Längeneinheit von 1 μm einzuschreiben.

Geschwindigkeit FB 'Init', PCD-Register für Parameter 10

Für die Erstinbetriebnahme empfiehlt es sich, mit einer kleinen Geschwindigkeit zu arbeiten. Der Wert ist in der gleichen Längeneinheit wie beim Maschinenfaktor als Integer-Wert ins dafür vorgesehene PCD-Register einzugeben

Empfehlung: 0.05 m/s → 50 mm/s bzw. 50000 $\mu\text{m/s}$

Es ist zu beachten, dass der gewählte Antrieb die parametrisierte Geschwindigkeit tatsächlich erreichen kann, ansonsten der Regler dauernd eine Abweichung von der Sollgeschwindigkeit feststellt und diese Abweichungen aufaddiert. Dies wird dann zu einer fehlerhaften Bremsrampe bzw. zu einem abrupten Abbruch der Bewegung führen.

Beschleunigung FB 'Init', PCD-Register für Parameter 11

Für die Erstinbetriebnahme empfiehlt es sich, mit einer kleinen Beschleunigung zu arbeiten. Der Wert ist in der gleichen Längeneinheit wie beim Maschinenfaktor als Integer-Wert ins dafür vorgesehene PCD-Register einzugeben

Empfehlung: 0.01 m/s² → 10 mm/s² bzw. 10000 $\mu\text{m/s}^2$

Vor dem Starten einer Bewegung sollte durch manuelles Bewegen der Achse (bei ausgeschaltetem Verstärker) die Funktion der Encoder überprüft werden. Siehe auch Abschnitt 9.4.5.

9.4.4 Drehrichtung, Wegmessung (Encoder)

Zur Überprüfung der Drehrichtung und der Wegmessung müssen folgende Voraussetzungen erfüllt sein:

- Die Verbindung vom Stellgrösseausgang ($\pm 10V$) des H3-Moduls zur Leistungsstufe wird an den Klemmen 6 und GND für die Achse vom H3-Modul gelöst.
- Die PCD1/2 ist eingeschaltet (das Inbetriebnahmeprogramm geladen) und in "Run".
- Das Programmiergerät ist angeschlossen und die Inbetriebnahme-Software gestartet. Im "Configure"-Menu wurden alle notwendigen Einstellungen vorgenommen. Die Einstellungen müssen mit den Definitionen in der D2H310_B.MBA-Datei übereinstimmen.

Für die folgenden Prüfungen wird die Anzeige der Istposition verfolgt.

Prüfung der Drehrichtung:

1. Die Antriebswelle in positiver Richtung drehen
→ Istposition muss zunehmen
2. Die Antriebswelle in negativer Richtung drehen
→ Istposition muss abnehmen

Die Antriebswelle kann, sofern dies möglich ist, von Hand gedreht werden. Andernfalls muss diese durch das Anlegen eines Sollwertes (mittels einer Spannungsquelle von $\pm 0.5 \dots 10V$) an der Leistungsstufe bewegt werden.

Definition der positiven und negativen Drehrichtung:

- Positive Drehrichtung entspricht der Bewegungsrichtung bei Anlegen einer positiven Sollwertspannung ($0 \dots +10V$) an der Leistungsstufe.
- Negative Drehrichtung entspricht der Bewegungsrichtung bei Anlegen einer negativen Sollwertspannung ($0 \dots -10V$) an der Leistungsstufe.
- Bei einem umgekehrten Verhalten sind die beiden Phasensignale A und B des Encoders zu vertauschen.

Prüfung der Wegmessung:

Antriebswelle eine Umdrehung drehen und Istpositionsanzeige verfolgen.

Ein dem Maschinenfaktor k (Parameter 9 im FB 'Init') entsprechender Wert für den Fahrweg muss als Istposition angezeigt werden. Ist die Anzeige nicht richtig, so ist die Berechnung und Eingabe des Maschinenfaktors 'k' noch einmal zu prüfen.

Prüfung des Indexsignals IN:

Diese Prüfung ist nur erforderlich, falls das Indexsignal (auch Nullimpuls genannt) vom Encoder zur Definition der Nullposition ausgewertet wird.

Vorgehen:

(alle folgenden Aktionen werden durch Kommandos des Inbetriebnahmeprogrammes manuell ausgeführt)

1. Anzeige des Indexpositionsregisters beobachten → ein beliebiger Wert wird angezeigt.
2. Den Befehl 'SetIdxPos' ausführen.
→ Beim Erfassen des nächsten Indeximpulses wird die Istposition in das Indexpositionsregister geschrieben.
3. Achse drehen bis die aktuelle Position in das Indexpositionsregister geschrieben wird.
Die Position muss innerhalb von einer Encoderumdrehung einmal in das Register geschrieben werden.
4. Den Befehl "SetIdxPos" wieder ausführen und prüfen, ob die aktuelle Position während einer Umdrehung nur einmal in das Register geschrieben wird.

Ergibt die Prüfung ein anderes Resultat, so kann dies folgende Ursachen haben:

- Defekter Encoder
- Die Signale A, B und IN vom Encoder weisen nicht die Folge auf, wie diese am Eingang des Positionsdecoders vom H3-Modul verlangt werden.

Ist die Folge der Encodersignale nicht bekannt, so muss diese mit Hilfe eines Oszilloskops bestimmt werden.

9.4.5 PID-Regler

Zur Überprüfung und Abstimmung des PID-Reglers müssen folgende Voraussetzungen erfüllt sein:

- Der Regelkreis wird geschlossen, indem bei ausgeschalteter Steuerung das Sollwertkabel der Leistungsstufe an den Klemmen 6 und GND vom H3-Modul wieder angeschlossen wird.
- Die Steuerung einschalten:
ACHTUNG: Notaus betätigen, wenn die Achse unkontrolliert verfahren sollte.
- Das Programmiergerät ist angeschlossen und die Inbetriebnahme-Software gestartet.

Prüfungen:

1. Zielposition 0 laden und Startbefehl ausführen.
 → Die Regelung vom H3-Modul wird eingeschaltet und die Achse wird durch den Regler in ihrer Position gehalten.
2. Eine Zielposition laden, welche innerhalb des erlaubten Verfahrbereichs liegt und die Bewegung starten.
 → Die Achse wird in die vorgegebene Zielposition gefahren.

Falsches Verhalten:

- Die Achse fährt mit maximaler Geschwindigkeit.

Mögliche Ursache	<ul style="list-style-type: none"> • Lageregelkreis (H3-Modul) oder Drehzahlregelkreis (Leistungsstufe) falsch gepolt. Das bedeutet, dass sich bei einer positiven Sollwertspannung (= positive Zielposition) in die Achse in die negative Bewegungsrichtung bewegt. • Der Analogausgang des H3-Moduls ist defekt und gibt eine konstante maximale Spannung von ca. +12V oder -12V an den Stellgrössenausgang.
Behebung	<ul style="list-style-type: none"> • Prüfung der Drehrichtung Kap. 9.4.4 noch einmal ausführen und die notwendigen Umpolungen vornehmen. • H3-Modul austauschen.

- Die Achse fährt mit konstanter langsamer Geschwindigkeit (Drift).

Mögliche Ursache	<ul style="list-style-type: none"> • Der Sollwert gelangt nicht zur Leistungsstufe, Regelung im H3-Modul nicht aktiv.
Behebung	<ul style="list-style-type: none"> • Verdrahtung überprüfen. • Prüfen, ob nach einem Startbefehl 'StartMot' der Profilgenerator im H3-Modul eine Sollgeschwindigkeit (Befehl 'RdTargVel') ausgibt, was bedeutet, dass der Regler im H3-Modul richtig arbeitet.

- Die Achse fährt kurz an und stoppt wieder.

Mögliche Ursache	<ul style="list-style-type: none"> • Die Überwachung des Positionsfehlers hat angesprochen und stoppt den Antrieb. (Der Motor kann der vorgegebenen Bewegung nicht folgen)
Behebung	<ul style="list-style-type: none"> • Eventuelle mechanische Probleme beheben. • Geschwindigkeit und Beschleunigung anpassen. • Max. Wert für den Positionsfehler vergrößern.

- Die Achse fährt ruckartig in eine falsche Zielposition.

Mögliche Ursache	<ul style="list-style-type: none"> • Mechanische Verbindung von Encoder und Motor lose (Kupplung).
Behebung	<ul style="list-style-type: none"> • Eventuelle mechanische Probleme beheben.

Beim Anfahren der Zielposition kann festgestellt werden, dass ein Positionsfehler bestehen bleibt. Dies ist begründet durch die bleibende Regelabweichung bei einem reinen P-Regler (I- und D-Faktor sind 0).

In einem nächsten Schritt müssen also die Regelfaktoren bestimmt werden, um die gewünschte Güte (Genauigkeit und Härte) des Reglers zu erreichen.

Die nachfolgend angegebenen Einstellregeln sind aus den Erfahrungen mit praktischen Anwendungen und Tests entstanden.

Einstellung des Proportionalfaktors (kp):

1. Einen kleinen Proportionalfaktor einstellen (Erfahrungswert 10). Integral- und Differential-Faktor müssen 0 sein (→ reiner P-Regler).
2. Mit der Achse langsam fahren (Zielposition anfahren oder mit der Funktion 'GoForw' resp. 'GoBackw')
3. Kp-Faktor in Schritten vergrössern bis der Regelkreis zu schwingen beginnt. Anschliessend den ermittelten Wert um etwa 30% reduzieren und als kp-Faktor laden. Der Proportionalfaktor ist somit eingestellt und soll vorerst nicht mehr verändert werden.

Einstellung des Integralfaktors (ki):

Der ki-Faktor wird in Schritten vergrössert, bis die gewünschte Ausregelzeit des Positionsfehlers erreicht ist. Der Integralfaktor ist somit eingestellt und soll vorerst nicht mehr verändert werden.

Wird ein zu starkes Überschwingen beim Anfahren der Zielposition festgestellt, so kann das folgende Gründe haben:

- Die Beschleunigung resp. Bremsung ist zu gross gewählt, der Motor kann dem Sollwert nicht folgen. → Beschleunigung reduzieren.
- Gleiches Verhalten kann sich auch bei einer zu gross gewählten Geschwindigkeit ergeben.
- ki-Faktor ist zu gross gewählt.
- Die Praxis hat gezeigt, dass wenn der Integralfaktor vergrössert und der Differentialfaktor auf Null belassen wird, sich die Schwingungsneigung des Reglers erhöht. → Der Differentialfaktor ist gleichzeitig mit dem Integralfaktor zu erhöhen, um das Überschwingen zu reduzieren.

Durch Beobachten der Integrationssumme (Befehl 'RdIntSum') kann ein solches Verhalten gut verfolgt werden. Summiert sich die Integrationssumme während der Bewegung zu einem grossen Wert, so ist mit einem Überschwingen zu rechnen.

Massnahme: - Reduzierung der oben aufgeführten Parameter.
 - Begrenzung der Integrationssumme (Befehl 'LdIntLim').

Einstellung des Differentialfaktors (kd):

Der kd-Faktor wird in Schritten vergrößert, bis die gewünschte Überschwingweite oder Ausregelzeit des Positionsfehlers erreicht ist. Somit ist auch der Differentialfaktor eingestellt.

Einstellregel der Abtastzeit (Befehl 'LdSampInt') für den D-Anteil:

Für einen Betrieb mit langsamen Geschwindigkeiten eine eher grössere Abtastzeit wählen.

Die Erfahrung zeigt, dass die Werte für die Abtastzeit in der Regel zwischen 2 ms und 9 ms betragen. Das entspricht den Werten 5 und 25 für den Befehl 'LdSampInt'.

Optimierung der Einstellungen:

Falls der Regelvorgang nach den vorher eingestellten Werten noch nicht befriedigend ist, muss versucht werden, die einzelnen Regelparameter weiter so zu variieren, dass ein befriedigendes Resultat erreicht wird.

9.4.6 Einfluss der einzelnen Faktoren auf das Verhalten des Reglers

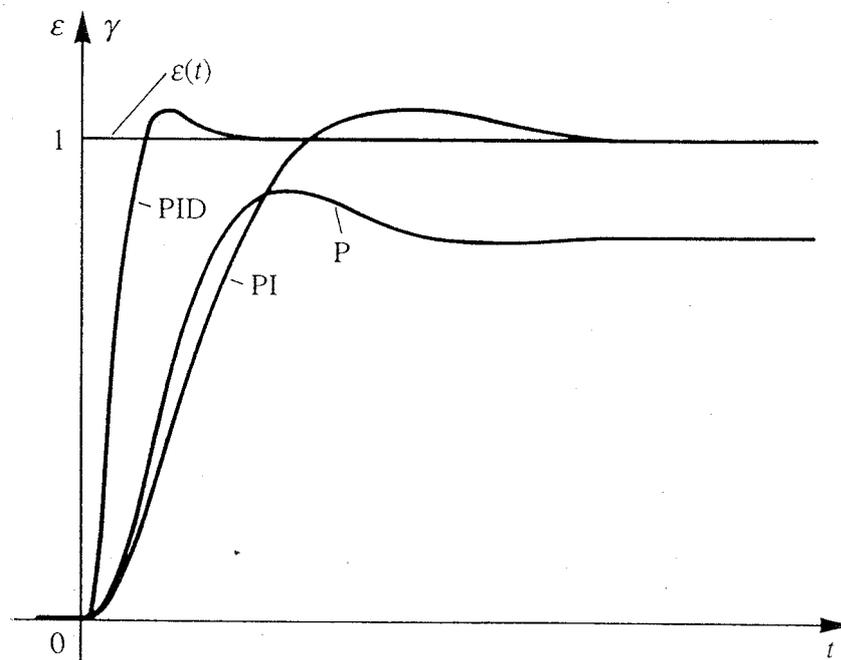
Durch die Vergrößerung des **kp-Faktors** wird die Schwingungsneigung erhöht, jedoch zugleich die bleibende Regelabweichung verkleinert und die Härte des Reglers vergrößert.

Die Erhöhung des **ki-Faktors** erhöht die Schwingungsneigung, sorgt aber gleichzeitig für eine schnellere Ausregelung des Positionsfehlers.

Ein korrekt eingestellter **kd-Faktor** stabilisiert das System und sorgt für ein kleineres Überschwingen und eine kürzere Ausregelzeit. Ein zu hoher kd-Faktor führt zum Schwingen des Systems.

Sprungantwort eines zu regelnden Systems

Einfluss der Parameter k_i , k_p und k_d auf das Verhalten des Reglers.



9.4.7 Einfaches Inbetriebnahmeprogramm

Ein komfortables, grafisches Inbetriebnahmeprogramm (Commissioning Tool) in FUPLA ist im Abschnitt 10 beschrieben.

Ein einfachstes Anwenderprogramm für eine Hin- und Herbewegung mit Anzeige der aktuellen Position, mit einer einzelnen Achse, sieht, unter Verwendung der vorgängig vorgeschlagenen Parameter, etwa wie folgt aus:

(Erläuterungen zu diesem Anwenderprogramm sind dem Abschnitt 6.1.1: "Einsteiger-Beispiel in IL mit Warteschleifen" zu entnehmen).

```

#include D2H310_B.equ
$group H310

xob    16

LD     R 1000
      2.0           ; Mecanic factor
LD     R 1001
      50000        ; Initial absolute speed
LD     R 1002
      10000       ; Initial absolute acceleration

CFB    Init        ; Initialization FB
      K 1          ; Module number
      10          ; Proportional factor (regulator)
      0           ; Intergrative factor (regulator)
      0           ; Derivative factor (regulator)
      30000       ; Integrative limit value
      15         ; Derivative term sampling interval
      2000       ; Position tolerance
      1          ; Behavior in case of position error
      R 1000     ; Mecanic factor register
      R 1001     ; Initial velocity register
      R 1002     ; Initial acceleration register

exob
-----
;
;
cob    0
      0

start: sth  i 0      ; 'Start' OK?

CFB    Exec        ; Executable FB
      K 1          ; Module number
      RdActPos    ; Command: Read Actual Position
      R 90        ; Result register
DSP    R 90        ; Display register

jr     l start     ; if 'Start' not OK, wait

LD     R 100       ; Target Position
      10000      ; Absolute value

CFB    Exec        ; Executable FB
      K 1          ; Module number
      LdDestAbs   ; Command: Load Absolute Destination
      R 100       ; Absolute Destination register

```

```

pos1:      CFB      Exec      ; Executable FB
           K 1          ; Module number
           StartMot    ; Command: Start motion
           rNotUsed    ; Dummy register

           CFB      Exec      ; Executable FB
           K 1          ; Module number
           RdActPos    ; Command: Read Actual Position
           R 90        ; Actual Position register
           DSP R 90    ; Display register

           CFB      Exec      ; Executable FB
           K 1          ; Module number
           RdStatRg    ; Command: Read Status Register
           R 0         ; Value of Status Register

           STH      fOnDest_1 ; Position reached?
           jr      l pos1     ; if no, wait

           ld      t 0
           50

pause1:    sth      t 0
           jr      h pause1

           LD      R 100      ; Target Position
           0          ; Absolute value

           CFB      Exec      ; Executable FB
           K 1          ; Module number
           LdDestAbs   ; Command: Load Absolute Destination
           R 100       ; Absolute Destination register

           CFB      Exec      ; Executable FB
           K 1          ; Module number
           StartMot    ; Command: Start motion
           rNotUsed    ; Dummy register

pos2:      CFB      Exec      ; Executable FB
           K 1          ; Module number
           RdActPos    ; Command: Read Actual Position
           R 90        ; Result register
           DSP R 90    ; Display register

           CFB      Exec      ; Executable FB
           K 1          ; Module number
           RdStatRg    ; Command: Read Status Register
           R 0         ; Value of Status Register

           STH      fOnDest_1 ; Position reached?
           jr      l pos2     ; if no, wait

           ld      t 0
           50

pause2:    sth      t 0
           jr      h pause2

ecob

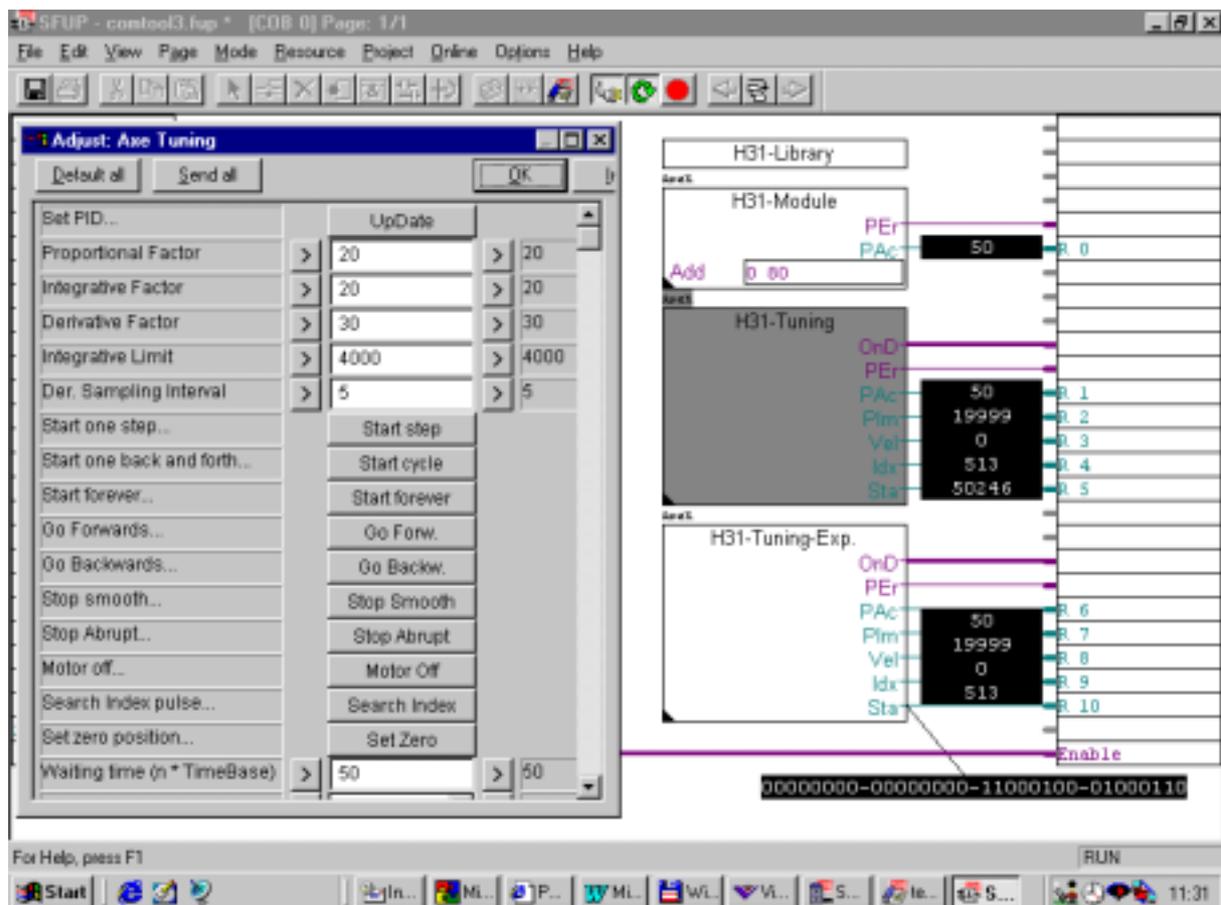
$endgroup

```

10. Inbetriebnahmeprogramm (Commissioning Tool)

Unter der Bezeichnung 'comtool-fup' wird auf der Diskette PCD8.H31 ein Inbetriebnahmeprogramm einer mittleren Komfortstufe angeboten. Das Programm basiert auf dem FUPLA des PG4 ab Version V2.0.70.

