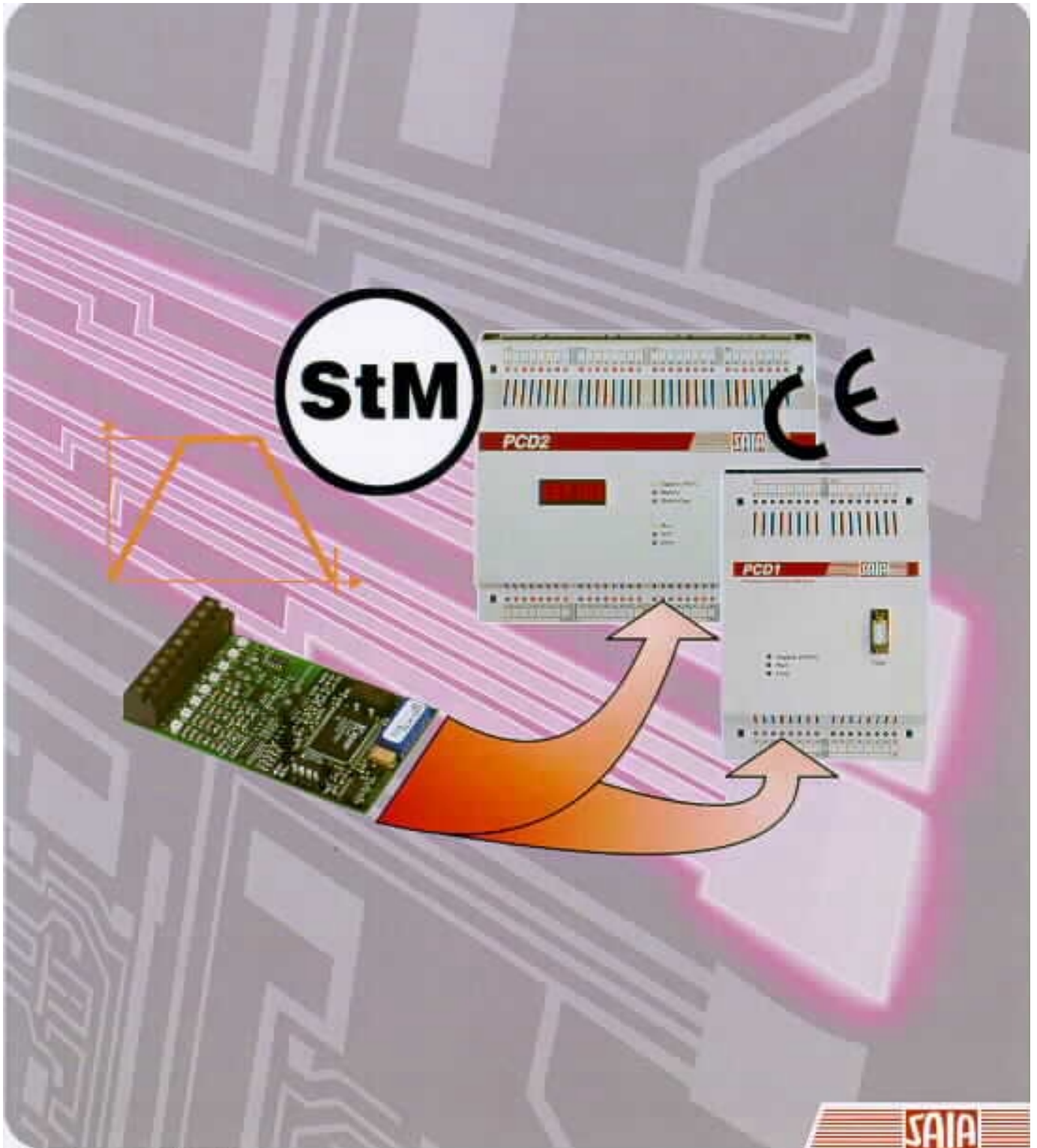


SAIA® PCD
Process Control Devices

PCD2.H210
Positioniermodul
für Schrittmotoren



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

Positioniermodul für Schrittmotoren

PCD2.H210

SAIA-Burgess Electronics AG 1999. Alle Rechte vorbehalten
Ausgabe 26/760 D2 - 05.99

Technische Änderungen vorbehalten

Anpassungen

Handbuch: PCD2.H210 - Positioniermodul für Schrittmotoren - Ausgabe D2

Datum	Abschnitt	Seite	Beschreibung
15.05.2000	7.1.2	7-5/6	Beschreibung: Nullposition
15.05.2000	Anhang A	A-13	Offset zur Referenzposition

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
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 Blockschaltbild des Moduls	5-1
5.2 Beschreibung des Moduls	5-2
5.3 Ergänzende Information: Das Frequenz-Profil	5-6
5.4 Ergänzende Information: Referenzfahren (FB Home)	5-7
6. Der schnelle Einstieg	
6.1 Einstieg mit Programmierung in IL	6-2
6.2 Einstieg mit Programmierung in FUPLA	6-5

	Seite
7. Programmierung	
7.1 Programmierung in IL mit FBs	7-2
7.1.1 Das IL-Paket	7-2
7.1.2 Die einzelnen FBs	7-5
7.1.3 Bestimmung der Parameter	7-7
7.2 Programmierung in FUPLA mit FBoxen	7-9
7.3 Programmierung in GRAFTEC mit FBoxen	7-10
8. Fehler-Behandlung und Diagnose	
8.1 Definitionsfehler durch Assembler überprüft	8-1
8.2 Fehlerbehandlung in RUN	8-2
8.2.1 Falsche Parameter	8-2
8.2.2 Fehler während einer Referenzfahrt	8-3
9. Anwendungsbeispiele	
9.1 Typisches Anwendungsbeispiel in GRAFTEC und IL	9-1
9.2 Typisches Anwendungsbeispiel in reinem FUPLA	9-
9.3 Typisches Anwendungsbeisp. in GRAFTEC und FUPLA	9-
Anhang A: Zusammenfassung aller Funktions-Blocks (FB)	
Init: Initialisierungs-FB	A-1
Exec: Ausführungs-FB	A-2
LdAcc: Laden der Bescheinigung	A-3
LdVmin: Laden der Start-Stop Frequenz	A-4
LdVmax: Laden der Maximalgeschwindigkeit	A-5
LdDestRel: Laden des relativen Fahrweges	A-6
LdDesrAbs: Laden der absoluten Endposition	A-7
RdPosition: Lesen der aktuellen Position	A-8
Start: Starten einer Bewegung	A-9
Stop: Stoppen einer Bewegung	A-9
Continue: Weiterlaufen lassen	A-10
RdIdent: Lesen der Modulidentifikation	A-11
SetOut2/3: Ausgang 2 resp. 3 setzen	A-12
ResOut2/3: Ausgang 2 resp. 3 rücksetzen	A-12
Home: Referenzpositions-FB	A-13
Anhang B: Zusammenfassung aller Function-Boxes (FBox)	



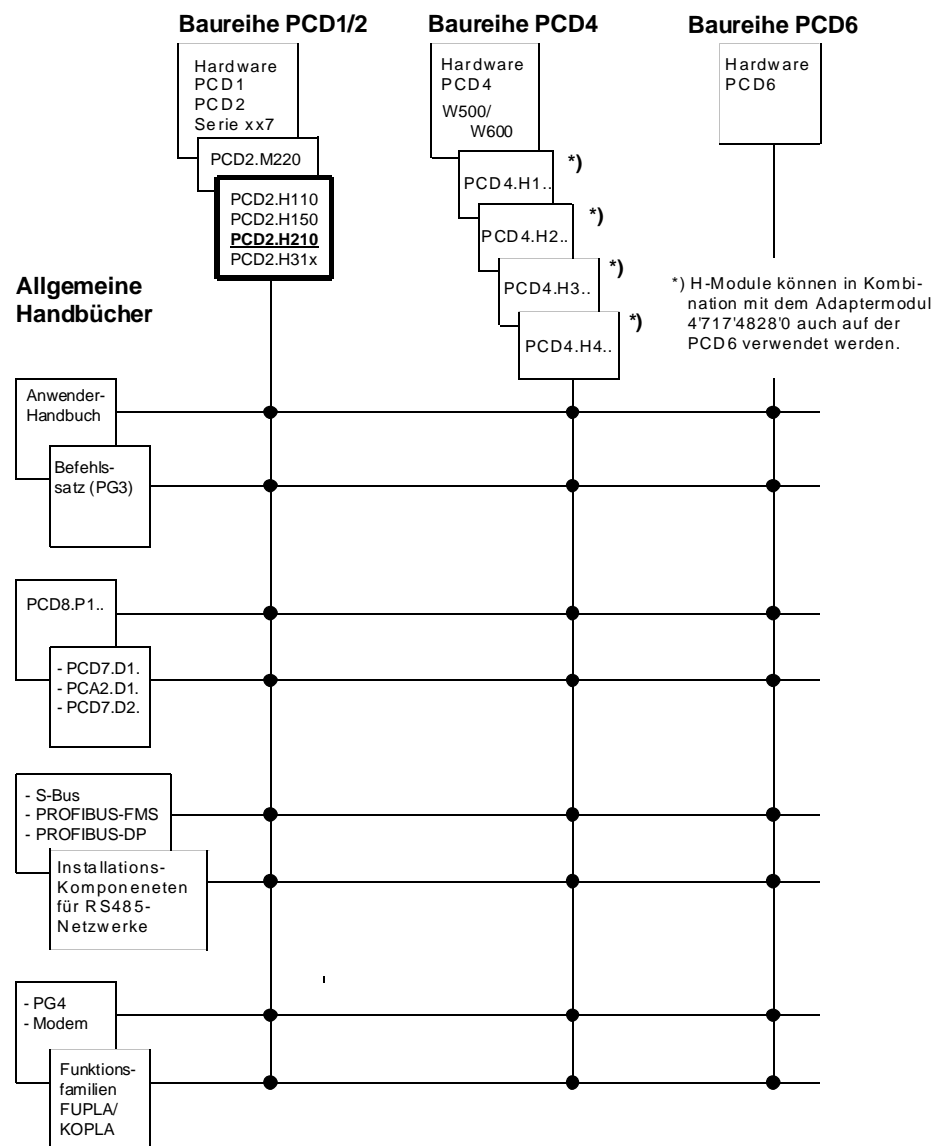
Wichtiger Hinweis:

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.