Das Programm besteht aus einer Organisations-FBox, einer FBox für die Initialisierung eines PCD2.H31x, einer FBox mit einem einfachen Fahrprogramm, wobei alle PID-Parameter zwecks Optimierung des Regelverhaltens sowie alle Fahrparameter online verändert werden können und schliesslich einer weiteren FBox, mit welcher ein Fahrprogramm mit 4 beliebigen Sequenzen programmiert und online verändert werden kann. Diese so ermittelten optimalen PID-Parameter werden in das definitive Anwenderprogramm übernommen (FB 'Init' und ev. weitere FB 'Exec' zur Anpassung einzelner PID-Parameter).



Das Programm, mit geöffnetem Einstellfenster der FBox mit dem einfachen Fahrprogramm, präsentiert sich auf dem Bildschirm folgendermassen:

Die PID-Parameter werden individuell eingestellt, einzeln übergeben und dann gemeinsam mit 'Update' aktiviert.

Mit den folgenden Tasten wird eine einzelne Fahrt (Start Step), eine Hin- und Herfahrt (Start Cycle) oder eine dauernde Hin- und Herfahrt (Start for ever) aktiviert. Mit dem Parameter 'Waiting time' kann beim Fahrprogramm nach jeder Sequenz ein Pause definiert werden. Auch kann beliebig vor- oder rückwärts gefahren werden (Go forward oder Go backward) und die Fahrt weich (Stop smooth) oder hart (Stop abrupt) gestoppt werden. Mit 'Set Zero' kann jede Position zur neuen Null-Position erklärt werden. Es kann weiter die Motorsteuerung ausgeschaltet (Motor off) oder die nächste Indexposition gesucht werden.

Das Einstellfenster der eben erläuterten FBox für die einfache Hin- und Herfahrt des Schlittens wird nachfolgend in der ganzen Grösse gezeigt..

Set PID...		UpDate	
Proportional Factor	> 20	>	20
Integrative Factor	> 20	>	20
Derivative Factor	> 30	>	30
Integrative Limit	> 4000	>	4000
Der. Sampling Interval	> 5	>	5
Start one step...			Start step
Start one back and forth...			Start cycle
Start forever...			Start forever
Go Forwards...			Go Forw.
Go Backwards...			Go Backw.
Stop smooth...			Stop Smooth
Stop Abrupt...			Stop Abrupt
Motor off...			Motor Off
Search Index pulse...			Search Index
Set zero position...			Set Zero
Waiting time (n * TimeBase)	> 50	>	50
Search index dir.	> Backwards	>	Backwards
Position Ref. Type	> Absolute	>	Absolute
Velocity Ref. Type	> Absolute	>	Absolute
Acceleration Ref. Type	> Absolute	>	Absolute
Destination A	> 50	>	50
Velocity A	> 20	>	20
Acceleration A	> 50	>	50
Destination B	> 0	>	0
Velocity B	> 50	>	50
Acceleration B	> 100	>	100

Wie bereits erwähnt, werden die PID-Parameter individuell eingestellt, einzeln übergeben und dann gemeinsam mit 'Update' aktiviert.

Die Parameter des unteren Teils, also des Fahrprogramms, werden sofort nach der Bestätigung übernommen.

Einstellfenster der Initialisierungs-FBox.

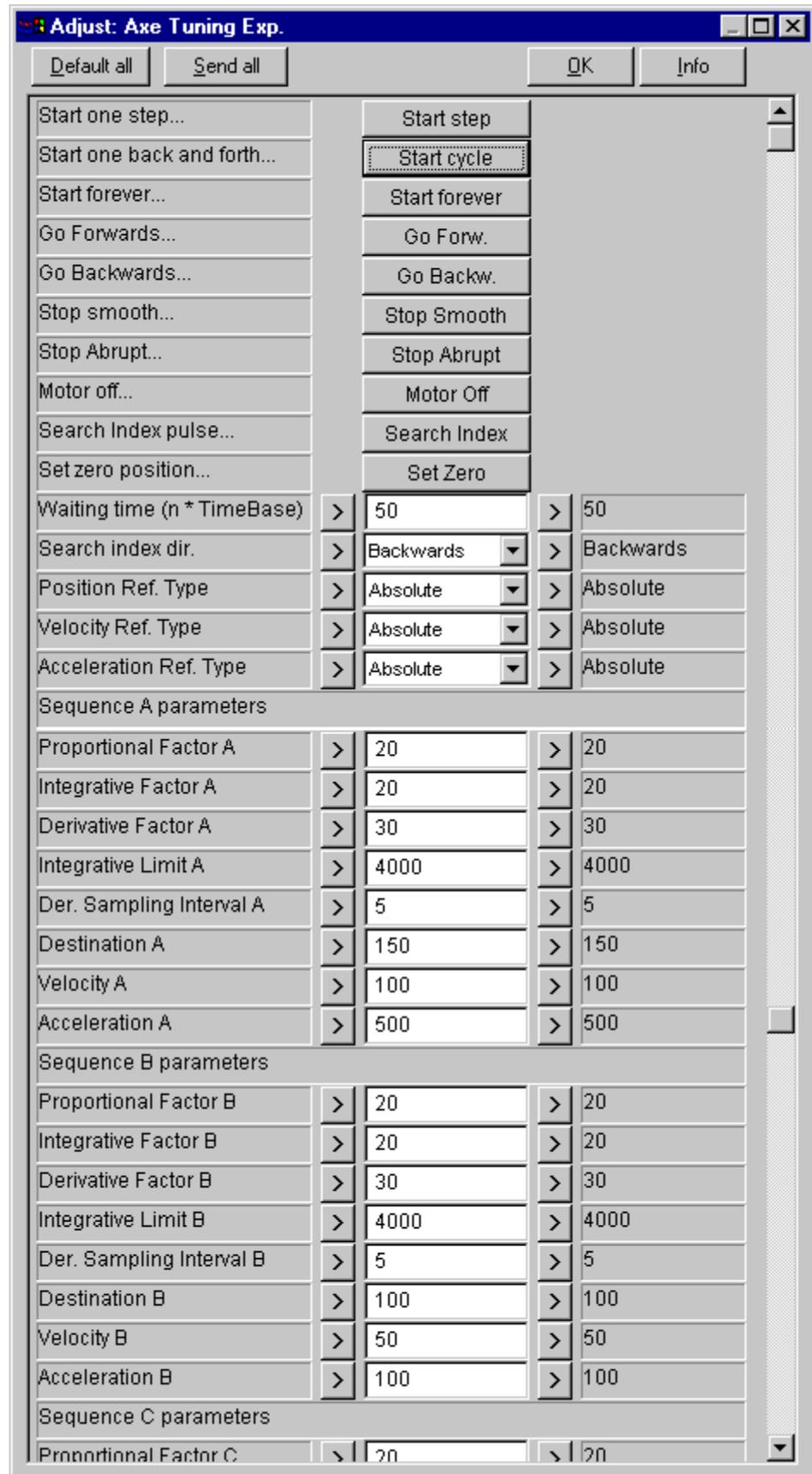
Die Parameter werden nur nach einem Download übernommen. Dieses Einstellfenster ist also **vor** dem Compilieren und **vor** dem Download in die PCD zu parametrieren.

Parameter	Value
Mecanic Factor	4.000000E+02
Initial Velocity	10
Initial Acceleration	200
Proportional Factor	20
Integrative Factor	20
Derivative Factor	30
Integrative Limit	4000
Der. Sampling Interval	5
Position Tolerance	1000
Position Error Behaviour	Set flag

Mit der letzten der 4 FBoxen kann ein Fahrprogramm mit 4 beliebigen Sequenzen programmiert werden. Alle Parameter jeder Sequenz sind einzeln parametrier- und online veränderbar.

Die Bedienung ist die gleiche wie beim einfachen 'Hin- und Her' Programm. Mit der Taste 'Start Step' werden die einzelnen Sequenzen nacheinander abgefahren. Mit 'Start Cycle' wird eine einzelne ganze Sequenz gestartet und mit 'Start for ever' bleibt der Ablauf der Sequenzen bis zum Abbruch mit 'Stop ...' oder 'Motor Off' in Bewegung. Mit dem Parameter 'Waiting time' kann nach jeder Sequenz eine Pause definiert werden. Auch kann beliebig vor- oder rückwärts gefahren werden (Go forward oder Go backward) und die Fahrt weich (Stop smooth) oder hart (Stop abrupt) gestoppt werden. Mit 'Set Zero' kann jede Position zur neuen Null-Position erklärt werden. Es kann weiter die Motorsteuerung ausgeschaltet (Motor off) oder die nächste Indexposition gesucht werden (Backwards oder Forwards). Die Positionen, die Geschwindigkeit und die Beschleunigung können 'Absolute' oder 'Relative' parametrierbar werden.

Das Einstellfenster für die FBox mit dem 4-Sequenzen-Ablauf präsentiert sich wie folgt: (die eingetragenen Werte sind nur als Beispiel gedacht, bewähren sich jedoch mit dem Demomodell V-PCX 24 gut)



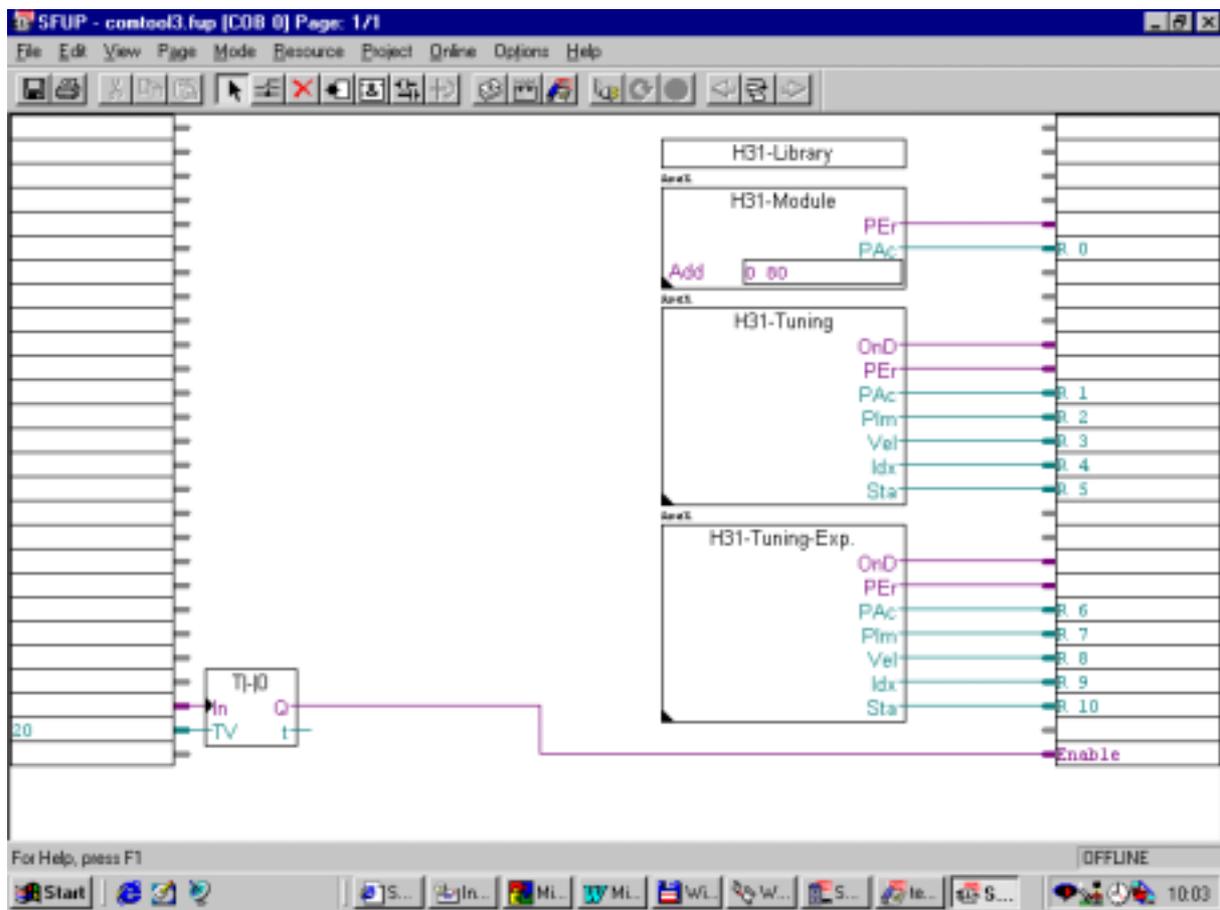
(Einstellfenster Fortsetzung)

Acceleration B	>	100	>	100
Sequence C parameters				
Proportional Factor C	>	20	>	20
Integrative Factor C	>	20	>	20
Derivative Factor C	>	30	>	30
Integrative Limit C	>	4000	>	4000
Der. Sampling Interval C	>	5	>	5
Destination C	>	30	>	30
Velocity C	>	200	>	200
Acceleration C	>	400	>	400
Sequence D parameters				
Proportional Factor D	>	20	>	20
Integrative Factor D	>	20	>	20
Derivative Factor D	>	30	>	30
Integrative Limit D	>	4000	>	4000
Der. Sampling Interval D	>	5	>	5
Destination D	>	0	>	0
Velocity D	>	10	>	10
Acceleration D	>	20	>	20

Für eine bestmögliche Inbetriebnahme, sollte zusätzlich auch ein Speicheroszilloskop zur Anzeige der $\pm 10V$ -Regelspannung zur Verfügung stehen., da nur so das Verhalten des Reglers unter den verschiedenen Belastungen verfolgt und optimal eingestellt werden kann. Leider beinhaltet das vorliegende Commissioning-Tool diese Anzeige nicht.

Beschreibung der einzelnen FBoxen des Commissioning Tools.

Das Commissioning Tool präsentiert sich wie folgt:



Die Anzugverzögerung am unteren Bildrand ist im Originaltool nicht dabei. Es empfiehlt sich, die Freigabe der Motorsteuerung erst nach der Selbstdiagnose der CPU frei zu geben, damit der Schlitten beim Einschalten der Steuerung keine unkontrollierten Sprünge macht.

FBox 'H31-Library'

Organisations-FBox. Muss am Seitenanfang des Tools stehen. Die FBox hat keine Ein- und Ausgänge und auch kein Einstellfenster.

FBox 'H31-Module'

Initialisierung des PCD2.H31x Moduls.

Im Fester 'Add' ist die Basisadresse des Moduls einzuschreiben, z.B. O 80.

Ausgänge der FBox:

'PEr'	binär	Positions-Error
'PAc'	ganzzahl	aktuelle Position

Einstellfenster: siehe Seite 10-4

FBox 'H31-Tuning'

Einfaches Hin- und Her-Fahrprogramm zum Suchen und Optimieren der bestmöglichen Parameter.

Ausgänge der FBox:

'OnD'	binär	On Destination
'PEr'	binär	Positions Error
'PAc'	ganzzahl	aktuelle Position (in Längen-Einheiten)
'PIm'	ganzzahl	aktuelle Position (in Impulsen) *)
'Vel'	ganzzahl	aktuelle Geschwindigkeit
'IdX'	ganzzahl	Index-Register
'Sta'	ganzzahl	Status-Register

Einstellfenster: siehe Seite 10-3

FBox 'H31-Tuning-Exp.'

FBox mit Fahrprogramm mit 4 beliebigen, individuell parametrierbaren Sequenzen.

Ausgänge der FBox:

'OnD'	binär	On Destination
'PEr'	binär	Positions Error
'PAc'	ganzzahl	aktuelle Position (in Längen-Einheiten)
'PIm'	ganzzahl	aktuelle Position (in Impulsen) *)
'Vel'	ganzzahl	aktuelle Geschwindigkeit
'IdX'	ganzzahl	Index-Register
'Sta'	ganzzahl	Status-Register

Einstellfenster: siehe Seite 10-6/7

*) Der Ausgang 'PIm' zeigt als Resultat den Wert des Ausgangs 'PAc' multipliziert mit dem 'Mechanical Factor'.

Notizen

11. Sicherheitsaspekte

Wird eine Antriebseinheit mit einem PCD2.H31x-Modul angesteuert, sollten folgende Punkte besonders beachtet werden:

Einschaltphase des Antriebes

Nach dem Einschalten der Hauptspeisung für die PCD dauert es max. 2 Sekunden bis alle Ein/Ausgangsmodule initialisiert sind, die CPU in den RUN-Zustand geht und somit das Anwenderprogramm abgearbeitet wird.

Der analoge Stellgrössenausgang des H3-Moduls kann während dieser Einschaltphase (max. 2 Sekunden) einen beliebigen Wert zwischen -10V und +10V aufweisen. Aus diesem Grund ist es absolut notwendig, dass der Leistungsteil des Antriebes mit dem Anwenderprogramm über einen digitalen Ausgang entweder eingeschaltet oder freigegeben wird. Somit ist gewährleistet, dass der Antrieb während dieser Einschaltphase nicht unkontrolliert verfährt.

Überwachen des Positionsfehlers

(Siehe Flag 'fPosErr_x')

Eine Überschreitung des zulässigen Positionsfehlers signalisiert ernsthafte Probleme und sollte deshalb immer überwacht werden.

Folgende Ursachen können eine Positionsfehlerüberschreitung zur Folge haben:

1. Anschlussfehler in der H3-Installation.
Beispiele:
 - Lose Verbindung
 - Drehrichtungssinn von Motor und Inkremental-drehgeber stimmen nicht überein.
2. Schlecht eingestellte PID-Parameter.
3. Der Antrieb (Verstärker und/oder Motor) ist zu schwach dimensioniert.
4. Falsch gewählte Bewegungsparameter. Der Servoverstärker bzw. Motor kann der vom H3-Modul vergebenen Beschleunigung oder Geschwindigkeit nicht folgen.
5. Blockierter Rotor des Servomotors infolge mechanischer Probleme.
6. Hardwarefehler im H3-Modul (z.B. analoger Stellgrössenausgang defekt).
7. Hardwarefehler im Servoverstärker.

Eine Positionsfehlerüberschreitung infolge der Punkte 1 bis 4 wird meistens schon bei der Inbetriebnahme entdeckt und kann sofort behoben werden.

Positionsfehlerüberschreitungen infolge der Punkte 5 bis 7 können auch nach der Inbetriebnahme auftreten. Um Schäden an der Maschine zu vermeiden, sind folgende Massnahmen notwendig:

Flag 'fPosErr_x' (signalisiert Positionsfehlerüberschreitung) mit dem Anwenderprogramm permanent überwachen und im Fehlerfall den Antrieb über einen digitalen Ausgang abschalten.

Der Antrieb kann auf unterschiedliche Arten abgeschaltet werden. Bei manchen Antrieben genügt es, wenn am Leistungsteil die Reglerfreigabe weggenommen wird. Bei andern wiederum ist es notwendig, den Antrieb so schnell wie möglich abzubremsen (durch Kurzschliessen des Sollwerteinganges am Leistungsteil) und die Haupteinspeisung zeitverzögert (einige Millisekunden) zu unterbrechen. Auf welche Weise abgeschaltet werden kann bzw. muss, ist von der jeweiligen Maschine abhängig und muss von Fall zu Fall entschieden werden.