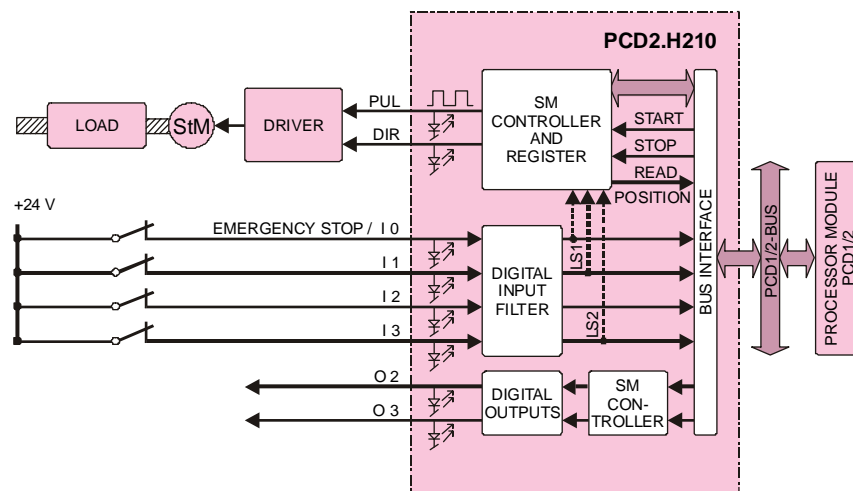
Werden alle diese Punkte berücksichtigt, verfügen Sie mit der SAIA® PCD über eine moderne und sichere programmierbare Steuerung, die Ihre Anlage über viele Jahre zuverlässig steuern, regeln und überwachen wird.

1. Einführung

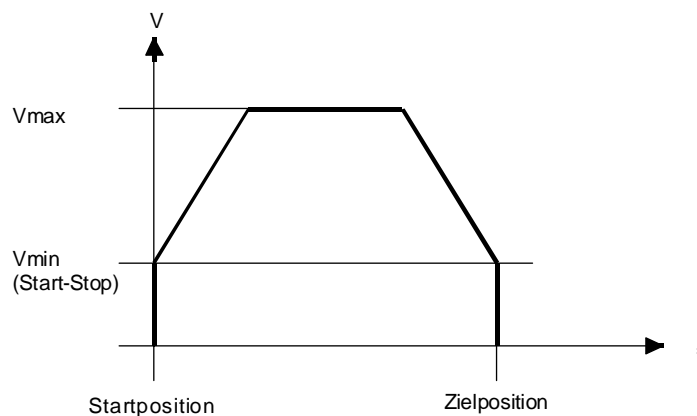
1.1 Allgemeines

Mit dem Modul PCD2.H210 kann die Steuerung und Überwachung des Bewegungsablaufes eines Schrittmotors mit Hochlauf- und Bremsrampe vollkommen autonom erfolgen. Jedes Modul ..H210 steuert eine unabhängige Achse und liefert eine einphasige Impulskette, welche einer geeigneten Ansteuerelektronik zugeführt wird.

Blockschaltbild eines Schrittmotorantriebes



Typisches Geschwindigkeitsprofil



1.2 Funktion und Anwendung

Dieses Low-Cost Modul lässt sich an jedem beliebigen E/A-Steckplatz einer PCD1 oder PCD2 einstecken. Es dient der Ansteuerung der Leistungsstufe zu einer Schrittmotorachse bis zu einer Frequenz von 19.454 kHz.

Mit der PCD2 lassen sich so max. 16 Achsen, mit einer PCD1 max. 4 Achsen steuern. Da jedes Modul ..H210 zusätzlich über 4 unabhängige digitale Eingänge und 2 digitale Ausgänge verfügt, stehen bei 16 Achsen der übrigen Prozess-Steuerung total weitere 64 digitale Eingänge und 32 digitale Ausgänge zur Verfügung.

Mit dem Modul ..H210 kann die Steuerung und Überwachung des Bewegungsablaufes eines Schrittmotors mit Hochlauf- und Bremsrampe vollkommen autonom erfolgen. Die erforderlichen Befehle für den Ablauf der Schrittmotorbewegungen werden dem Modul über Funktionsbausteine im Anwenderprogramm übergeben. Während der Bewegung überwacht der SM-Prozessor das Frequenzprofil und die Beschleunigungs- und Bremsrampen, um die Achse ohne Schrittverluste auf die Ziel Position zu fahren. Jedes Modul ..H210 steuert eine unabhängige Achse. Mehrere Achsen können jedoch auch im koordinierten Quasi-Synchronbetrieb gestartet werden.

1.3 Die wichtigsten Eigenschaften

Kostengünstige Steuerung im "Open Loop"-Betrieb mit hoher Präzision und Zuverlässigkeit.

- Die Geschwindigkeit von 9.5 bis 19 454 Hz sowie die Beschleunigung und Anzahl Impulse werden quazgenau gesteuert.
- Jederzeit Zugriff zur aktuellen Achsposition bzw. zur Anzahl der ausgegebenen Impulse.
- Der Eingang E2 kann als gewöhnlicher digitaler Eingang 24 VDC gebraucht werden , wenn dieser nicht vom Funktionsbaustein "Home" benutzt wird.
- Weitere 3 Eingänge stehen zur Verfügung. Diese können bei der Initialisierung als gewöhnliche digitale Eingänge 24VDC, als Endschalter (E1 und E3, wobei beim Ansprechen ein Stop mit Bremsrampe erfolgt) und/oder als Notstop (E0) konfiguriert werden.
- 2 digitale Ausgänge 0.5 A, 24 VDC auf dem gleichen Modul sind für andere Prozessaufgaben verfügbar.
- Einfache Programmierung dank übersichtlicher Funktionsbausteine (FBs und FBoxen).