Limit Switches

Diese können unterschiedlich verwendet werden:

- im Zusammenhang mit der Funktion 'Home' zum Suchen der Nullposition beim Einschalten der Anlage
- als Alarmmeldung im Anwenderprogramm zur Signalisierung einer Positionsüberschreitung (wenn entsprechend positioniert und programmiert)

Sicherheitsendschalter

Die Sicherheitsendschalter (Notausendschalter) sollten immer direkt den Antrieb abschalten. (Haupteinspeisung des Antriebes unterbrechen.)

Watchdog

Den Watchdog der PCD unbedingt aktivieren und den Watchdogkontakt verwenden, um den Antrieb direkt abzuschalten.

X0Bs für Hardwarefehler der PCD

XOB 0 bis 5 programmieren und den Antrieb nötigenfalls über einen digitalen Ausgang direkt abschalten.

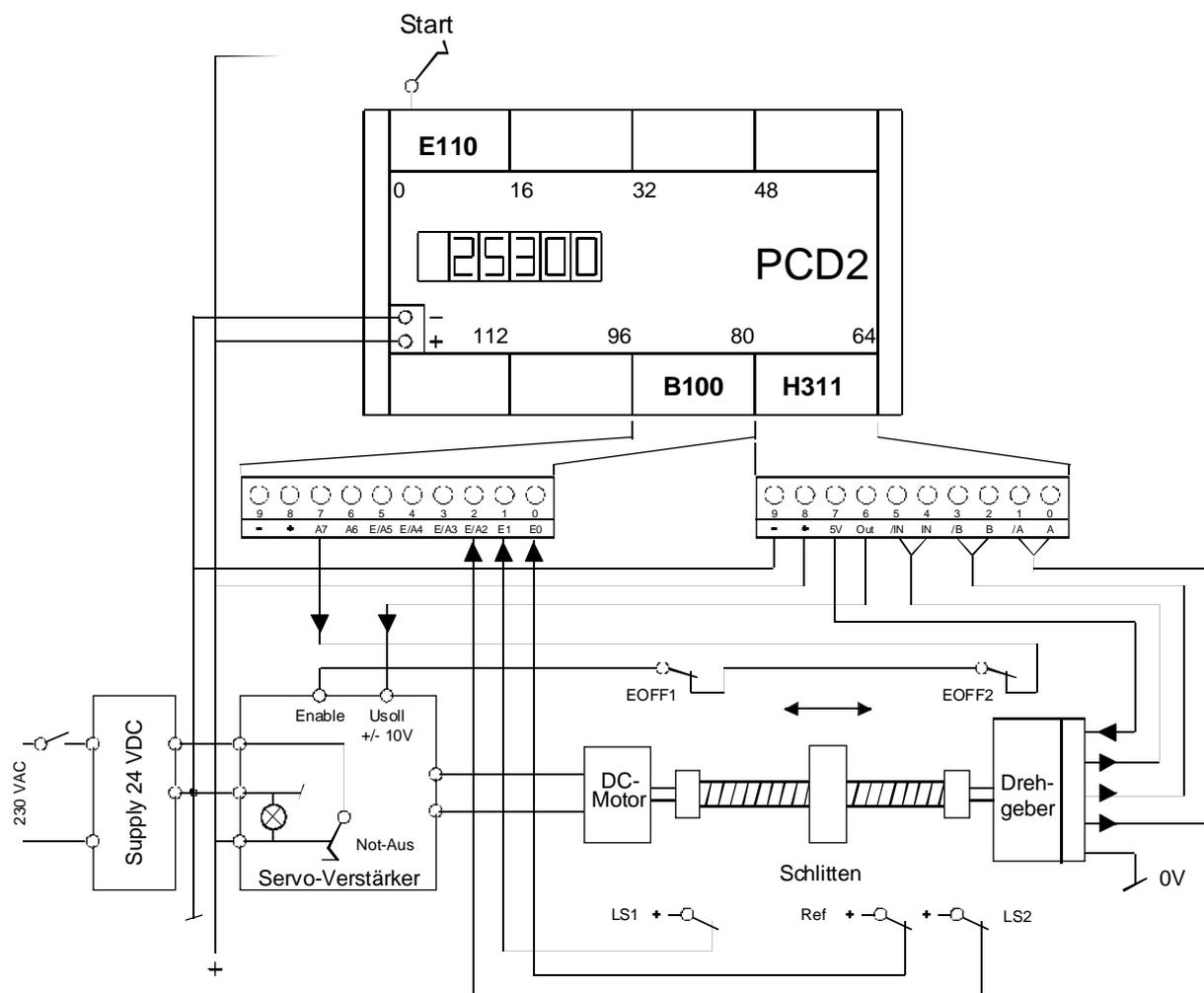
12. Anwendungsbeispiele

12.1 Beispiel: 1 Achse mit Referenzfahrt

Es handelt sich bei diesem Beispiel um eine einfache Hin- und Herbewegung eines Schlittens, wie bereits im Kapitel 6: "Der schnelle Einstieg" beschrieben. Es werden im vorliegenden Beispiel zusätzlich der Referenz- und die beidseitig des Fahrbereichs angebrachten Endschalter für eine automatische Referenzfahrt mit einbezogen.

Die Hardware beruht auf dem Workshopmodell V-PCX 24, bestehend aus dem Schlitten mit Gleichstrommotor, Referenz-, End- und Sicherheitsendschaltern, dem Servoverstärker mit Notausschalter, der Speisung und einer PCD2 mit 6-stelligem Display.

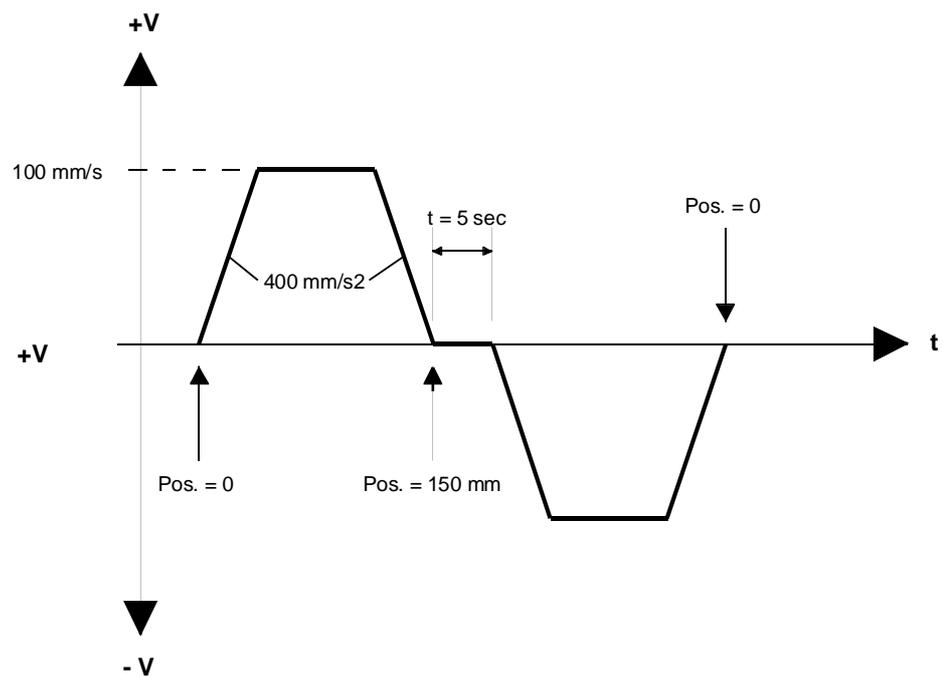
Das bestückte und verkabelte Modell präsentiert sich wie folgt:



Daten des Schlittens:

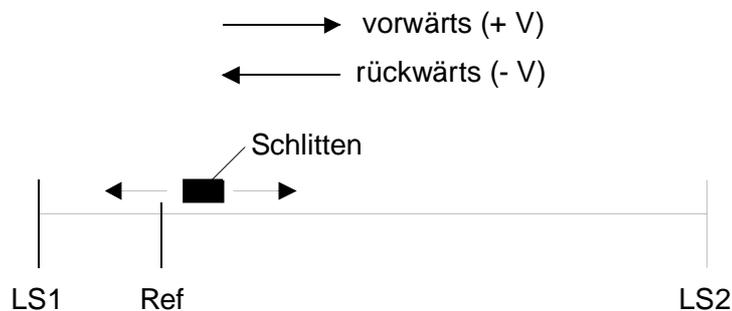
Encoder:	500 Impulse/Umdrehung
Spindel:	Steigung 2 mm
Maximale Geschwindigkeit:	100 mm/s
Max. zulässige Beschleunigung:	50 mm/s ²

Nach einer Referenzfahrt, unmittelbar nach dem Einschalten des Systems, soll folgender Bewegungsablauf ausgeführt werden:



Referenzfahrt

Vorgängig der Präsentation des ganzen Anwenderprogramms, soll die Referenzfahrt etwas näher betrachtet werden.



Die Anordnung der Schalter auf dem Modell ist etwa gemäss obiger Skizze. Der Schlitten steht bei der normalen Programmausführung in der gezeichneten Position.

Es wird mal davon ausgegangen, dass der Schlitten im Normalfall zwischen der Position 'Ref' und 'LS2' steht. Es soll bei einer Referenzfahrt der Schlitten in Richtung zum Referenzschalter, also mit der Suchrichtung 'rückwärts' fahren. Beim Erreichen des Referenzschalters ist dieser in Vorwärtsfahrrichtung bis zum Erreichen des nächsten Indexsignals des Inkrementaldrehgebers frei zu fahren. Die Referenzposition ist damit erreicht und das Fahrprogramm kann gestartet werden.

Der FB 'Home' ist dazu folgendermassen zu parametrieren:

```
LD      R 1010      ; PCD-Register mit Geschwindigkeit zum
          500        ; Freifahren vom Referenzschalter (langsam)

LD      R 1011      ; PCD-Register mit Geschwindigkeit zum
          5000       ; Suchen des Referenzschalters (schnell)

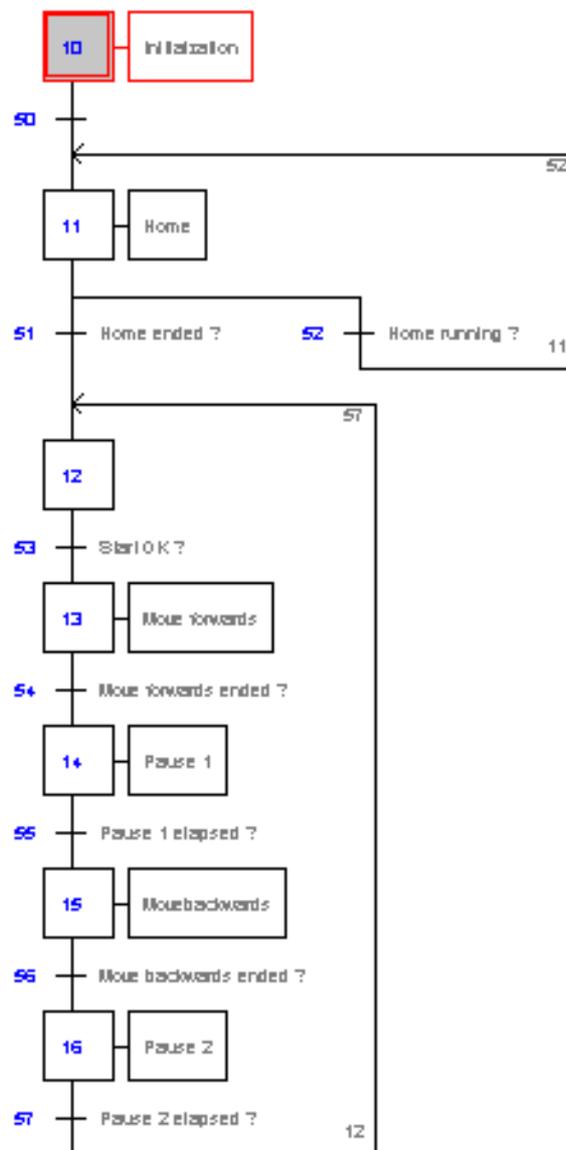
CFB     Home        ; Aufruf des FB 'Home'
K 1     1           ; Par 1: Modul-Nummer
          0           ; Par 2: Suchrichtung rückwärts
          1           ; Par 3: Freifahrrichtung vorwärts
R 1010  1           ; Par 4: Geschwindigkeit für Freifahren
R 1011  1           ; Par 5: Geschwindigkeit für Suchen
          30          ; Par 6. Timeout 30s
I 64    1           ; Par 7: Adresse des Referenzeingangs
```

Verhalten bei verschiedenen Ausgangspositionen:

- a) Ausgangsposition 'normal' (Schlitten zwischen Pos 'Ref' und 'LS2'): Verhalten wie eben beschrieben. Wird der Referenzpunkt innerhalb des definierten Timeouts nicht gefunden, wird die Referenzfahrt abgebrochen. Ist der Referenzschalter beim Einschalten betätigt (Schlitten steht auf dem Referenzschalter), so erfolgt nur das Freifahren und die Fahrt zum nächsten Indexsignal.
- b) Ausgangsposition mit Schlitten zwischen 'LS1' und 'Ref': Der Schlitten fährt zuerst rückwärts bis zu 'LS1', fährt danach vorwärts zum Referenzschalter 'Ref'. Der Referenzschalter wird mit der Freifahrtgeschwindigkeit überfahren. Nach dem Freifahren wird wiederum bis zum Erreichen des nächsten Indexsignals des Inkrementaldregebers weitergefahren. Die Referenzposition ist damit erreicht und das Fahrprogramm kann gestartet werden.
- c) Die Ausgangsposition liegt rechts von 'LS2': Die Referenzfahrt läuft gleich ab wie bei a). Der 'LS2' wird ohne Beeinflussung der Referenzfahrt überfahren.
- d) Die Ausgangsposition liegt links vom 'LS1'. Der Schlitten wird gemäss Definition rückwärts bis zum Sicherheitsendschalter fahren. Die Referenzposition kann nicht gefunden werden. Wird der Sicherheitsendschalter innerhalb des definierten Timeouts nicht erreicht, wird die Referenzfahrt abgebrochen.

Um allen Eventualitäten vorzubeugen, z.B. mehrmalige Betätigung der Limit Switches während der Referenzfahrt von Hand, ist das 'fHomeErr_x'-Flag in den Programmablauf einzubeziehen.

Das Anwenderprogramm in GRAFTEC (int-home.sfc)



Die Initialisierung erfolgt im IST. Es wird danach die Referenzfahrt ausgeführt (ST 1). Dieser ST 1 wird, solange die Referenzfahrt noch nicht beendet ist, dauernd durchlaufen. Das Ende der Referenzfahrt wird mit dem Flag 'fEndHome_x' signalisiert. Vorgängig des Aufrufs des 'Home'-FB sind die Endschalter 'LS1' und 'LS2', welche auf digitale PCD-Eingänge verdrahtet sind, auf die Flags 'fLS1_x' und 'fLS2_x' einzulesen.

Nach beendeter Referenzfahrt geht das Programm zu TR 53, wo die Startbedingung des eigentlichen Fahrprogramms abgewartet wird.

Die Beschreibung des Fahrprogramms kann dem Abschnitt 6.1.2 entnommen werden.

Der Code des Programms "int-home.sfc"

(Um diese Darstellung zu erhalten, ist die Datei "int-home.sfc" in "int-home.src" umzubenennen.

```

SB      0
;-----
IST     10      ;Initialization
        O 50

$include D2H310_B.equ
$group H310

LD      R 1000
        1.00      ; Mecanic factor
LD      R 1001
        100000    ; Initial absolute speed
LD      R 1002
        400000    ; Initial absolute acceleration

CFB     Init      ; Initialization FB
        K 1        ; Module number
        250       ; Proportional factor (regulator)
        0         ; Intergrative factor (regulator)
        60        ; Derivative factor (regulator)
        4000      ; Integrative limit value
        5         ; Derivative term sampling interval
        500       ; Position tolerance
        0         ; Behavior in case of position error
        R 1000    ; Mecanic factor register
        R 1001    ; Initial velocity register
        R 1002    ; Initial acceleration register

ld      t 0       ; Delay for Enable
        20        ; 2 sec
EST     ;10

```

```

;-----
ST      11          ;Home
        I 50
        I 52          ;Home running ?
        O 51          ;Home ended ?
        O 52          ;Home running ?

set    o 71          ; Enable

res    fEndHome_1

stl    i 65
out    fLS1_1
stl    i 66
out    fLS2_1

ld     r 1010
        1000
ld     r 1011
        10000

cfb    home
        k 1          ; Module number
        0          ; search direction
        1          ; free run direction
        r 1010       ; min. speed
        r 1011       ; max. speed
        50          ; timeout
        i 64          ; input reference switch
EST    ;11

;-----
ST      12
        I 51          ;Home ended ?
        I 57          ;Pause 2 elapsed ?
        O 53          ;Start OK ?
EST    ;12

;-----
ST      13          ;Move forwards
        I 53          ;Start OK ?
        O 54          ;Move forwards ended ?

LD     R 100        ; Target Position
        60000       ; Absolute value

CFB    Exec         ; Executable FB
        K 1          ; Module number
        LdDestAbs   ; Command: Load Absolute Destination
        R 100        ; Absolute Destination register

CFB    Exec         ; Executable FB
        K 1          ; Module number
        StartMot    ; Command: Start motion
        rNotUsed    ; Dummy register
EST    ;13

;-----
ST      14          ;Pause 1
        I 54          ;Move forwards ended ?
        O 55          ;Pause 1 elapsed ?

ld     t 0
        50
EST    ;14

```

```

;-----
ST      15          ;Movebackwards
        I 55          ;Pause 1 elapsed ?
        O 56          ;Move backwards ended ?

LD      R 100       ; Target Position
        0            ; Absolute value

CFB     Exec        ; Executable FB
        K 1          ; Module number
        LdDestAbs   ; Command: Load Absolute Destination
        R 100       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1          ; Module number
        StartMot    ; Command: Start motion
        rNotUsed    ; Dummy register
EST     ;15

;-----
ST      16          ;Pause 2
        I 56          ;Move backwards ended ?
        O 57          ;Pause 2 elapsed ?

ld      t 0
        50
EST     ;16

;-----
TR      50
        I 10          ;Initialization
        O 11          ;Home

stl     t 0
ETR     ;50

;-----
TR      51          ;Home ended ?
        I 11          ;Home
        O 12

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdActPos    ; Command: Read Actual Position
        R 90        ; Actual Position register

DSP     R 90        ; Display register

sth     fEndHome_1
ETR     ;51

;-----
TR      52          ;Home running ?
        I 11          ;Home
        O 11          ;Home

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdActPos    ; Command: Read Actual Position
        R 90        ; Actual Position register

DSP     R 90        ; Display register

stl     fEndHome_1
ETR     ;52

```

```

;-----
TR      53      ;Start OK ?
        I 12
        O 13      ;Move forwards
CFB     Exec    ; Executable FB
        K 1      ; Module number
        RdActPos ; Command: Read Actual Position
        R 90     ; Actual Position register
DSP     R 90    ; Display register

sth     i 0
ETR     ;53

;-----
TR      54      ;Move forwards ended ?
        I 13      ;Move forwards
        O 14      ;Pause 1

CFB     Exec    ; Executable FB
        K 1      ; Module number
        RdActPos ; Command: Read Actual Position
        R 90     ; Actual Position register
DSP     R 90    ; Display register
CFB     Exec    ; Executable FB
        K 1      ; Module number
        RdStatRg ; Command: Read Status Register
        R 0      ; Value of Status Register

STH     fOnDest_1 ; Position reached?
ETR     ;54

;-----
TR      55      ;Pause 1 elapsed ?
        I 14      ;Pause 1
        O 15      ;Movebackwards
stl     t 0
ETR     ;55

;-----
TR      56      ;Move backwards ended ?
        I 15      ;Movebackwards
        O 16      ;Pause 2

CFB     Exec    ; Executable FB
        K 1      ; Module number
        RdActPos ; Command: Read Actual Position
        R 90     ; Actual Position register
DSP     R 90    ; Display register
CFB     Exec    ; Executable FB
        K 1      ; Module number
        RdStatRg ; Command: Read Status Register
        R 0      ; Value of Status Register

STH     fOnDest_1 ; Position reached?
ETR     ;56

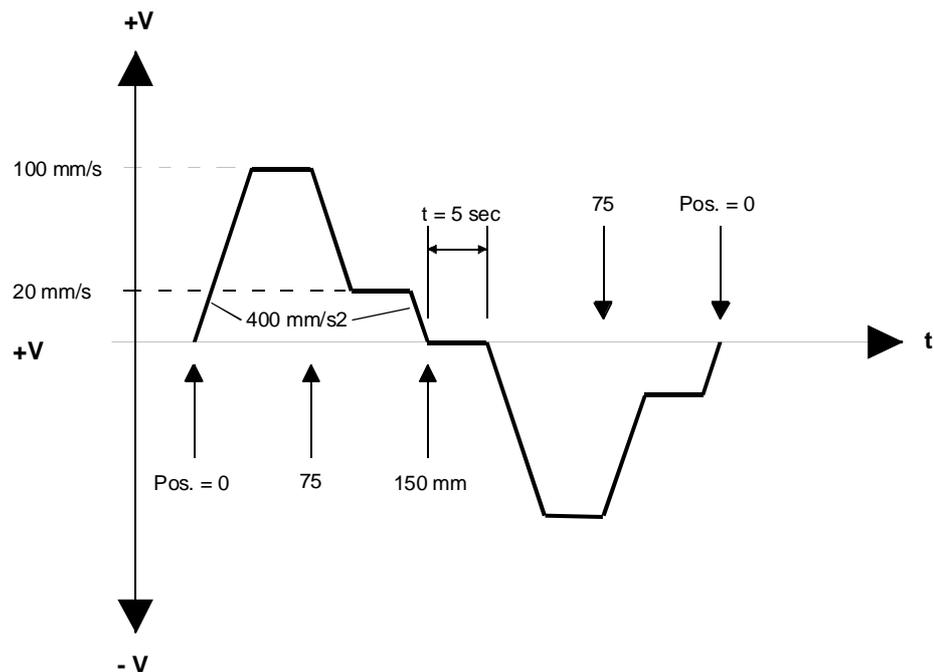
;-----
TR      57      ;Pause 2 elapsed ?
        I 16      ;Pause 2
        O 12
stl     t 0
$endgroup
ETR     ;57

ESB     ;0

```

12.2 Beispiel: 1 Achse mit zwei Geschwindigkeiten

Es handelt sich bei diesem Beispiel wieder um eine Hin- und Herbewegung eines Schlittens, wobei die Fahrgeschwindigkeit sowohl bei der Vorwärts- wie auch bei Rückwärtsfahrt von schnell auf langsam umgeschaltet wird. Es soll an diesem Beispiel vor allem der Mechanismus des Änderns von Parametern während der Fahrt gezeigt werden. Um das Programm nicht zu überlasten, wird auf die 'Home'-Funktion verzichtet.