1.4 Typische Einsatzgebiete

- Handling- und Montageautomaten
- Pick- and Place-Funktionen
- Palettisierereinrichtungen
- Automatische Winkelsteuerung z. B. von Kameras, Scheinwerfern, Antennen usw.
- Allgemein zur Steuerung von Antrieben, die ein hohes Stillstand-Drehmoment erfordern
- Positionierung statischer Achsen (Set up)

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" entweder in IL (Instruction List) oder grafisch (FUPLA).

Initialisierungsbefehl

INIT Wahl der Modul-Nummer
 Wahl des Frequenzbereichs
 Aktivierung der Notstop-Funktion
 Aktivierung der Endschalter
 Eingabe von Vmin (Start-Stop Frequenz)
 Eingabe von Vmax
 Eingabe der Beschleunigung

Ausführungsbefehl

EXEC zum Laden des relativen Ziels
 zum Laden des absoluten Ziels
 zum Starten einer Impulsausgabe
 zum Stoppen (unterbrechen) einer Impulsausgabe
 zum Fortsetzen einer Impulsausgabe
 zum Lesen des Zählers bzw. der Position
 zum Lesen der Modulidentität
 zum Betätigen der frei verfügbaren Ausgänge

Einrichtungsbefehl

HOME zur automatischen Suche des Referenzschalters.

Diagnose und Fehlerbehandlung

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

2. Technische Daten

2.1 Technische Daten der Hardware

Digitale Eingänge

Anzahl Eingänge:	4
Nennspannung:	24V
Bereich "Low":	- 30 ... +5V
Bereich "High":	+15 ... +30V
Nur Quellbetrieb	aus Sicherheitsgründen sollten Öffner-Kontakte (normally closed bzw. negative Logik) verwendet werden.
Eingangsstrom (typisch):	6.5 mA
Schaltungsart	galvanisch verbunden
Eingangsfiter	< 1 ms

Digitale Ausgänge

Anzahl Ausgänge	4
Schaltleistung	je 0.5A im Bereich von 10 ... 32 VDC, Restwelligkeit max. 10%
Kurzschluss-Schutz	nein
Galvanische Trennung	nein
Spannungsabfall	max. 0.3V bei 0.5A
Logik	positiv (plus-schaltend)
Ausgangsverzögerung	kleiner als 1 μ s, bei induktiver Last länger, als Folge der Freilaufdiode

Stromversorgung

Interne Speisung ab PCD1/2-Bus	5 VDC, 20 ... 45 mA
Extern durch Anwender für alle Ausgänge	24VDC (10 ... 32 VDC), max. 2A, Restwelligkeit max. 10%

Betriebsbedingungen

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

Bestellangaben

PCD2.H210	Positionier-Modul für Schrittmotoren
PCD9.H21E	Software-Bibliothek mit Funktionsbausteinen

LED-Anzeigen

Anzahl 8

LED 0:	*) Spannung am Eingang 0:	(Notstop)
LED 1:	*) Spannung am Eingang 1:	(LS1)
LED 2:	*) Spannung am Eingang 2:	(REF)
LED 3:	*) Spannung am Eingang 3:	(LS2)
LED 4:	Spannung am Ausgang 0:	PUL
LED 5:	Spannung am Ausgang 1:	DIR
LED 6:	Spannung am Ausgang 2	
LED 7:	Spannung am Ausgang 3	

*) inverser Status bei Verwendung als Endschalter

Programmierung

Auf PCD-Anwenderprogramm (PG4) basierend und unterstützt von FB- und FBox-Bibliothek.

2.2 Elektrische Spezifikationen

Interne Leistungsaufnahme

+5V: 20 ... 45 mA
U_{ext} 0 ... 10 mA (ohne Laststrom)

Externe Speisung

Klemmen +/-: 10 ... 32 VDC geglättet, zulässige Welligkeit max. 10%
TVS-Diode 39V ±10%.
max. 2A für die Ausgänge, kein Schutz gegen falsche Polung.

Digitale Eingänge

4 digitale Eingänge (E0 ... E3)
(siehe Abschnitt 2.1)

Digitale Ausgänge

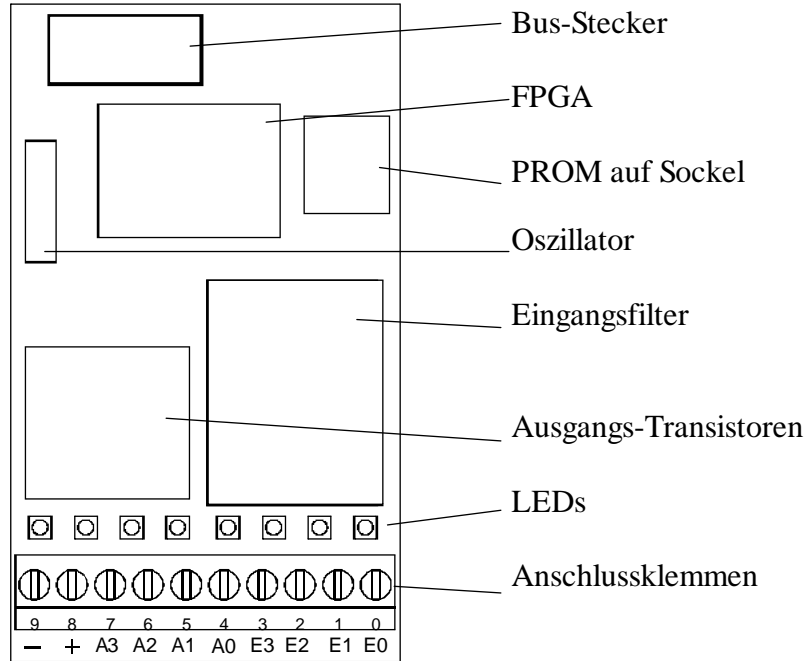
4 digitale Ausgänge (A0 ... A3)
(siehe Abschnitt 2.1)