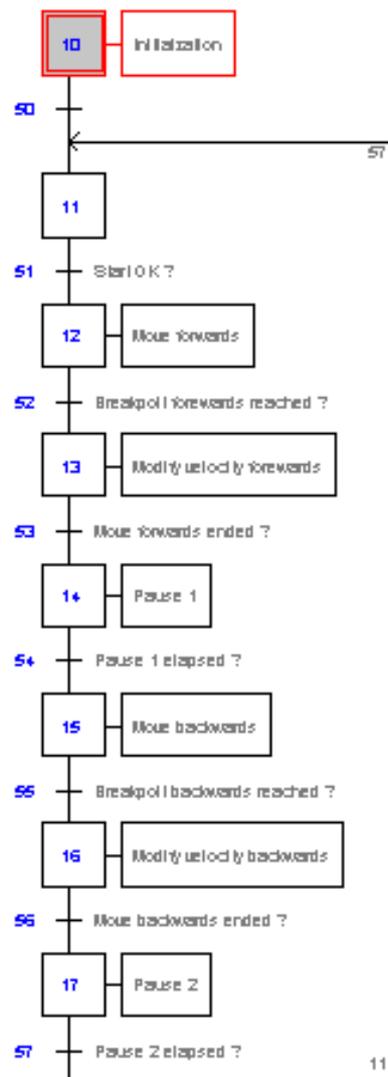


In den Beispielen der Abschnitte 6.1.2 "Einsteiger-Beispiel" und 11.1 "1 Achse mit Referenzfahrt" wurde jeweils mit fixen Parametern für Geschwindigkeit, Beschleunigung und PID-Werten gearbeitet. Alle diese Parameter wurden im FB 'Init' einmal festgelegt und während dem Programmablauf nicht mehr verändert. Die einzigen Werte, welche verändert werden, sind die relativen oder absoluten Positionen für die Fahrten des Schlittens.

Im vorliegenden Beispiel soll während der Fahrt die Geschwindigkeit gemäss dem Diagramm verändert werden. Es wird dazu die Technik mit einem Breakpoint angewandt. Es wird in den ST 12 und 15 nicht nur die nächste Zielposition sondern auch ein Breakpoint definiert. Die Position des Breakpoints ist der Umschaltzeitpunkt der Geschwindigkeit.

In den TR 52 und 55 wird das Erreichen des Breakpoints durch die Abfrage des Flags 'fBrkPt_x' ermittelt und danach in den ST 13 und 16 die neue Geschwindigkeit geladen und mit je einem FB 'Start' aktiviert. Das Ändern anderer Parameter während der Fahrt läuft nach dem gleichen Schema ab.

Das Anwenderprogramm in GRAFTEC (2-speed.sfc)



Die Initialisierung erfolgt im IST.

In den ST 12 und 15 wird sowohl die nächste Zielposition als auch der Breakpoint zum Umschalten der Geschwindigkeit sowie die nächste Fahrgeschwindigkeit eingegeben und mit 'Start' aktiviert.

In den TR 52 und 55 wird jeweils das Erreichen des Breakpoints, zwecks Steuerung des Anwenderprogramms, abgewartet. Das korrekte Umschalten der Fahrgeschwindigkeit geschieht durch das Modul selbst und nicht durch das Anwenderprogramm.

In den ST 13 und 16 wird die neue (langsame) Geschwindigkeit definiert und mit 'Start' aktiviert.

In den TR 53 und 56 wird das Erreichen der jeweiligen Zielposition zwecks Steuerung des Anwenderprogramms abgewartet. Das korrekte Anhalten der Fahrt geschieht durch das Modul selbst und nicht durch das Anwenderprogramm.

In den meisten TR wird die aktuelle Position gelesen und zum Display gebracht. Auf diese Programmteile kann in der Praxis verzichtet, bzw. nur dort vorgesehen werden, wo dies sinnvoll erscheint.

Der Code des Programms "2-speed.sfc"

(Um diese Darstellung zu erhalten, ist die Datei "int-home.sfc" in "2-speed.src" umzubenennen.

```

SB      0
;-----
IST     10           ;Initialization
        O 50

$include D2H310_B.equ
$group H310

LD      R 1000
        1.00         ; Mecanic factor
LD      R 1001
        100000       ; Initial absolute speed
LD      R 1002
        400000       ; Initial absolute acceleration

CFB     Init        ; Initialization FB
        K 1          ; Module number
        250         ; Proportional factor (regulator)
        0           ; Intergrative factor (regulator)
        60          ; Derivative factor (regulator)
        4000        ; Integrative limit value
        5           ; Derivative term sampling interval
        500         ; Position tolerance
        0           ; Behavior in case of position error
        R 1000       ; Mecanic factor register
        R 1001       ; Initial velocity register
        R 1002       ; Initial acceleration register

ld      t 0         ; Delay for Enable
        20          ; 2 sec

EST     ;10

;-----
ST      11
        I 50
        I 57         ;Pause 2 elapsed ?
        O 51         ;Start OK ?

set     o 71        ; Enable

EST     ;11

```

```

;-----
ST      12          ;Move forwards
        I 51        ;Start OK ?
        O 52        ;Breakpoint forewards reached ?

LD      R 100       ; Target Position
        60000      ; Absolute value

LD      R 101       ; Breakpoint Position
        30000      ; Absolute value

LD      R 102       ; Speed
        100000     ; Absolute value

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdDestAbs  ; Command: Load Absolute Destination
        R 100       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdBrkPtAbs ; Command: Load Absolute Breakpoint
        R 101       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdVelAbs   ; Command: Load absolute Velocity
        R 102       ; Speed

CFB     Exec        ; Executable FB
        K 1         ; Module number
        StartMot   ; Command: Start motion
        rNotUsed   ; Dummy register

EST     ;12

;-----
ST      13          ;Modify velocity forewards
        I 52        ;Breakpoint forewards reached ?
        O 53        ;Move forwards ended ?

LD      R 102       ; Speed
        20000      ; Absolute value

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdVelAbs   ; Command: Load absolute Velocity
        R 102       ; Speed

CFB     Exec        ; Executable FB
        K 1         ; Module number
        StartMot   ; Command: Start motion
        rNotUsed   ; Dummy register

EST     ;13

;-----
ST      14          ;Pause 1
        I 53        ;Move forwards ended ?
        O 54        ;Pause 1 elapsed ?

ld      t 0
        50

EST     ;14

```

```

;-----
ST      15          ;Move backwards
        I 54        ;Pause 1 elapsed ?
        O 55        ;Breakpoint backwards reached ?

LD      R 100       ; Target Position
        0           ; Absolute value

LD      R 101       ; Target Position
        30000      ; Absolute Breakpoint

LD      R 102       ; Speed
        100000     ; Absolute value

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdDestAbs  ; Command: Load Absolute Destination
        R 100       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdBrkPtAbs ; Command: Load Absolute Breakpoint
        R 101       ; Absolute Destination register

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdVelAbs   ; Command: Load absolute Velocity
        R 102       ; Speed

CFB     Exec        ; Executable FB
        K 1         ; Module number
        StartMot   ; Command: Start motion
        rNotUsed   ; Dummy register

EST     ;15

;-----
ST      16          ;Modify velocity backwards
        I 55        ;Breakpoint backwards reached ?
        O 56        ;Move backwards ended ?

LD      R 102       ; Speed
        20000      ; Absolute value

CFB     Exec        ; Executable FB
        K 1         ; Module number
        LdVelAbs   ; Command: Load absolute Velocity
        R 102       ; Speed

CFB     Exec        ; Executable FB
        K 1         ; Module number
        StartMot   ; Command: Start motion
        rNotUsed   ; Dummy register

EST     ;16

;-----
ST      17          ;Pause 2
        I 56        ;Move backwards ended ?
        O 57        ;Pause 2 elapsed ?

ld      t 0
        50

EST     ;17

```

```

;-----
TR      50          ;Initialization
        I 10
        O 11
ETR     ;50
;-----

TR      51          ;Start OK ?
        I 11
        O 12          ;Move forwards

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdActPos    ; Command: Read Actual Position
        R 90        ; Actual Position register
DSP     R 90        ; Display register

sth     i 0

ETR     ;51
;-----

TR      52          ;Breakpoint forewards reached ?
        I 12          ;Move forwards
        O 13          ;Modify velocity forewards

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdActPos    ; Command: Read Actual Position
        R 90        ; Actual Position register
DSP     R 90        ; Display register

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdStatRg    ; Command: Read Status Register
        R 0          ; Value of Status Register

STH     fBrkPt_1    ; Breakpoint position reached?

ETR     ;52
;-----

TR      53          ;Move forwards ended ?
        I 13          ;Modify velocity forewards
        O 14          ;Pause 1

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdActPos    ; Command: Read Actual Position
        R 90        ; Actual Position register
DSP     R 90        ; Display register

CFB     Exec        ; Executable FB
        K 1          ; Module number
        RdStatRg    ; Command: Read Status Register
        R 0          ; Value of Status Register

STH     fOnDest_1   ; Position reached?

ETR     ;53

```

```

;-----
TR      54      ;Pause 1 elapsed ?
        I 14      ;Pause 1
        O 15      ;Move backwards

stl    t 0

ETR     ;54

;-----
TR      55      ;Breakpoint backwards reached ?
        I 15      ;Move backwards
        O 16      ;Modify velocity backwards

CFB     Exec     ; Executable FB
        K 1       ; Module number
        RdActPos ; Command: Read Actual Position
        R 90      ; Actual Position register
DSP    R 90      ; Display register

CFB     Exec     ; Executable FB
        K 1       ; Module number
        RdStatRg ; Command: Read Status Register
        R 0       ; Value of Status Register

STH     fBrkPt_1 ; Breakpoint position reached?

ETR     ;55

;-----
TR      56      ;Move backwards ended ?
        I 16      ;Modify velocity backwards
        O 17      ;Pause 2

CFB     Exec     ; Executable FB
        K 1       ; Module number
        RdActPos ; Command: Read Actual Position
        R 90      ; Actual Position register
DSP    R 90      ; Display register

CFB     Exec     ; Executable FB
        K 1       ; Module number
        RdStatRg ; Command: Read Status Register
        R 0       ; Value of Status Register

STH     fOnDest_1 ; Position reached?

ETR     ;56

;-----
TR      57      ;Pause 2 elapsed ?
        I 17      ;Pause 2
        O 11

stl    t 0

$endgroup

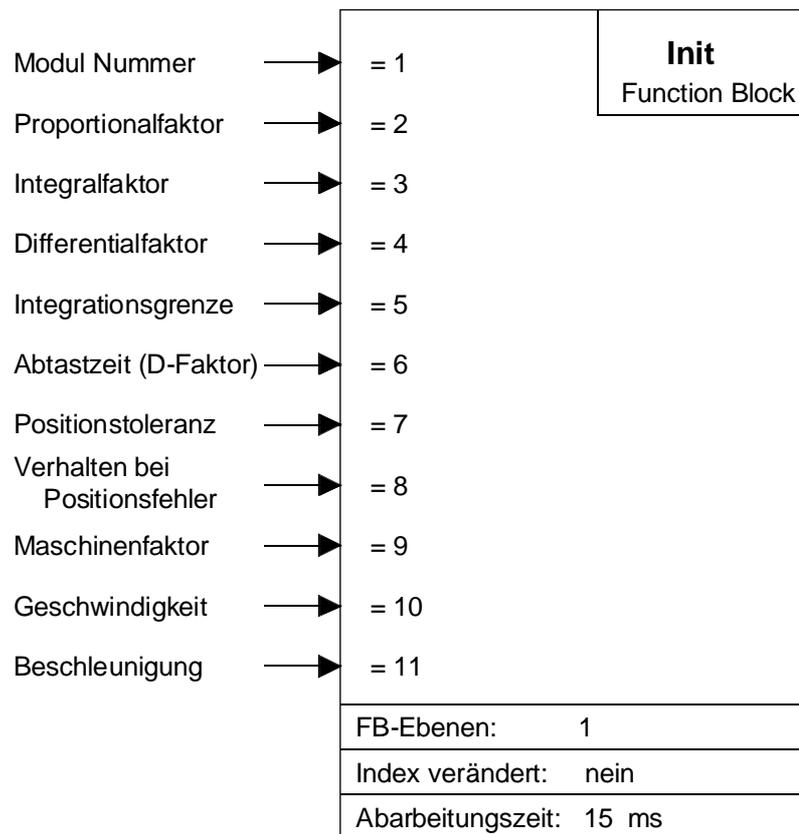
ETR     ;57

ESB     ;0

```


Anhang A: Zusammenfassung aller Software-Elemente für die Programmierung in IL

Der Funktionsblock 'Init'

Init
FB: Initialisierung eines H31x Moduls


Funktionsbeschreibung:

Dieser FB definiert die Einstellungen eines PCD2.H31x Moduls und liest die Basisadresse des Moduls aus der D2H310_B.MBA-Datei.

Parameter '1' ist als Konstante 'K', Parameter '9' bis '11' sind als PCD-Registeradressen (absolut oder symbolisch), alle andern Parameter als ganzzahlige (Integer-) Werte anzugeben.

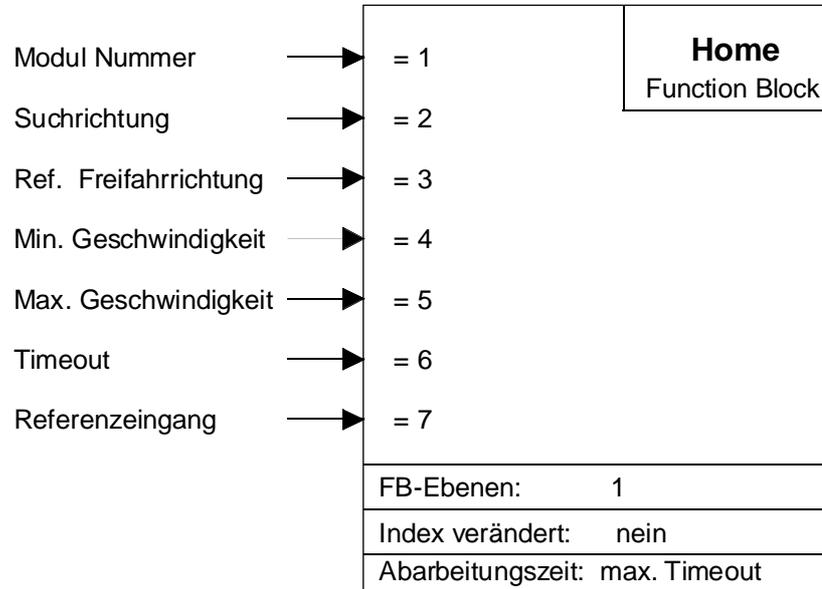
Parameter '9' ist der Maschinenfaktor. Dieser enthält die Parameter des Encoders und den mechanischen Parameter des Systems. (Muss im Floatingpoint-Format, z.B. für '4' als 4.0 oder 4E1, eingegeben werden).

Parameter	Bedeutung	Typ	Format	Wert	Kommentar
= 1	Modul Nummer.	K	K n	K 1 – K 16	
= 2	Proportionalfaktor		Integer	0 - 32767	P-Faktor des PID-Reglers.
= 3	Integralfaktor		Integer	0 - 32767	I-Faktor des PID-Reglers.
= 4	Differentialfaktor		Integer	0 - 32767	D-Faktor des PID-Reglers.
= 5	Integrationsgrenze		Integer	0 - 32767	Begrenzung des Einflusses des I-Anteils im System.
= 6	Abtastzeit des D-Faktors		Integer	0 - 255	Abtastintervall für den D-Faktor
= 7	Positionstoleranz		Integer	0 – 32767	Maximaler Positionsfehler, verglichen mit dem berechneten Wert.
= 8	Verhalten bei Positionsfehler		Integer	0 - 1	Verhalten bei Überschreitung der Positionstoleranz 0 = Errorflag = H 1 = Stop der Bewegung
= 9	Maschinenfaktor	R	Floating	$0..9.223371 * 10^{18}$	Faktor, welcher die Parameter des Encoders und des Systems enthält.
= 10	Geschwindigkeit	R	Integer	--	Initialgeschwindigkeit, vom Maschinenfaktor definiert.
= 11	Bescheunigung	R	Integer	--	Initialbeschleunigung, vom Maschinenfaktor definiert.

Funktionsblock 'Home'

Home

FB: Initialisierung der Referenzposition



Funktionsbeschreibung:

Dieser FB definiert die Einstellungen für die Referenzfahrt für das PCD2.H31x-Modul.

Parameter 1 ist als 'K'-Konstante, Parameter 7 als 'I'-Eingang, alle andern Parameter sind als Ganzzahlwerte (integer) einzugeben.

Das PCD2.H310-Modul hat einen 'Ref'-Eingang. Die Adresse dieses Eingangs ist die Basisadresse des Moduls + 11. Für ein H310-Modul auf der Basisadresse 64 liegt der 'Ref'-Eingang auf Adresse I 75.

Das PCD2.H311-Modul hat keinen 'Ref'-Eingang. Es kann dafür ein beliebiger PCD-Eingang gewählt werden.

Das Flag 'fEndHome' ist vor jeder Referenzfahrt rückzusetzen, andernfalls kann die Referenzfahrt nicht gestartet werden. Dieses Flag wird am Ende jeder Referenzfahrt automatisch = H gesetzt.