2.3 Funktionsspezifische Daten

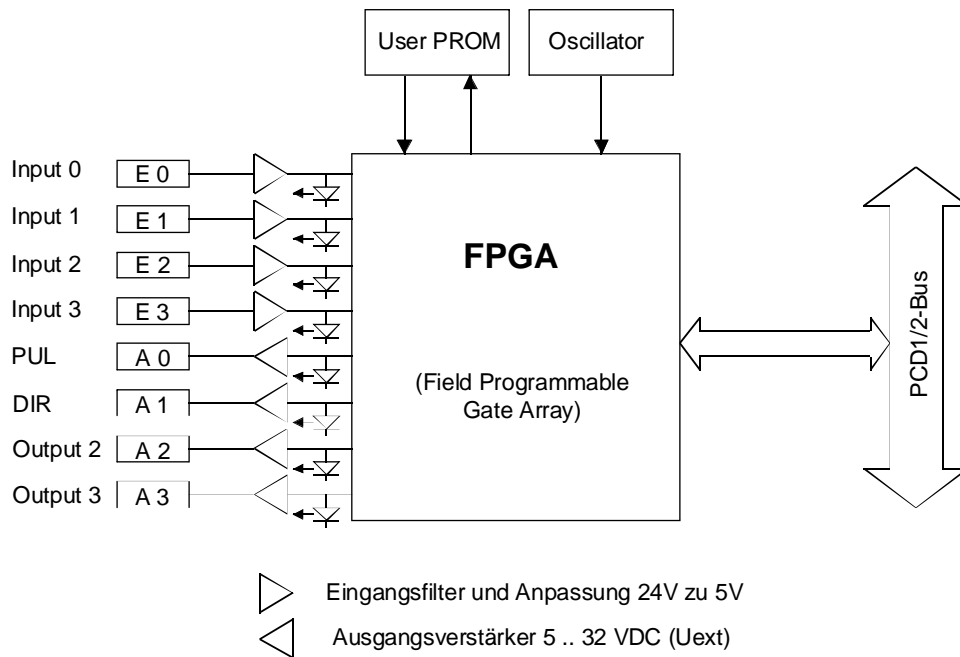
Anzahl Systeme	1
Positionierdistanz (Zählbereich)	0 bis 16'777'215 (24 Bit)
Frequenzbereiche (wählbar)	9.5 ... 2431 Hz 19 ... 4864 Hz 38 ... 9727 Hz 76 ... 19'454 Hz (siehe Tabelle im Abschnitt 7.1.3)
Beschleunigung	0.6 ... 1224 kHz/s nicht lineare Bereichsaufteilung, abhängig vom gewählten Fre- quenzbereich (siehe Tabelle in Ab- schnitt 7.1.3)
Profilgenerator	mit symmetrischen Beschleuni- gungs- und Bremsrampen
Datensicherung	Alle Daten dieses Moduls sind flüchtig (nicht flüchtige PCD- Register stehen zur Verfügung)

3. Präsentation

Bestücktes Modul



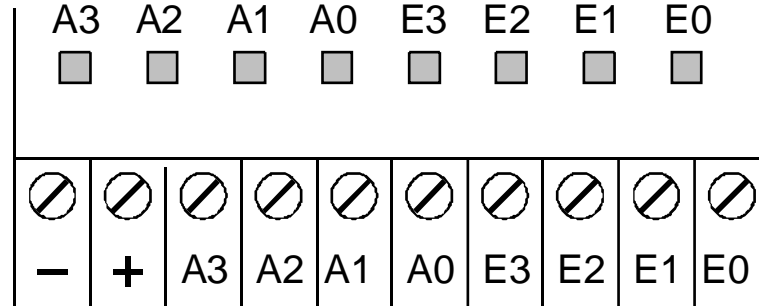
Einfaches Blockschaltbild



4. Anschlüsse und Bedeutung der LEDs

Anschlussklemmen und LEDs

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 =	E 0:	Konfigurierbar als Notstop resp. frei verwendbar
Klemme 1 =	E 1:	Konfigurierbar als Endschalter LS1 resp. frei verwendbar
Klemme 2 =	E 2:	Konfigurierbar als Referenzschalter resp. frei verwendbar
Klemme 3 =	E 3:	Konfigurierbar als Endschalter LS2 resp. frei verwendbar

Ausgänge:

Anzahl		4
Klemme 4 =	A 0:	Ausgang PUL (Impulse für den Motor)
Klemme 5 =	A 1:	Ausgang DIR (Drehrichtung des Motors)
Klemme 6 =	A 2:	frei verwendbar
Klemme 7 =	A 3:	frei verwendbar

Speisung:

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

LED-Anzeigen

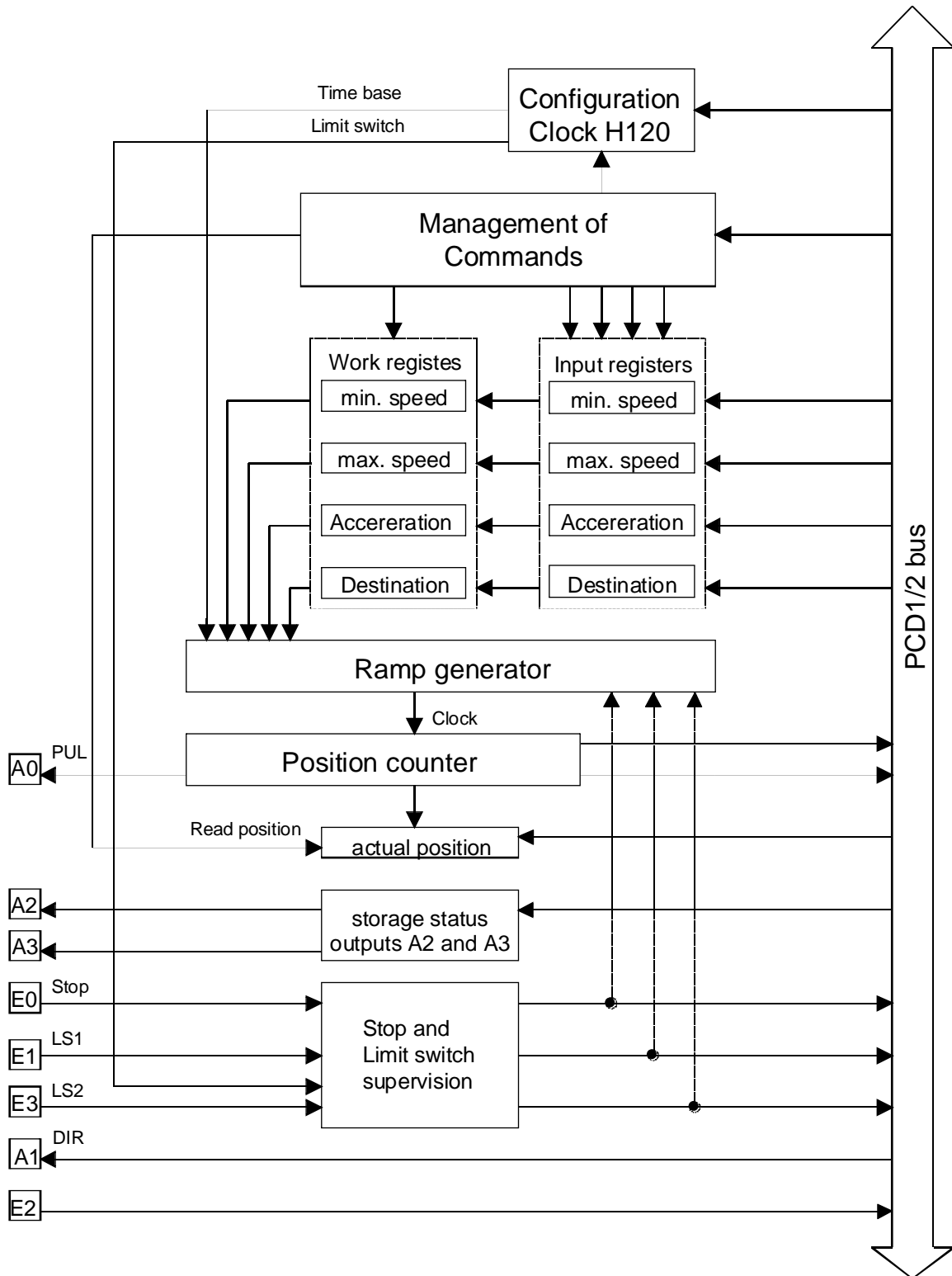
Anzahl 8

LED 0: *) Spannung am Eingang 0: (Notstop)
LED 1: *) Spannung am Eingang 1: (LS1)
LED 2: *) Spannung am Eingang 2: (REF)
LED 3: *) Spannung am Eingang 3: (LS2)
LED 4: Spannung am Ausgang 0: PUL
LED 5: Spannung am Ausgang 1: DIR
LED 6: Spannung am Ausgang 2
LED 7: Spannung am Ausgang 3

*) inverser Status bei Verwendung als Endschalter

5. Funktionsbeschreibung

5.1 Blockschaltbild des Moduls



5.2 Beschreibung des Moduls

Um einen Schrittmotor erfolgreich zu betreiben sind 4 Parameter zu definieren:

- Die Start-/Stop-Frequenz, d.h. die Frequenz mit welcher ein Schrittmotor ohne Verlust von Schritten direkt gestartet und gestoppt werden kann
- Die maximale Frequenz auf welche der Schrittmotor durch Hochfahren unter allen Umständen betrieben werden kann
- Die optimale Beschleunigung, um von der minimalen zur maximalen Frequenz und wieder zurück zu gelangen
- Die Anzahl Schritte, welche für eine Bewegung ausgeführt werden sollen

Die ersten 3 Parameter sind motor- bzw. anlagenspezifisch, d.h. werden einmal definiert und für ein- und dieselbe Anwendung normalerweise nicht mehr verändert. (Parameter 2 und 3 können ev. der Last angepasst werden). Der 4. Parameter, die auszuführende Schrittzahl, ist von der Aufgabe abhängig und muss im Anwenderprogramm laufend angepasst werden.