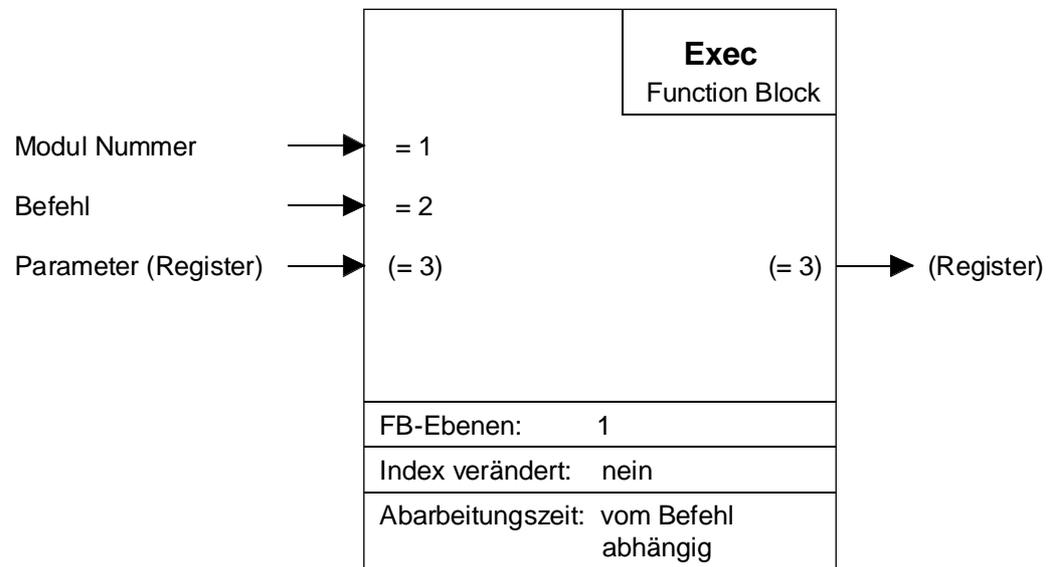
Das Flag 'fHomeErr_x' wird bei einer fehlerhaften Referenzfahrt gesetzt und kann bei Bedarf im Anwenderprogramm ausgewertet werden.

Parameter	Bedeutung	Typ	Format	Wert	Kommentar
= 1	Modul Nummer	K	K n	K 1 – K 16	
= 2	Suchrichtung		Integer	0 - 1	Definiert die Richtung, um den Referenzschalter zu finden: 0 = rückwärts 1 = vorwärts
= 3	Freifahrtrichtung vom Referenzschalter		Integer	0 - 1	Definiert die Freifahrtrichtung, weg vom Referenzschalter: 0 = rückwärts 1 = vorwärts
= 4	Minimale Geschwindigkeit	R	Integer	--	Geschwindigkeit zum Verlassen des Referenzschalters.
= 5	Maximale Geschwindigkeit	R	Integer	--	Geschwindigkeit zum Suchen des Referenzschalters.
= 6	Timeout		Integer	0 – 65535 [s]	Zeit, bevor ein Abbruch der Referenzfahrt erfolgt.
= 7	Referenzeingang	I	Integer	--	Eingang, an welchen der Referenzschalter angeschlossen wird.

Funktionsblock 'Exec'

Exec

FB: Ausführung eines Befehls für das H31x-Modul



Funktionsbeschreibung:

Mit diesem FB werden die Ausführungsbefehle zum PCD2.H31x-Modul geschickt.

Die Modul-Nr. (Parameter 1) muss eine Konstante sein (k 1...k 16). Die Basisadresse wird in der Datei 'D2H310_B.MBA' definiert. Die FBs unterstützen max. 16 PCD2.H31x-Module pro PCD-System.

Die einzelnen Befehle (Parameter 2) werden auf den folgenden Seiten behandelt.

Der Parameter eines Befehls (z.B. der Beschleunigungswert beim Befehl LdAccAbs) wird in einem Register übergeben (Parameter 3). Falls ein Befehl keinen Parameter benötigt (z.B. Start), kann irgend ein Register oder das 'rNotUsed' übergeben werden.

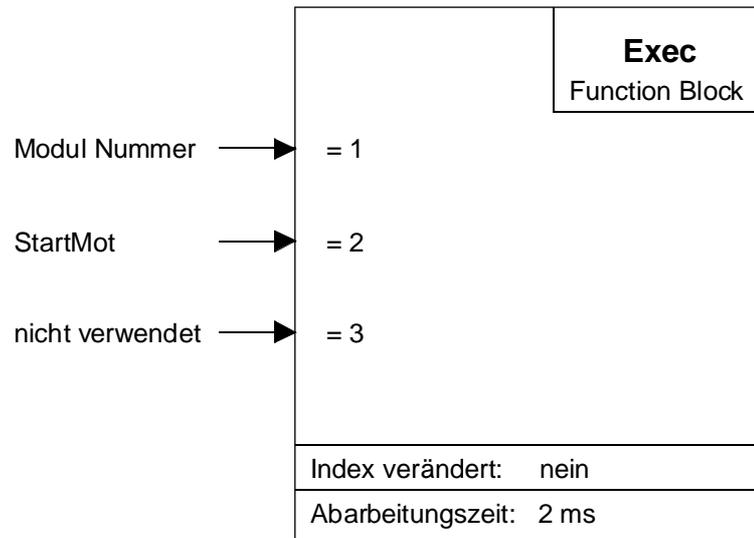
Die einzelnen Befehle für das PCD2.H310 (FB-Parameter)

No.	Symbol	Befehl (English)	Befehl (deutsch)	Seite
01	StartMot	Start Motion	Start der Bewegung (Positions-Modus)	A-7
02	StopUrg	Stop motion in Urgency	Notstop	A-8
03	Stop	Stop motion	Stoppen der Bewegung	A-9
04	MotOff	Motor regulation Off	Motorregelung ausschalten	A-10
05	RdActPos	Read Actual Position	Lesen der aktuellen Position	A-11
06	RdActVel	Read Actual Velocity	Lesen der aktuellen Geschwindigkeit	A-12
07	RdIntSum	Read Integration Sum	Lesen der Integrations-Summe	A-13
08	RdIndexRg	Read Index Register	Lesen des Index-Registers	A-14
09	RdStatRg	Read Status Register	Lesen des Status-Registers	A-15
10	RdTargPos	Read Target Position	Lesen der Zielposition	A-16
11	RdTargVel	Read Target Velocity	Lesen der Ziel-Geschwindigkeit	A-17
12	GoForw	Go Forwards	Bewegung vorwärts (Geschw.-Modus)	A-18
13	GoBackw	Go Backwards	Bewegung rückwärts (Geschw.-Modus)	A-19
14	SgStpFor	Single Step Forwards	Einzelschritt vorwärts	A-20
15	SgStpBak	Single Step Backwards	Einzelschritt rückwärts	A-21
16	LdDestAbs	Load Dest. Absolute	Laden der absoluten Ziel-Position	A-22
17	LdDestRel	Load Dest. Relative	Laden der relativen Ziel-Position	A-23
18	LdVelAbs	Load Velocity Absolute	Laden der absoluten Geschwindigkeit	A-24
19	LdVelRel	Load Velocity Relative	Laden der relativen Geschwindigkeit	A-25
20	LdAccAbs	Load Acc. Absolute	Laden der absoluten Beschleunigung	A-26
21	LdAccRel	Load Acc. Relative	Laden der relativen Beschleunigung	A-27
22	LdPropG	Load Proportional Gain	Laden des Proportionalfaktors	A-28
23	LdIntG	Load Integrative Gain	Laden des Integralfaktors	A-29
24	LdDerG	Load Derivative Gain	Laden des Differentialfaktors	A-30
25	LdSamplnt	Load derivative Sampling Interval	Laden der Abtastzeit	A-31
26	LdIntLim	Load Integrative Limit	Laden der Integrationsgrenze	A-32
27	ActRegFact	Activate Regulation Factors	Aktivieren der Regelfaktoren	A-33
28	LdBrkPtAbs	Load Breakp. Absolute	Laden eines absoluten Breakpoints	A-34
29	LdBrkPtRel	Load Breakp. Relative	Laden eines relativen Breakpoints	A-35
30	ResStatRg	Reset Status Register	Rücksetzen des Statusregisters	A-36
31	SetIdxPos	Set Index Position	Setzen der Indexposition	A-37
32	SetZero	Set Zero position	Setzen der Null-Position	A-38
33	MotConf	Motion Configuration	Konfigurieren Drehzahl- bzw. Positionsgeregelter Betrieb	A-39
34	SetPosTol	Set Position Tolerance	Setzen der Positionstoleranz	A-40

Die Zahl in der ersten Kolonne (0 - 34) ist der absolute Wert des Parameters Nr. 2 im FB 'Exec'. Mit Hilfe dieser Zahl kann, wenn das Anwenderprogramm im Debugger verfolgt wird, die Funktion des FB 'Exec' ge- deutet werden.

StartMot**Befehl:** Start Motion (Start der Bewegung)

[01]

**Funktionsbeschreibung:**

Mit diesem Befehl wird eine Bewegung mit der definierten Geschwindigkeit und Beschleunigung zur vordefinierten Zielposition gestartet. Es wird mit diesem Befehl auch der 'positionsregelte Betrieb' gewählt.

Dieser Befehl schaltet 2 Bit im Status-Byte zurück:

Bit 6 : Breakpoint reached (Breakpoint erreicht)

Bit 5 : Excessive position error (Positionsfehler)

Der Befehl 'ResStatRg' ermöglicht das Zurücksetzen aller oder einem Teil der Statusbits. Das Statusbyte ist in den ersten 8 Bit des Signalregisters abgebildet. Diese 8 Bit werden gleichzeitig mit dem Statusbyte zurückgesetzt.

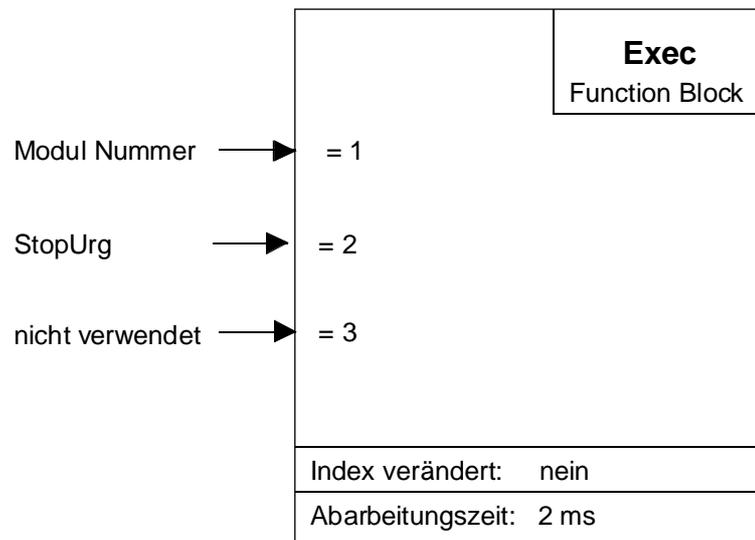
Der Befehl 'StartMot' setzt auch die Bit 10 bis 15 des Signalregisters, welche mit dem Befehl 'RdStatRg' gelesen werden können, zurück.

Die Flags 'fOnDest_x', 'fBrkPt_x' und 'fPosErr_x' werden nur nach einem Befehl 'StartMot', gefolgt von einem Befehl 'RdStatRg' aufgefrischt.

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: StartMot				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

StopUrg **Befehl:** Stop motion in Urgency (Notstop)

[02]



Funktionsbeschreibung:

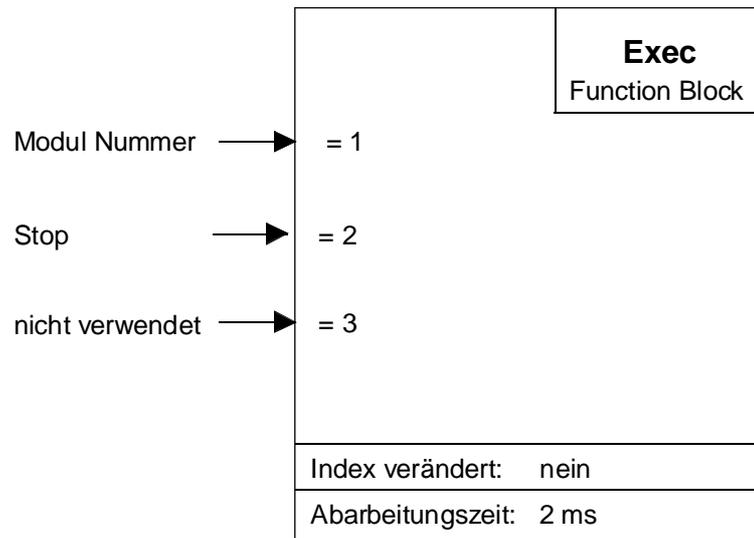
Mit diesem Befehl wird eine Bewegung abrupt gestoppt (ohne Bremsrampe).

Dieser Befehl ist sowohl im Positions- wie auch im Geschwindigkeitsmodus verwendbar.

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: StopUrg				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

Stop**Befehl:** Stop motion (Stoppen der Bewegung)

[03]

**Funktionsbeschreibung:**

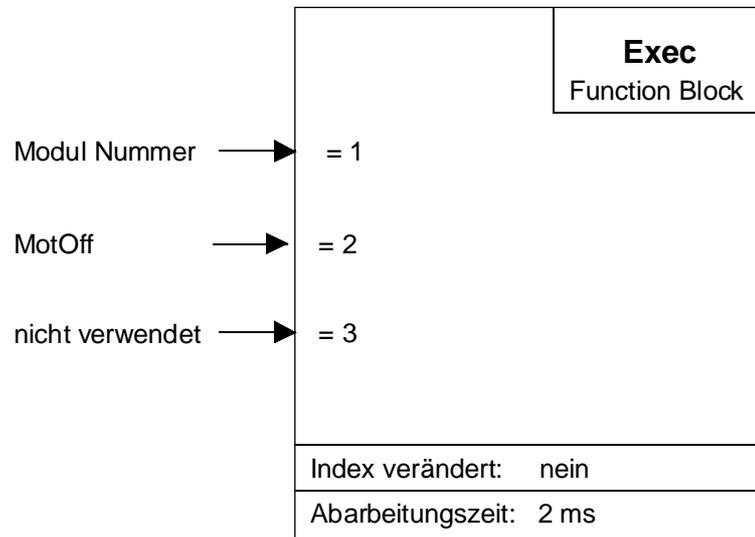
Mit diesem Befehl kann eine Bewegung unter Verwendung der normalen Bremsrampe gestoppt werden.

Dieser Befehl ist sowohl im Positions- wie auch im Geschwindigkeitsmodus verwendbar.

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: Stop				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

MotOff**Befehl:** Motor regulation is put Off (Motor-Regelung ausschalten)

[04]

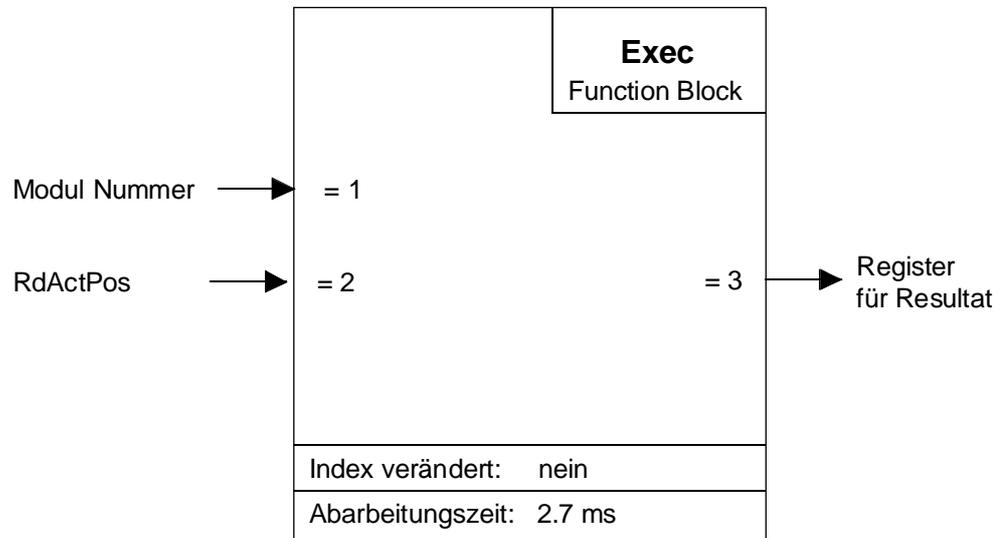
**Funktionsbeschreibung:**

Mit diesem Befehl wird die PID-Funktion, welche auf den Motor wirkt, ausgeschaltet. Die Wirkung ist die Gleiche, wie wenn die Speisung des Motors weggenommen würde.

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: MotOff				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

RdActPos**Befehl:** Read Actual Position (Lesen der aktuellen Position)

[05]

**Funktionsbeschreibung:**

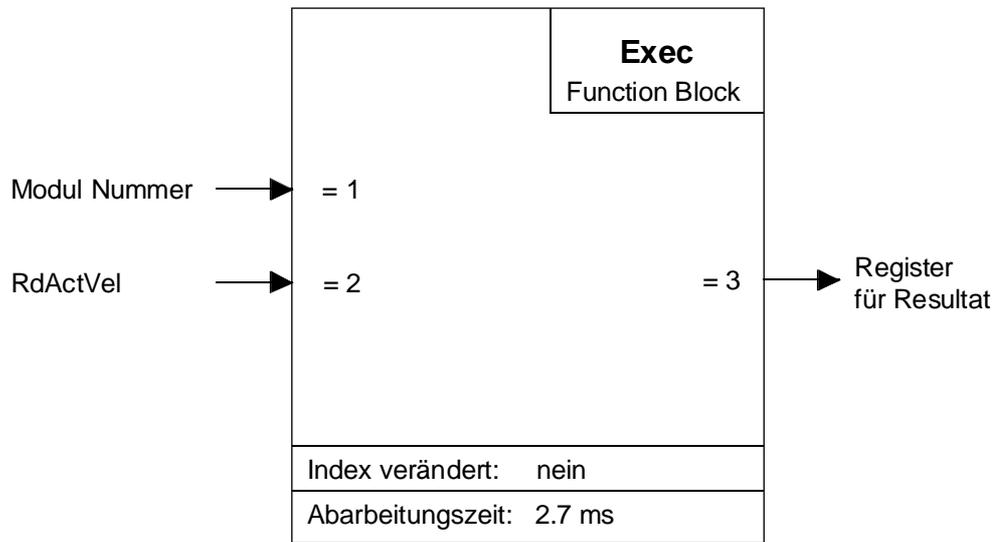
Mit diesem Befehl wird die aktuelle Position des Systems gelesen. Der Befehl ist jedesmal aufzurufen, wenn die Position gelesen werden soll.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdActPos				
= 3	PCD-Register für die aktuelle Position	R	integer	31 Bit	$-2^{30} \dots +(2^{30} - 1)$

RdActVel **Befehl: Read Actual Velocity (Lesen der aktuellen Geschwindigkeit)**

[06]



Funktionsbeschreibung:

Mit diesem Befehl kann die aktuelle Geschwindigkeit der Bewegung gelesen werden. Für jedes neue Lesen der Geschwindigkeit ist der Befehl auszuführen.

Das Format des Prozessors für die Geschwindigkeit ist 14 Bit Ganzzahl- (integer) und 16 Bit Nachkomma-Stellen (fractional number). Mit dem vorliegenden Befehl gibt der Prozessor nur den Integer-Anteil aus, so dass bei geringen Geschwindigkeiten die erhaltenen Werte nicht mit dem Resultat des Befehls 'RdTargVel' verglichen werden können, da hier der "fractional part" enthalten ist.

Ist der Maschinenfaktor kleiner als 1, wird es schwierig, kleine Geschwindigkeiten zu lesen, da der "fractional part" der Geschwindigkeit nicht vorhanden ist.

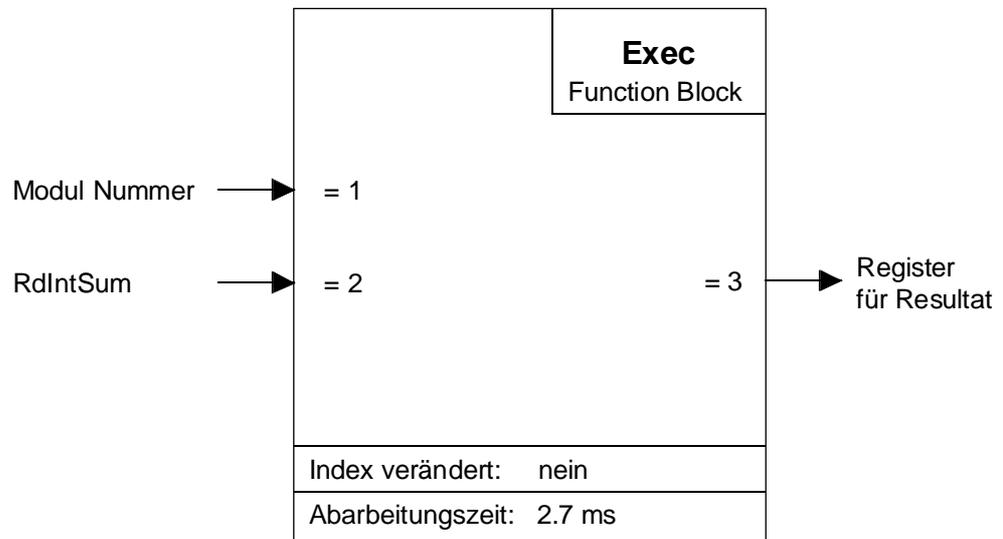
Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdActVel				
= 3	PCD-Register für die aktuelle Geschwindigkeit	R	integer	30 Bit *)	abhängig vom Maschinenfaktor

*) Auflösung 14 Bit

RdIntSum**Befehl:** Read Integration Sum (Lesen der Integrations-Summe)

[07]

**Funktionsbeschreibung:**

Mit diesem Befehl kann die Integrations-Summe des I-Anteils des PID-Reglers gelesen werden.

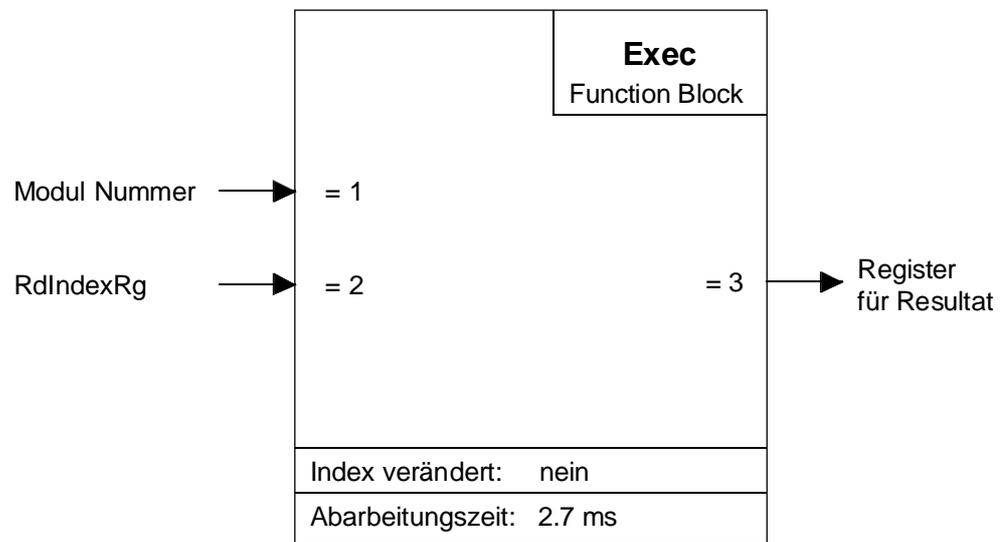
Der Befehl ist jedes Mal aufzurufen, wenn die Integrations-Summe gelesen werden soll.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdIntSum				
= 3	PCD-Register für die Integrations-Summe	R	integer	15 Bit	0 .. 32767

RdIndexRg**Befehl:** Read Index Register (Lesen des Index-Registers)

[08]

**Funktionsbeschreibung:**

Mit diesem Befehl wird das Index-Register (Umdrehungszähler des Encoders) des H310-Moduls gelesen. (Nicht zu verwechseln mit dem Index-Register der CPU der PCD). Wurde ein Befehl 'SetIdxPos' (Set Index Position) ausgeführt, wird beim nächsten Index-Impuls des Encoders die absolute Position aufgezeichnet.

Das Lesen des Index-Registers kann ein Teil des Error-Checking Programmteils sein. Die neue Index-Position minus die alte, geteilt durch die Encoder-Auflösung (Encoderteilung mal 4), muss immer einen ganzzahligen Wert ergeben.

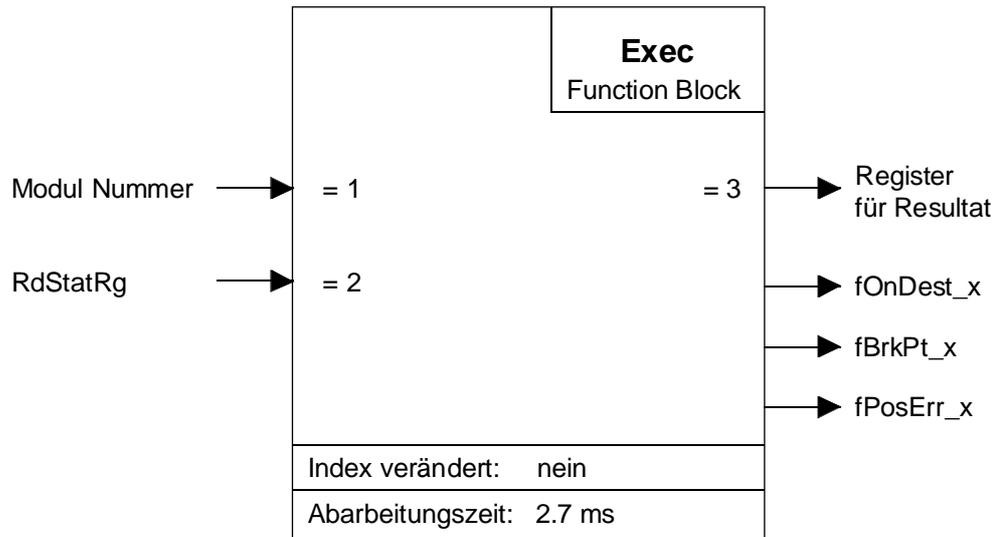
Der Befehl ist jedes Mal aufzurufen, wenn das Index-Register gelesen werden soll.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdIndexRg				
= 3	PCD-Register für den Wert des Index-Registers	R	integer	31 Bit	$-2^{30} .. +(2^{30} - 1)$

RdStatRg**Befehl:** Read Status Register (Lesen des Status-Registers)

[09]

**Funktionsbeschreibung:**

Mit diesem Befehl kann das Status-Register gelesen werden. Der Befehl muss ausgeführt werden **bevor** die Flags 'fOnDest_x', 'fPosErr_x' und 'fBrkPt_x' ausgewertet werden können.

Das Lesen des Status-Registers erlaubt den Zugang zu den folgenden Parametern:

- Bit 15: Host Interrupt
- Bit 14: Acceleration loaded
- Bit 13: Regulator parameters updated
- Bit 12: Forward direction
- Bit 11: Velocity mode
- Bit 10: On Target
- Bit 9: Turn off upon excessive position error
- Bit 8: Internal use only
- Bit 7: Motor off
- Bit 6: Breakpoint reached
- Bit 5: Excessive position error
- Bit 4: Wraparound occurred
- Bit 3: Index pulse acquired
- Bit 2: Trajectory complete
- Bit 1: Internal use only
- Bit 0: Acquire next index pulse

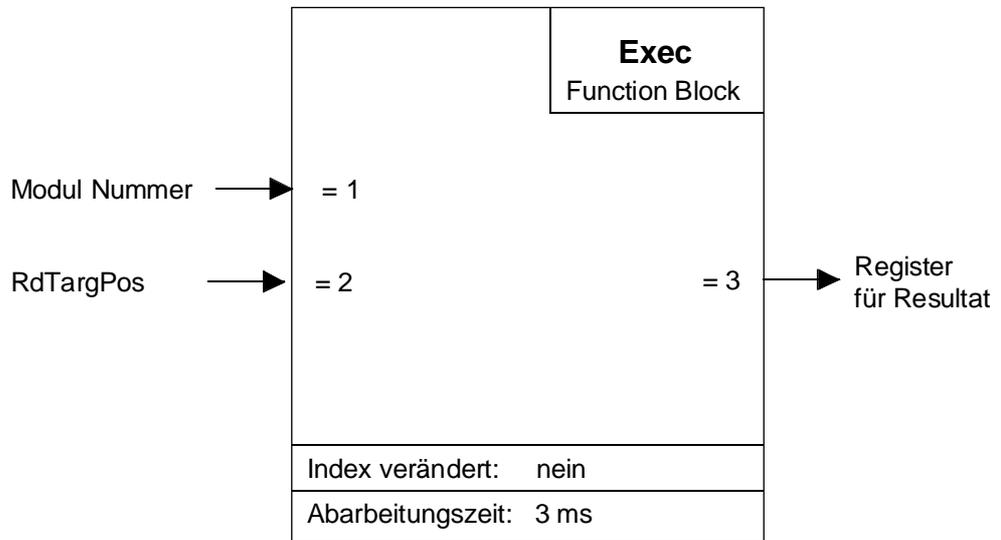
Der Befehl ist jedes Mal aufzurufen, wenn das Status-Register gelesen werden soll.

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdStatRg				
= 3	PCD-Register für den Wert des Status-Registers	R	integer	16 Bit	
fOnDest_x	Flag on destination	F			
fBrkPt_x	Flag breakpoint reached	F			
fPosErr_x	Flag excessive position error	F			

RdTargPos

Befehl: Read Target Position (Lesen der Soll-Position)

[10]

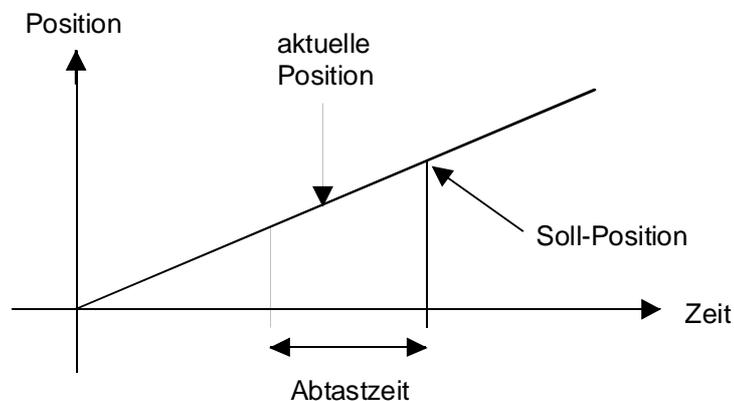


Funktionsbeschreibung:

Mit diesem Befehl kann die momentane Sollposition der aktuellen Bewegung gelesen werden.

Beschreibung der beteiligten Ein- und Ausgangelemente:

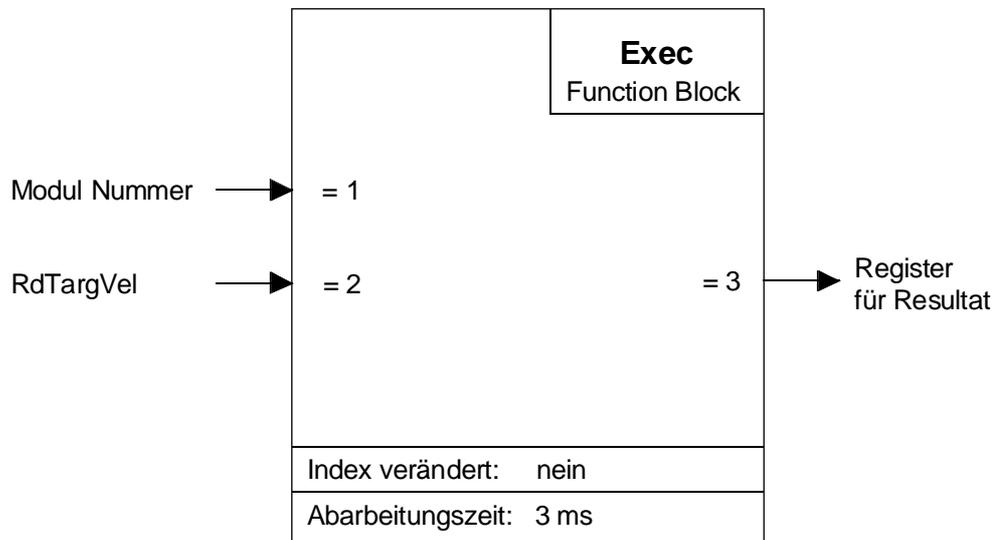
Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdTargPos				
= 3	PCD-Register für die Ziel-Position	R	integer	31 Bit	$-2^{30} .. +(2^{30} - 1)$



Es handelt sich um die errechnete Sollposition in Einheiten der Abtastzeit (341 μ s).

RdTargVel**Befehl:** Read Target Velocity (Lesen der Soll-Geschwindigkeit)

[11]

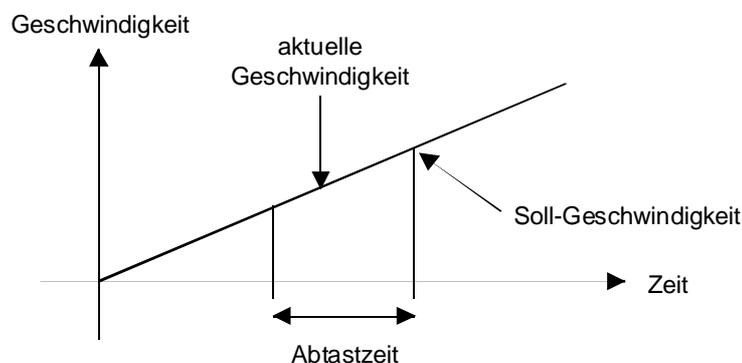
**Funktionsbeschreibung:**

Mit diesem Befehl kann die Soll-Geschwindigkeit des Systems gelesen werden. Der gelesene Wert entspricht dem im nächsten Abtastzyklus zu erreichende Wert.

Der Befehl ist jedes Mal aufzurufen, wenn die Soll-Geschwindigkeit gelesen werden soll.

Beschreibung der beteiligten Ein- und Ausgangelemente:

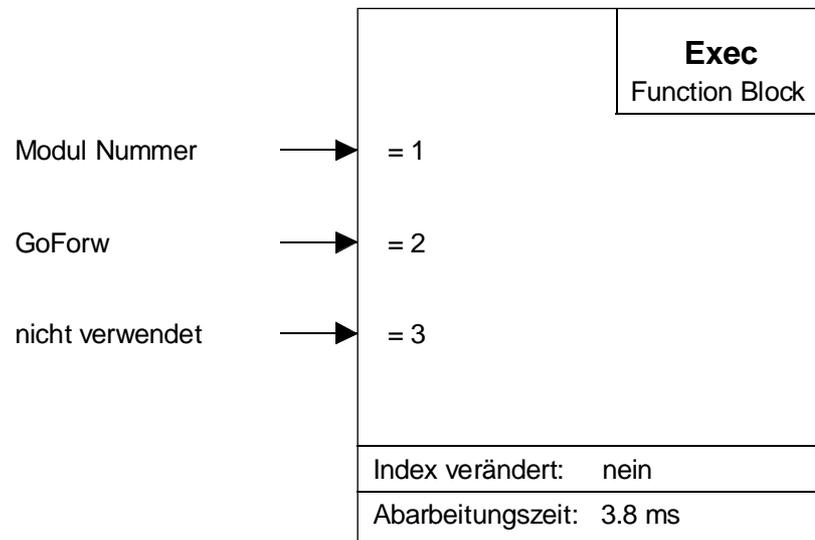
Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdTargVel				
= 3	PCD-Register für die Ziel-Geschwindigkeit	R	integer	30 Bit	abhängig vom Maschinenfaktor



Es handelt sich um die errechnete Sollgeschwindigkeit in Einheiten der Abtastzeit (341 μ s).

GoForw**Befehl:** Go Forwards (Bewegung vorwärts)

[12]

**Funktionsbeschreibung:**

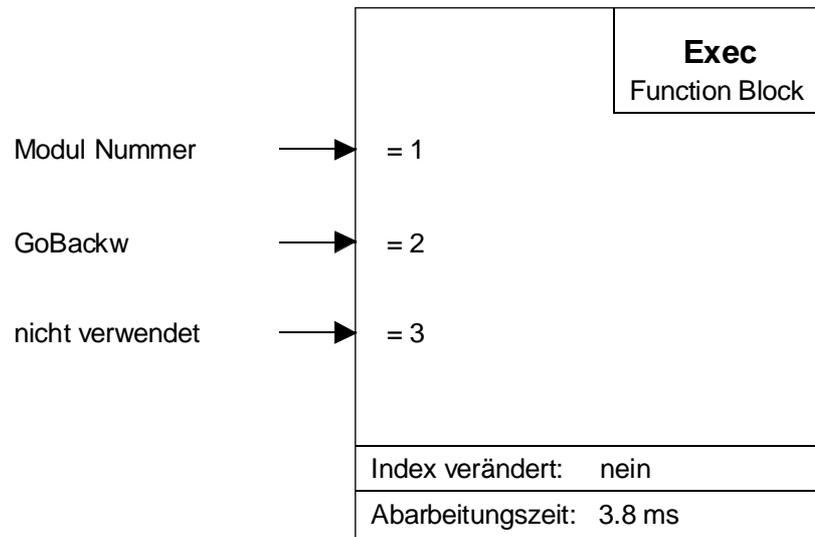
Mit diesem Befehl wird eine Vorwärtsbewegung mit der vordefinierten Geschwindigkeit und Beschleunigung, jedoch ohne Zielposition gestartet. Es wird mit diesem Befehl auch der 'drehzahlgeregelte Betrieb' gewählt.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: GoForw				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

GoBackw**Befehl:** Go Backwards (Bewegung rückwärts)

[13]

**Funktionsbeschreibung:**

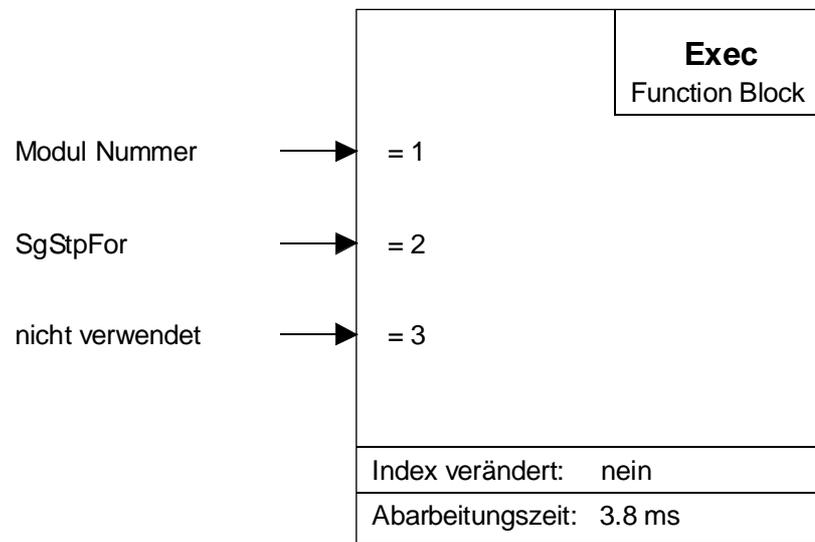
Mit diesem Befehl wird eine Rückwärtsbewegung mit der vordefinierten Geschwindigkeit und Beschleunigung, jedoch ohne Zielposition gestartet. Es wird mit diesem Befehl auch der 'drehzahlgeregelte Betrieb' gewählt.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: GoBackw				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

SgStpFor **Befehl:** Single Step Forwards (Einzelschritt vorwärts)

[14]



Funktionsbeschreibung:

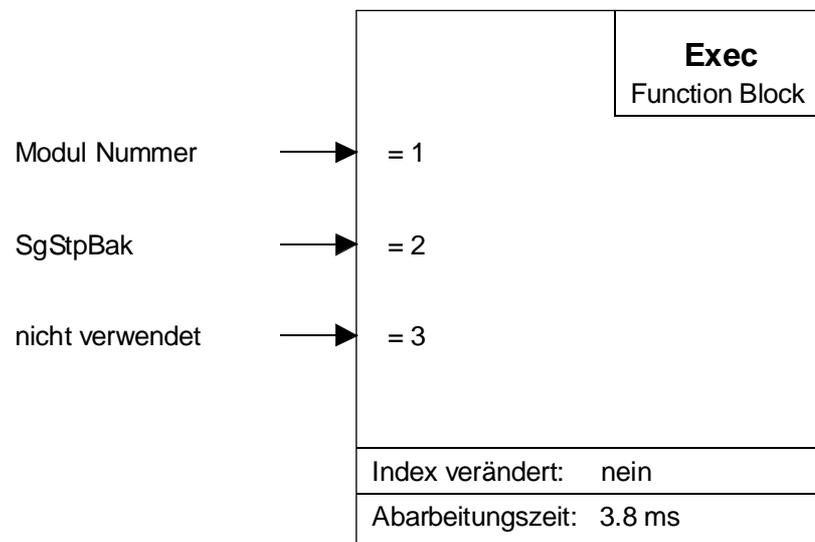
Mit diesem Befehl kann ein Einzelschritt der Encoderauflösung mit der vordefinierten Geschwindigkeit und Beschleunigung vorwärts geschaltet werden

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: SgStpFor				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

SgStpBak**Befehl:** Single Step Backwards (Einzelschritt rückwärts)

[15]

**Funktionsbeschreibung:**

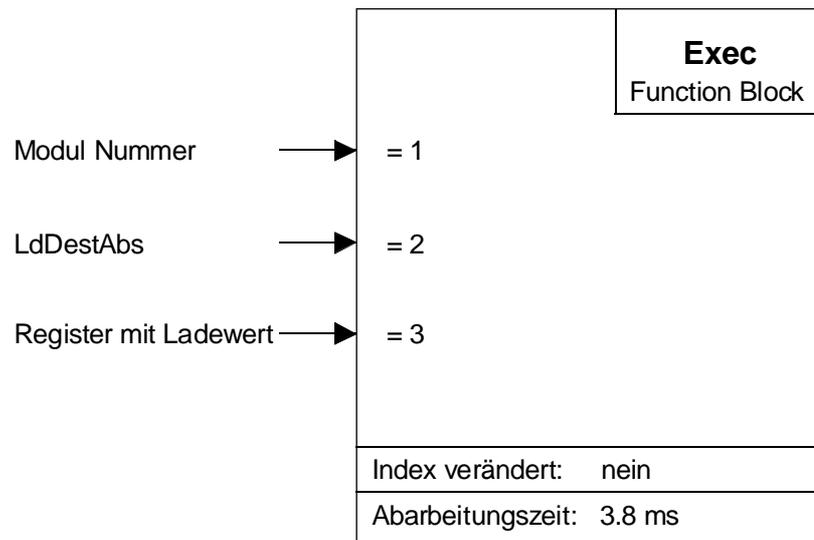
Mit diesem Befehl kann ein Einzelschritt der Encoderauflösung mit der vordefinierten Geschwindigkeit und Beschleunigung rückwärts geschaltet werden

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: SgStpBak				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

LdDestAbs **Befehl:** Load Destination Absolute (Laden der absoluten Ziel-Position)

[16]

**Funktionsbeschreibung:**

Mit diesem Befehl wird die absolute Ziel-Position geladen. Um die Funktion zu aktivieren, muss ein Befehl 'StartMot' folgen.