Die Anzahl Schritte werden im Anwenderprogramm normalerweise vor der Impulsausgabe geladen. In sehr schnellen Applikationen können die Anzahl Schritte für die nächste Bewegung bereits während der vorangehenden Bewegung ins Eingangsregister geladen (EXEC - 'Ld..') und nach Beendigung des Impulszugs sofort ins Arbeitsregister übergeben und die nächste Bewegung gestartet werden (EXEC - 'Start'). Mit dem Befehl (EXEC - 'Start') wird die Impulsausgabe bzw. die Bewegung gestartet. Die Impulse werden am Ausgang 'A0' (PUL) ausgegeben.

Die 3 ersten erwähnten Parameter werden bei der Initialisierung (FB INIT) geladen und können im Bedarfsfall im Laufe des Programmablaufs auch geändert werden.

Bei der einleitenden Konfiguration (FB - INIT) wird der Frequenzbereich definiert. Es kann zwischen 4 Bereichen gewählt werden:

9.5 ... 2431 Hz,	19 ... 4864 Hz,
38 ... 9727 Hz,	76 ... 19'454 Hz

Beim Erreichen der Zielposition wird ein Element (Flag 'ondest_x') gesetzt. Die Abfrage dieses Elements erlaubt die Freigabe des sequentiellen Ablaufs des Anwenderprogramms.

Es besteht die Möglichkeit, eine Unterbrechung eines Impulszuges zu veranlassen (EXEC - 'Stop') und danach die angefangene Fahrt fortzusetzen (EXEC - 'Continue') und gemäss der Vorgabe ohne Schrittverluste zu beenden. Dies muss jedoch im Anwenderprogramm vorgesehen werden.

Die jeweilige Position, d.h. der interne Zählerstand, kann jederzeit gelesen und auf einem Display zur Anzeige gebracht werden.(EXEC - 'RdPosition') Mit 'RdPosition' wird immer die verbleibende Anzahl Schritte gelesen. Die Absolutposition wird bei jedem 'RdPosition' im Register 'rPosAbs_x' aufgefrischt (_x = Modul Nr.).

Es wird zwischen einem relativen und einem absoluten Betrieb unterschieden (EXEC - 'LdDestRel' bzw. EXEC - 'LdDestAbs'). Im "relativen" Betrieb wird zum Anfahren des Ziels immer der Weg (d.h. die auszuführenden Schritte) geladen. Sollen 1000 Schritte vorwärts ausgeführt werden, wird als Destination 1000 geladen. Zur Rückkehr zum Ausgangspunkt wird nun -1000 programmiert. Bei negativen Werten wird jetzt automatisch der Ausgang 'A1' (DIR = Direction) = L gesetzt und es kann damit die Drehrichtung des Motors invertiert werden.

Im "absolutem" Betrieb wird direkt die anzufahrende, absolute Position als Destination geladen. Der FB berechnet anschliessend die relative Bewegung für den SM-Controller. Soll von einer Null-Position z.B. 1000 Schritte hoch und danach wieder zur Ausgangsposition zurückgefahren werden, wird im absoluten Betrieb die Position 1000 als Ziel eingegeben und der Motor gestartet. Um zur Ausgangsposition zurück zu kehren wird die Position 0 geladen. Ist die neue absolute Position kleiner als die bereits erreichte, wird automatisch die Drehrichtung geändert, d.h. der Ausgang 'DIR' wird = L.

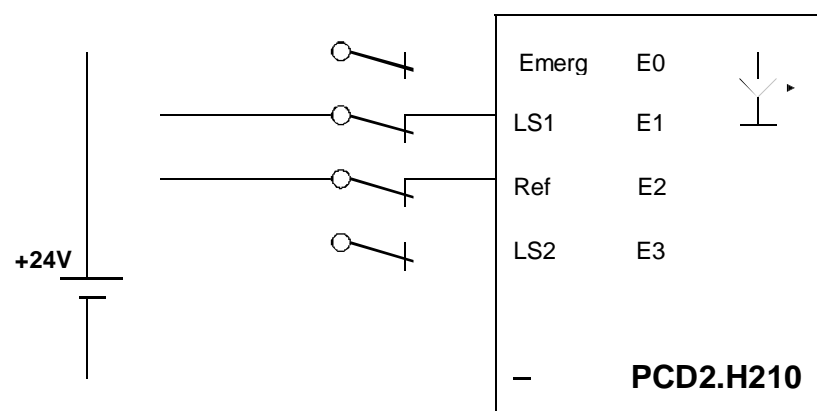
Der Eingang 'E0' kann als Notstop-Eingang konfiguriert werden (INIT), um damit eine laufende Bewegung brüsk (ohne Bremsrampe) zu stoppen. Da die effektive Position aber nicht mehr übereinstimmt, sollten nach einem Notstop die Werte neu definiert und die Bewegung neu initialisiert werden.

Neben dem als Notstop-Eingang (E0) konfigurierbaren Eingang stehen 3 weitere Eingänge (E1 - E3) zur Verfügung. E1 und E3 können als gewöhnliche digitale PCD-Eingänge oder als Endschalter (LS1 und LS2) konfiguriert werden (INIT). Der Eingang E2, wird vom FB 'Home' als Referenzschalter benützt (siehe Abschnitt 5.4). Wird der FB 'Home' nicht verwendet, steht auch E2 als gewöhnlicher PCD Eingang zur Verfügung.

Die Endschalter 'LS1' und 'LS2' sowie der Referenzschalter 'Ref' und der Notstop sind, wenn als Endschalter und als Notstop konfiguriert im unbetätigten Zustand (inaktiv) geschlossen und bringen +24V an die Eingänge. Es ist zu beachten, dass geglättete Gleichspannung verwendet werden muss (siehe technische Daten), da die Eingangsschaltungen so ausgelegt sind, dass ein Anhalten unverzögert, d.h. schrittgenau erfolgt (Zeitkonstante des Eingangsfilters < 1 ms). Sind diese Eingänge nicht als End- bzw. Notstop konfiguriert, ist deren Verhalten gleich wie dasjenige "normaler" PCD-Eingänge.

Die LED zeigen immer die angelegte Spannung am jeweiligen Eingang an.

Anschlussschema der Eingänge wenn als Notstop und als Endschalter konfiguriert:



Wenn 'LS1' oder 'LS2' aktiv werden (Spannung = 0, LED gelöscht), wird direkt eine Bremsrampe ausgelöst, sofern die Schalter als Endschalter konfiguriert sind. Die Position geht dabei nicht verloren.

Mit dem Eingangssignal 'LSxTrig_x' wird angezeigt, dass eine Bewegung durch einen Limit-Switch (LS) gestoppt wurde. Das Signal ist speichernd gesetzt, d.h. es bleibt auf 'H' auch wenn der LS bereits wieder inaktiv ist. Beim Starten einer neuen Bewegung wird 'LSxTrig_x' wieder 'L' (sofern kein LS aktiv ist).

Das Ansprechen eines LS ist ein abnormaler Zustand. Es sollte beim Erreichen dieses Zustandes ein Alarm ausgelöst werden. Auch sollte das Servicepersonal vor Ort das Geschehen analysieren. Danach ist die Achse manuell wieder in den Arbeitsbereich (zwischen den LS1 und den LS2) zurück zu bringen.

Es sind auch zwei Möglichkeiten vorgesehen, um automatisch wieder in den Normalbetrieb zurück zu kehren. Einerseits ermöglicht der FB 'Home' ein weiteres Betreiben der Achse, wobei die korrekte Suchrichtung im Anwenderprogramm ermittelt werden muss. Die zweite Möglichkeit besteht darin, dass der FB 'Init' die LS ausschaltet (Parameter 4) um dann mit einer Exec - 'Start' Bewegung zurück in den Arbeitsbereich zu fahren. Es darf danach nicht vergessen werden, die LS wieder aktiv zu schalten.

Wenn der Notstop (Emerg) aktiv wird, wird die Bewegung ohne Bremsrampe abgebrochen. Mit dem Eingangssignal 'EmergTrig_x' wird dies speichernd angezeigt. Wird nach einem Notstop die Position gelesen (RdPosition), zeigt diese an wo der Notstop ausgelöst wurde. Dies ist nicht unbedingt die Position an welcher der Motor zum Stillstand gekommen ist.

Der Eingang 'Ref'. hat keine direkte Auswirkung auf die Bewegung und wird nur vom FB 'HOME' zum Anfahren der Referenzposition verwendet.

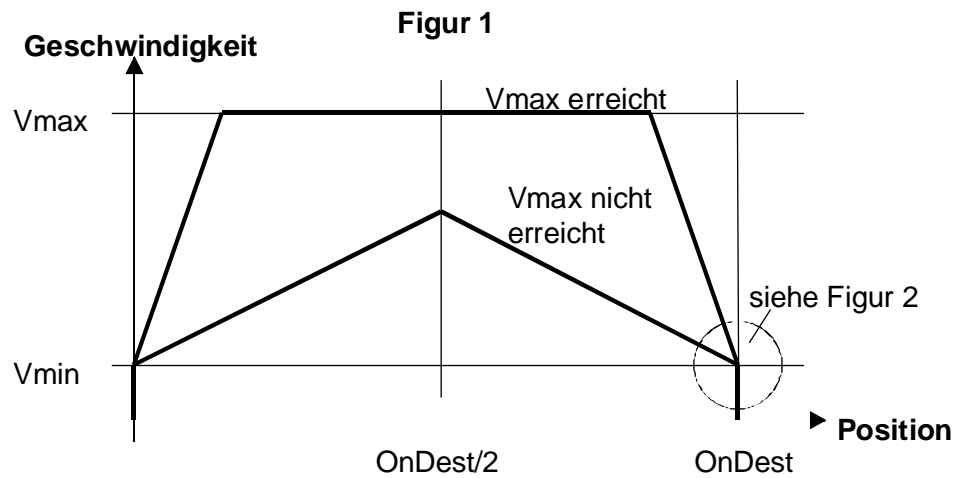
Das Abfragen der Eingänge erfolgt immer, d.h. unabhängig von der Konfiguration, auf die symbolischen Namen

Emerg_x	für Eingang E 0
LS1_x	für Eingang E 1
REF_x	für Eingang E 2
LS2_x	für Eingang E 3