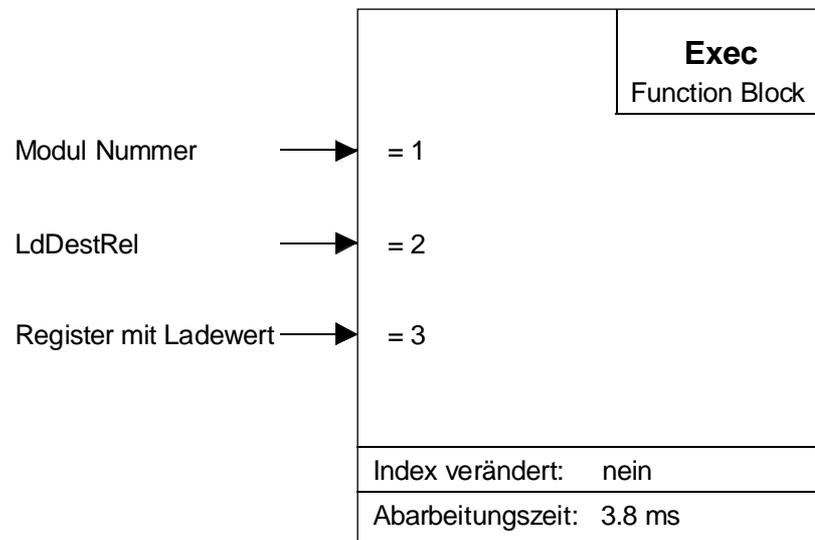
Die absolute Ziel-Position bezieht sich immer auf die Null-Position.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdDestAbs				
= 3	PCD-Register mit dem Ladewert	R	integer	31 Bit	$-2^{30} .. +(2^{30} - 1)$
fOnDest_x	Flag on destination	F			

LdDestRel **Befehl: Load Destination Relative (Laden der relativen Ziel-Position)**

[17]

**Funktionsbeschreibung:**

Mit diesem Befehl wird die relative Ziel-Position geladen. Um die Funktion zu aktivieren, muss ein Befehl 'StartMot' folgen.

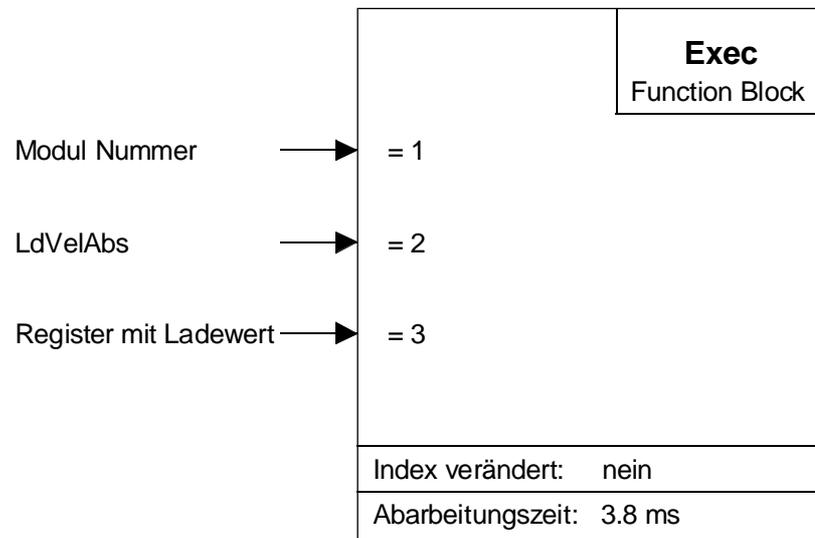
Die relative Ziel-Position bezieht sich immer auf die aktuelle Position.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdDestRel				
= 3	PCD-Register mit dem Ladewert	R	integer	31 Bit	$-2^{30} \dots +(2^{30} - 1)$
fOnDest_x	Flag on destination	F			

LdVelAbs **Befehl:** Load Velocity Absolute (Laden der absoluten Geschwindigkeit)

[18]

**Funktionsbeschreibung:**

Mit diesem Befehl kann die absolute Geschwindigkeit geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'StartMot' folgen.

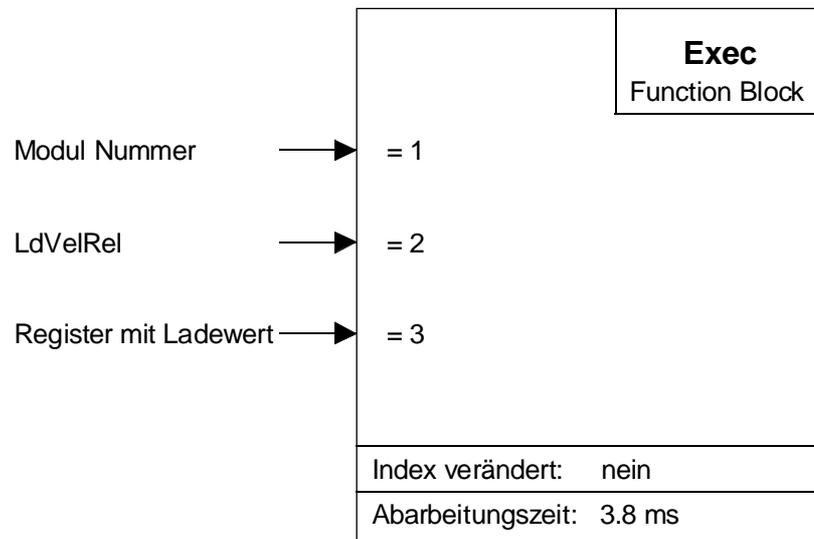
Die absolute Geschwindigkeit bezieht sich immer auf die Geschwindigkeit Null.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdVelAbs				
= 3	PCD-Register mit dem Ladewert	R	integer	30 Bit	abhängig vom Maschinenfaktor

LdVelRel**Befehl:** Load Velocity Relative (Laden der relativen Geschwindigkeit)

[19]

**Funktionsbeschreibung:**

Mit diesem Befehl kann die relative Geschwindigkeit geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'StartMot' folgen.

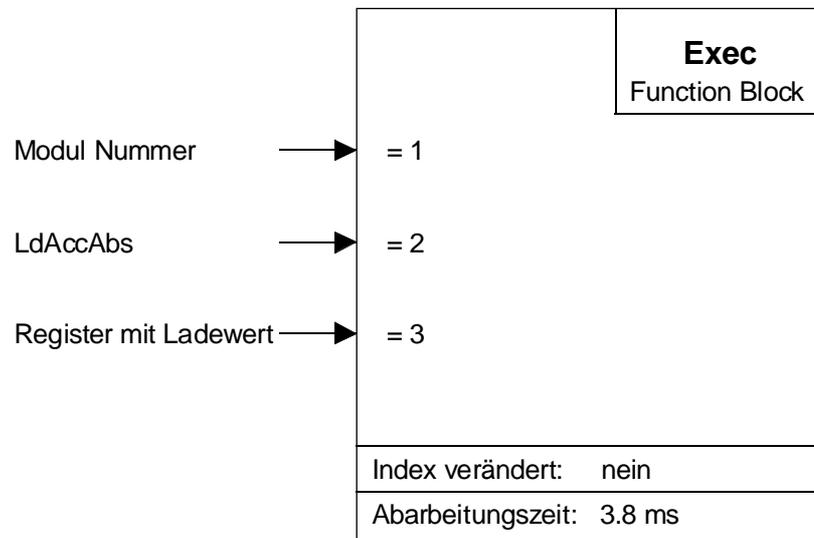
Die relative Geschwindigkeit bezieht sich immer auf die aktuelle Geschwindigkeit.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdVelRel				
= 3	PCD-Register mit dem Ladewert	R	integer	30 Bit	abhängig vom Maschinenfaktor

LdAccAbs **Befehl:** Load Acceleration Absolute (Laden der absoluten Beschleunigung)

[20]

**Funktionsbeschreibung:**

Mit diesem Befehl kann die absolute Beschleunigung geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'StartMot' folgen.

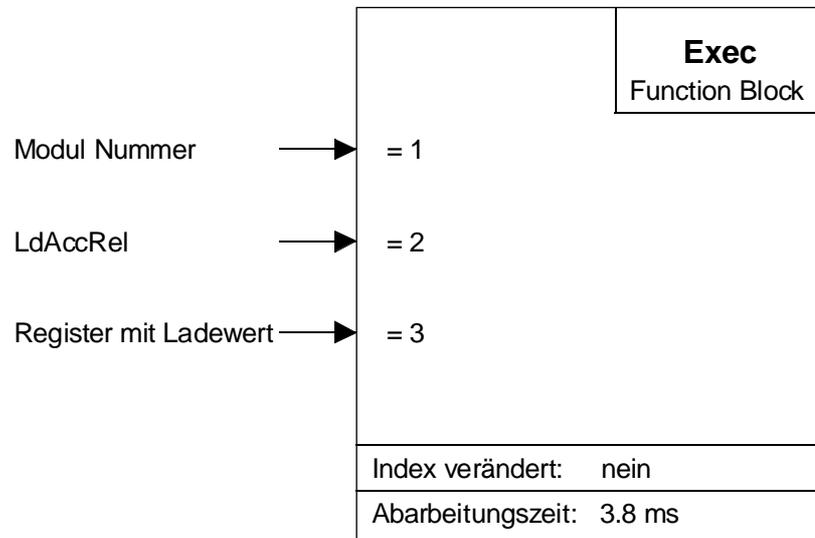
Die absolute Beschleunigung bezieht sich immer auf die Geschwindigkeit bzw. die Beschleunigung Null.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdAccAbs				
= 3	PCD-Register mit dem Ladewert	R	integer	30 Bit	abhängig vom Maschinenfaktor

LdAccRel **Befehl:** Load Acceleration Relative(Laden der relativen Beschleunigung)

[21]

**Funktionsbeschreibung:**

Mit diesem Befehl kann die relative Beschleunigung geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'StartMot' folgen.

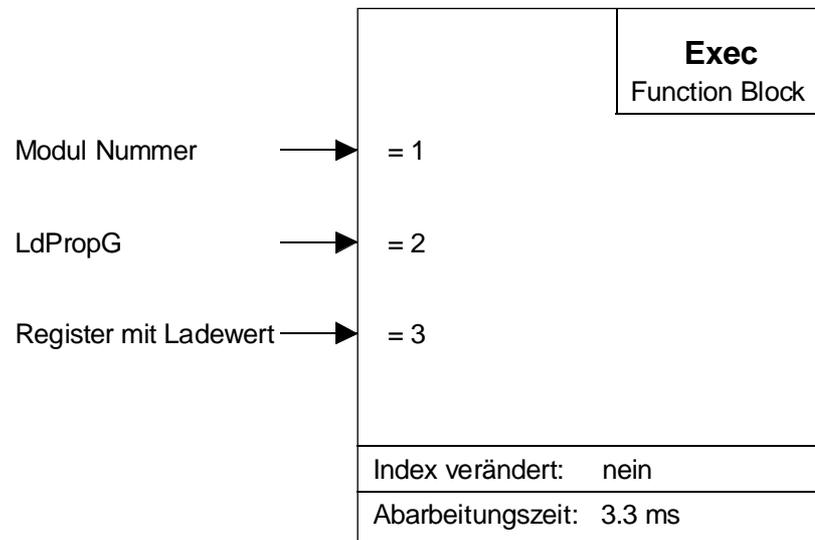
Die relative Beschleunigung bezieht sich immer auf die aktuelle Beschleunigung.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdAccRel				
= 3	PCD-Register mit dem Ladewert	R	integer	30 Bit	abhängig vom Maschinenfaktor

LdPropG**Befehl:** Load Proportional Gain (Laden des Proportionalfaktors)

[22]

**Funktionsbeschreibung:**

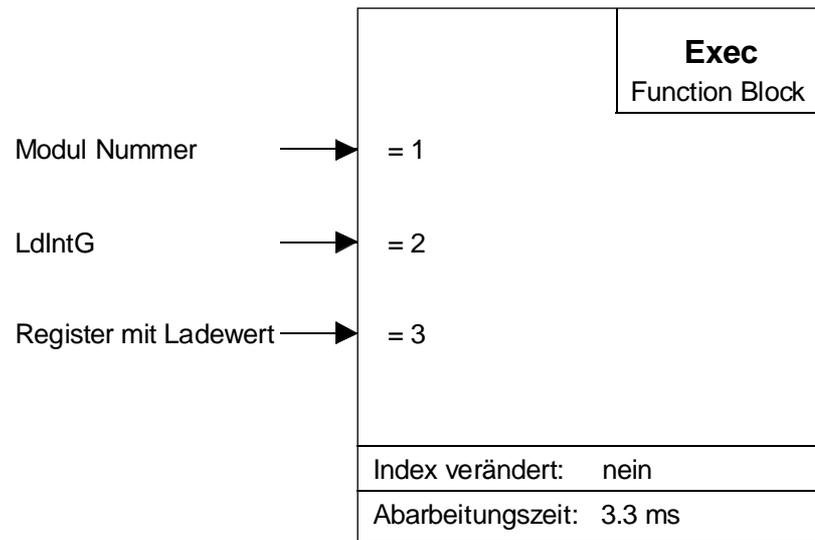
Mit diesem Befehl kann der Proportionalfaktor geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'ActRegFact' folgen.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdPropG				
= 3	PCD-Register mit dem Ladewert	R	integer	15 Bit	0 .. 32767

LdIntG**Befehl:** Load Integrative Gain (Laden des Integralfaktors)

[23]

**Funktionsbeschreibung:**

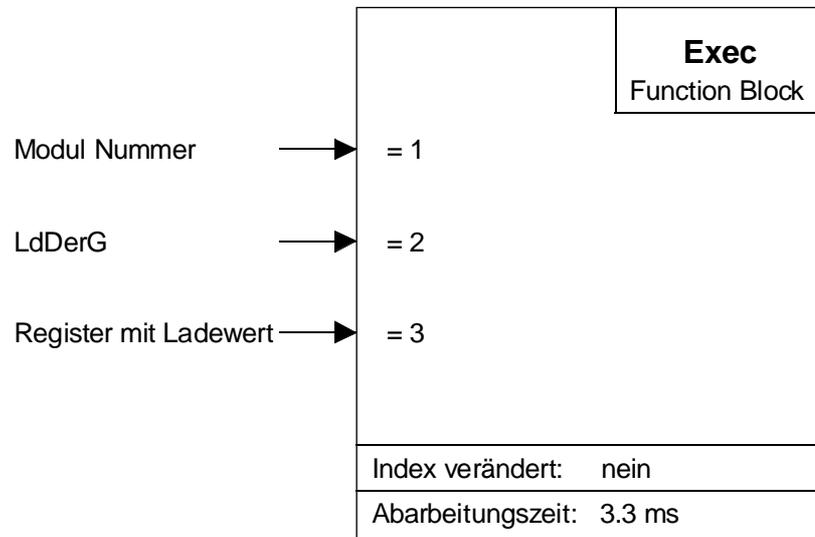
Mit diesem Befehl kann der Integralfaktor geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'ActRegFact' folgen.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdIntG				
= 3	PCD-Register mit dem Ladewert	R	integer	15 Bit	0 .. 32767

LdDerG**Befehl:** Load Derivative Gain (Laden des Differentialfaktors)

[24]

**Funktionsbeschreibung:**

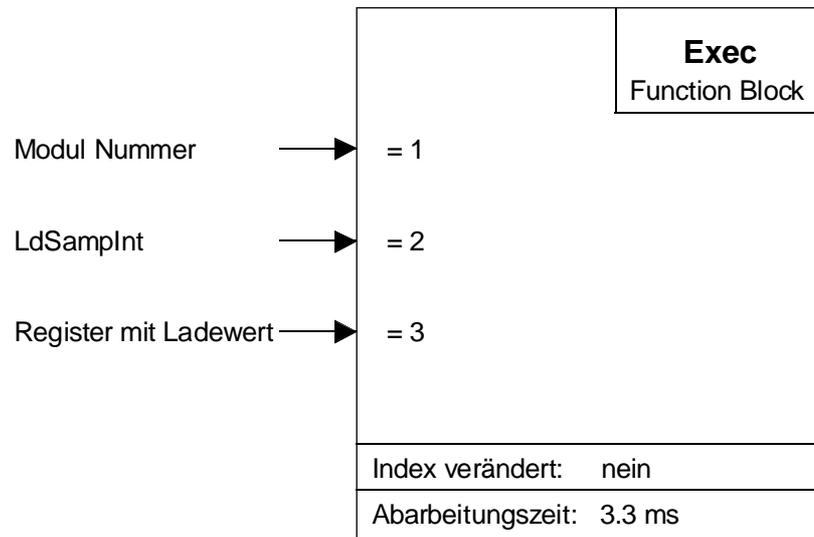
Mit diesem Befehl kann der Differentialfaktor geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'ActRegFact' folgen.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdDerG				
= 3	PCD-Register mit dem Ladewert	R	integer	15 Bit	0 .. 32767

LdSamplnt**Befehl:** Load Sampling Interval (Laden der Abtastzeit)

[25]

**Funktionsbeschreibung:**

Mit diesem Befehl kann die Abtastzeit des PID-Reglers geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'ActRegFact' folgen.

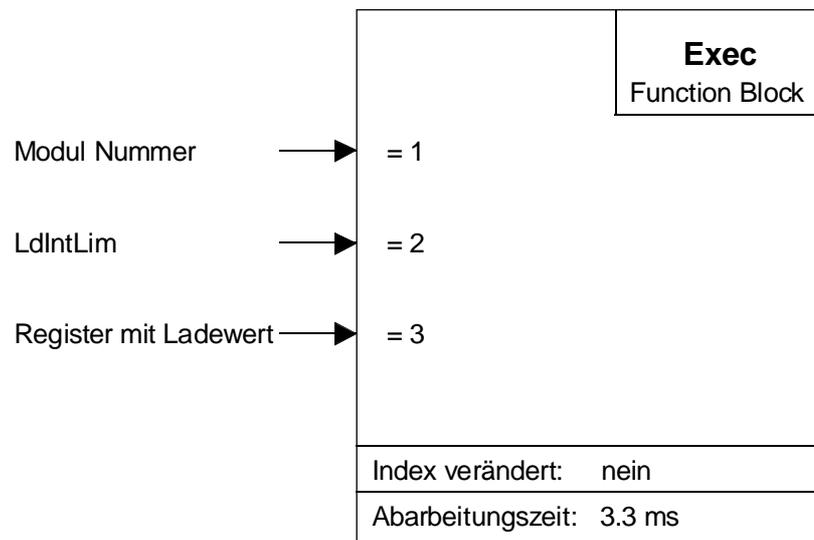
Die ursprüngliche Abtastzeit (341 μ s), welche als Faktor 'P' und 'T' verwendet werden sind für den D-Faktor normalerweise zu klein. Der Differentialfaktor arbeitet mit der Differenz der Errorsignals zwischen zwei Abtastsignalen, 341 μ s sind deshalb zu klein, um eine wirkliche Differenz zu haben. Die Erfahrung zeigt dass die Abtastzeit im allgemeinen zwischen 2 und 9 ms liegt, was Werten von 5 bis 25 entspricht.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdSampInt				
= 3	PCD-Register mit dem Ladewert	R	integer	8 Bit	0 .. 255

LdIntLim**Befehl:** Load Integrative Limit (Laden der Integrationsgrenze)

[26]

**Funktionsbeschreibung:**

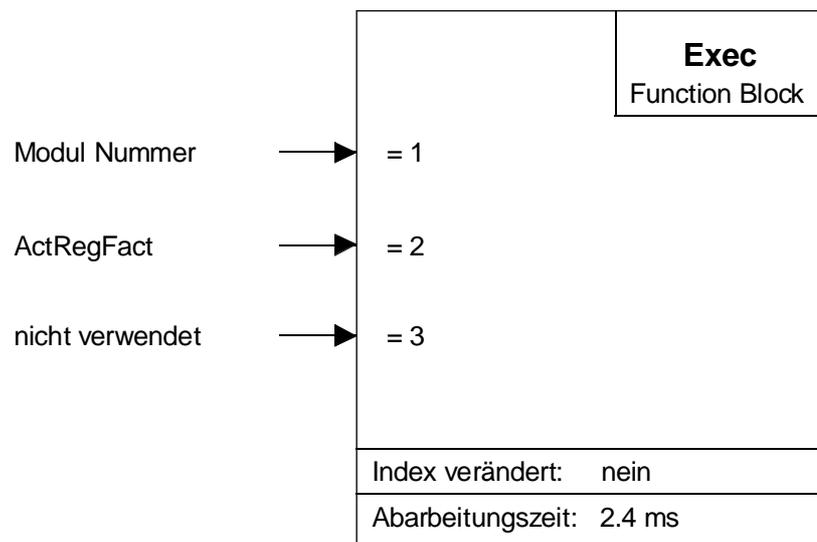
Mit diesem Befehl kann die Integrationsgrenze geladen bzw. verändert werden. Um die Funktion zu aktivieren, muss ein Befehl 'ActRegFact' folgen.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdIntLim				
= 3	PCD-Register mit dem Ladewert	R	integer	15 Bit	0 .. 32767

ActRegFact**Befehl:** Activate Regulation Factors (Aktivieren der Regelfaktoren)

[27]

**Funktionsbeschreibung:**

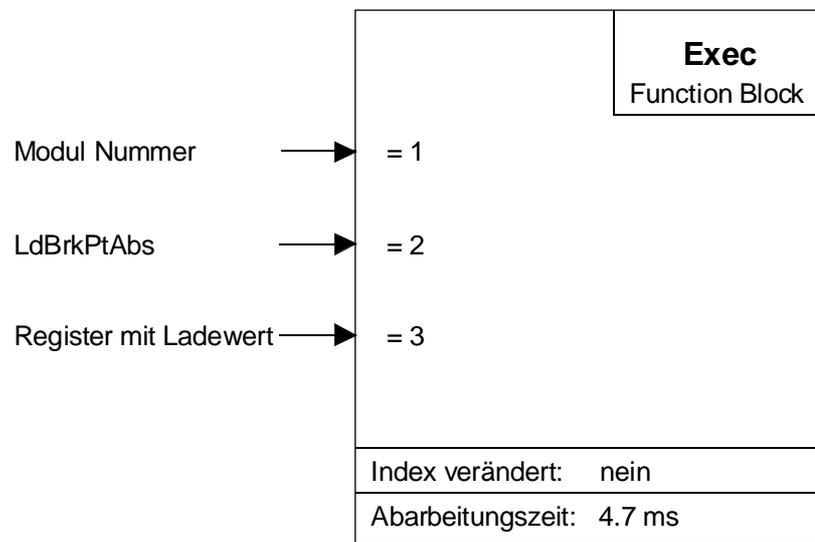
Mit diesem Befehl können alle Faktoren der PID-Regelung gleichzeitig aktiviert werden. Es kann auch nur ein einzelner Parameter aktiviert werden.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: ActRegFact				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

LdBrkPtAbs **Befehl:** Load Break Point Absolute (Laden eines absoluten Breakpoints)

[28]

**Funktionsbeschreibung:**

Mit diesem Befehl kann ein absoluter Breakpoint (Signalisation einer Zwischenposition) geladen werden. Die Absolut-Position bezieht sich immer auf die Null-Position.

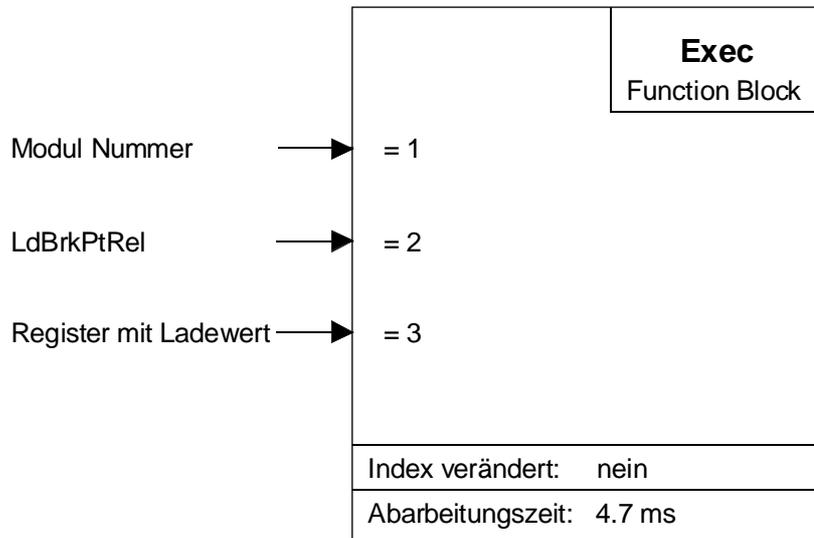
Wird der Breakpoint erreicht, wird das Flag 'fBrkPt_x' gesetzt. Dieses Flag jedoch wird erst nach der Abarbeitung des FB 'RdStatRg' aktiv.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdBrkPtAbs				
= 3	PCD-Register mit dem Ladewert	R	integer	31 Bit	$-2^{30} .. +(2^{30} - 1)$
fBrkPt_x	Flag 'Breakpoint'	F			Dieses Flag wird mit dem Befehl 'StartMot' oder 'ResStatRg' rückgesetzt.

LdBrkPtRel **Befehl:** Load Break Point Relative (Laden eines relativen Breakpunktes)

[29]

**Funktionsbeschreibung:**

Mit diesem Befehl kann ein relativer Breakpoint (Signalisation einer Zwischenposition) geladen werden. Die Relativ-Position bezieht sich immer auf die aktuelle Position.

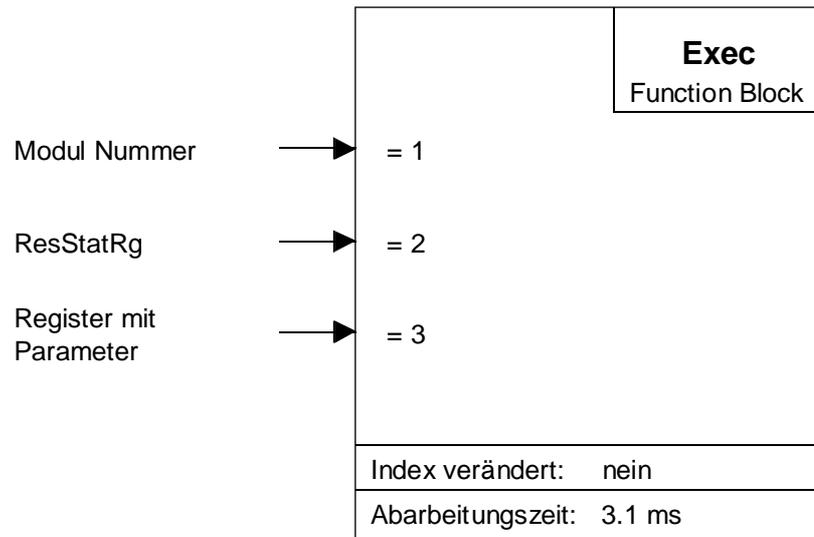
Wird der Breakpoint erreicht, wird das Flag 'fBrkPt_x' gesetzt. Dieses Flag jedoch wird erst nach der Abarbeitung des FB 'RdStatRg' aktiv.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdBrkPtRel				
= 3	PCD-Register mit dem Ladewert	R	integer	31 Bit	$-2^{30} .. +(2^{30} - 1)$
fBrkPt_x	Flag 'Breakpoint'	F			Dieses Flag wird mit dem Befehl 'StartMot' oder 'ResStatRg' rückgesetzt.

ResStatRg**Befehl:** Reset Status Register (Rücksetzen des Statusregisters)

[30]

**Funktionsbeschreibung:**

Mit diesem Befehl können alle oder einzelne Parameter, welche im Statusregister enthalten sind, zurückgeschaltet werden.

Statusregister:

Bit 7: Motor off	128
Bit 6: Breakpoint reached	64
Bit 5: Excessive position error	32
Bit 4: Wraparound occurred	16
Bit 3: Index pulse observed	8
Bit 2: Trajectory complete	4
Bit 1: Internal use only	2
Bit 0: Internal use only	1

	255

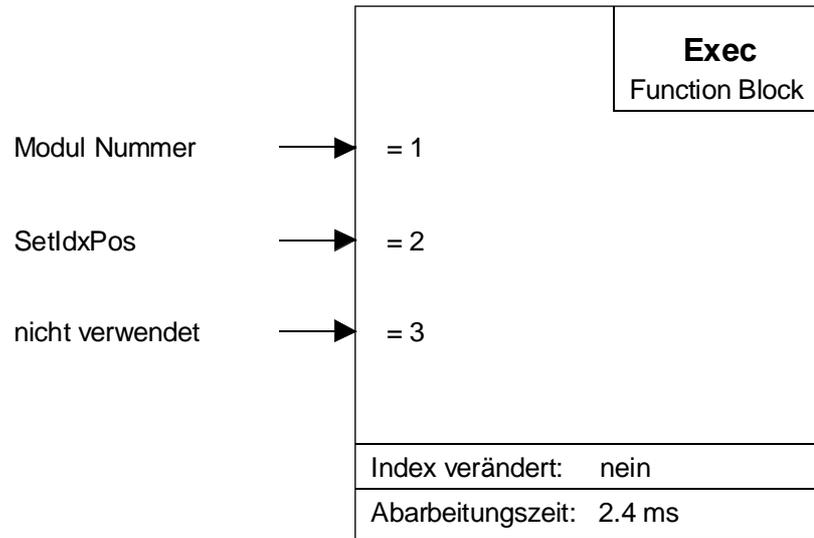
Zum Rücksetzen des gesamten Status-Registers ist das Register für den Parameter 3 mit dem Wert 0 zu laden.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: ResStatRg				
= 3	PCD-Register mit Rücksetzinformation	R	integer	8 Bit	0 .. 255
fBrkPt_x	Flag breakpoint	F			
fOnDest_x	Flag on destination	F			
fPosErr_x	Flag position error	F			

SetIdxPos**Befehl:** Set Index Position (Setzen der Indexposition)

[31]

**Funktionsbeschreibung:**

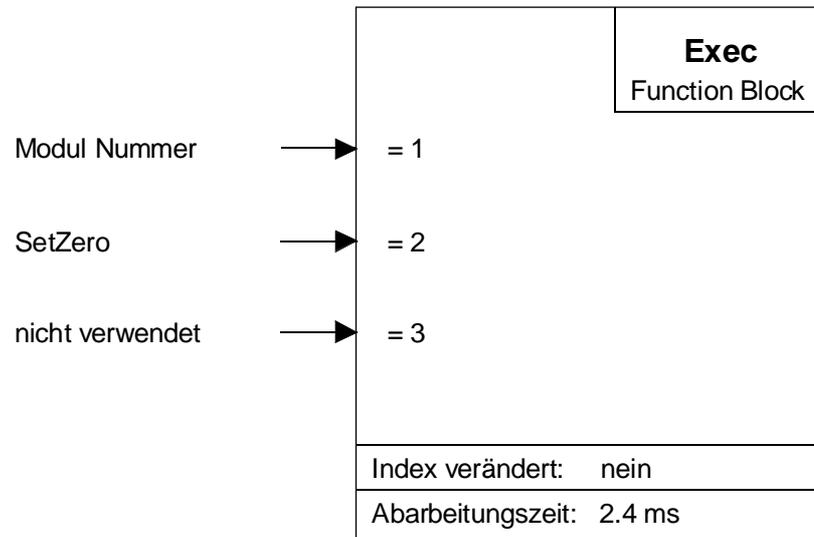
Mit diesem Befehl wird die absolute Indexposition beim Eintreffen des nächsten Indeximpulses vom Encoder ins Indexregister des Moduls geladen. Die Position wird übernommen, wenn beide Encodereingänge 'A' und 'B' und der Index-Eingang 'IN' = L sind. Die 'Home'-Prozedur verwendet diese Funktion zur Vereinfachung der Definition der 'Home'-Funktion.

Beschreibung der beteiligten Ein- und Ausgangelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: SetIdxPos				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

SetZero**Befehl:** Set Zero position (Setzen der Home- (Null) Position)

[32]

**Funktionsbeschreibung:**

Mit diesem Befehl kann irgend eine Position auf Null gesetzt werden. Die Psoition wird dann zur neuen 'Home'-Position.

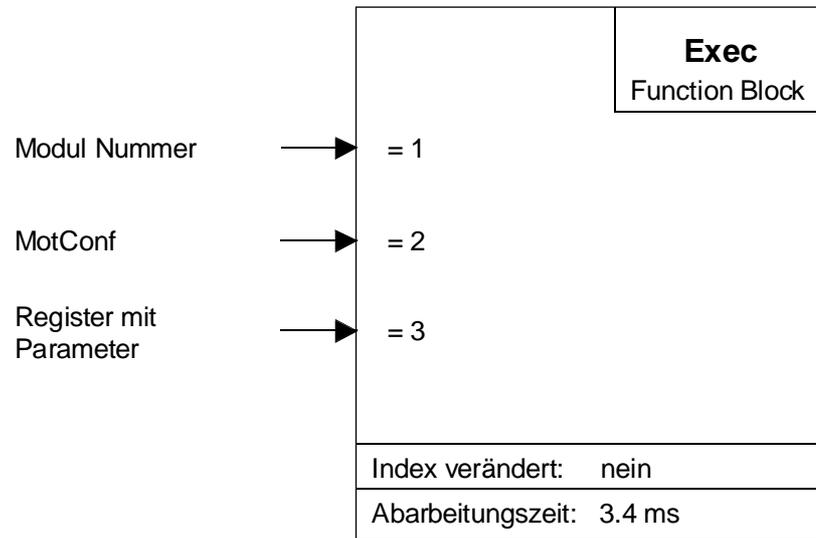
Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: SetZero				
= 3	Leeres PCD-Register bzw. 'rNotUsed'	R			

MotConf

Befehl: Motion Configuration (Konfigurierung Drehzahl-
bzw. Positionsgeregelter Betrieb)

[33]

**Funktionsbeschreibung:**

Mit diesem Befehl wird die Bewegung konfiguriert:

Bit 0: L = Drehzahl geregelter Betrieb
H = Positionsgeregelter Betrieb

Bit 1: Wirksam nur im positionsgeregelten Betrieb:
L = rückwärts
H = vorwärts

Standardmässig ist "positionsgeregelter Betrieb" gewählt.

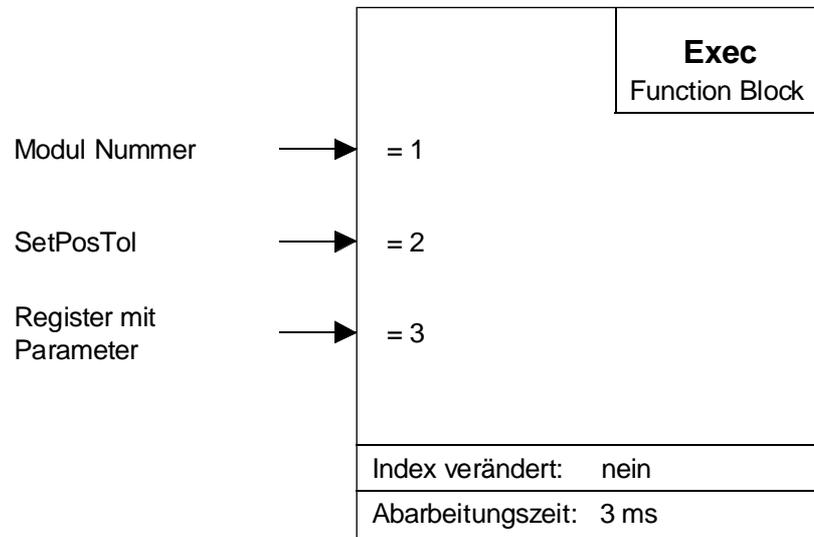
(Dieser Befehl ist nicht notwendig, wurde jedoch der Vollständigkeit halber belassen. Der Positions-Modus wird automatisch mit den Befehl 'StartMot' und der Geschwindigkeits-Modus mit den Befehlen 'GoForw' bzw. 'GoBackw' gewählt.)

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: Mot Conf				
= 3	PCD-Register für Konfiguration	R	integer	2 Bit	0 .. 3

SetPosTol**Befehl:** Set Position Tolerance (Setzen der Positionstoleranz)

[34]

**Funktionsbeschreibung:**

Mit diesem Befehl wird der maximal zulässige Positionsfehler zwischen der aktuellen und der Zielposition definiert, welcher durch den Prozessor für jede Abtastintervall berechnet wird. Das Verhalten im Falle eines übermässigen Fehlers ist im FB 'Init' mit Parameter 8 definiert. Es wird hier angegeben, ob der Motor gestoppt oder nur das 'fPosErr_x'-Flag gesetzt werden soll.

Bevor das 'fPosErr_x'-Flag ausgewertet werden kann, ist der FB 'RdStatRg' abzuarbeiten, da dieser die Prozedur des Positions-Error-Flags behandelt. Diese Flag wird beim nächsten 'StartMot'-Befehl zurück gesetzt.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: Set PosTol				
= 3	PCD-Register für Positionstoleranz	R	integer	15 Bit	0 .. 32767
fPosErr_x	Positions-Error-Flag				

Anhang B: Zusammenfassung aller FBoxen für die Programmierung in FUPLA

In Vorbereitung

Notizen

Absender:

Firma
Abteilung
Name
Adresse

Tel.

Datum

An:

SAIA-Burgess Electronics AG
Bahnhofstrasse 18
CH-3280 Murten (Schweiz)
<http://www.saia-burgess.com>

GB: Electronic Controllers

Handbuch PCD2.H31x

Falls Sie Vorschläge zu SAIA[®] PCD zu machen oder Fehler in diesem Handbuch gefunden haben, sind wir Ihnen für einen kurzen Bericht dankbar.

Ihre Vorschläge: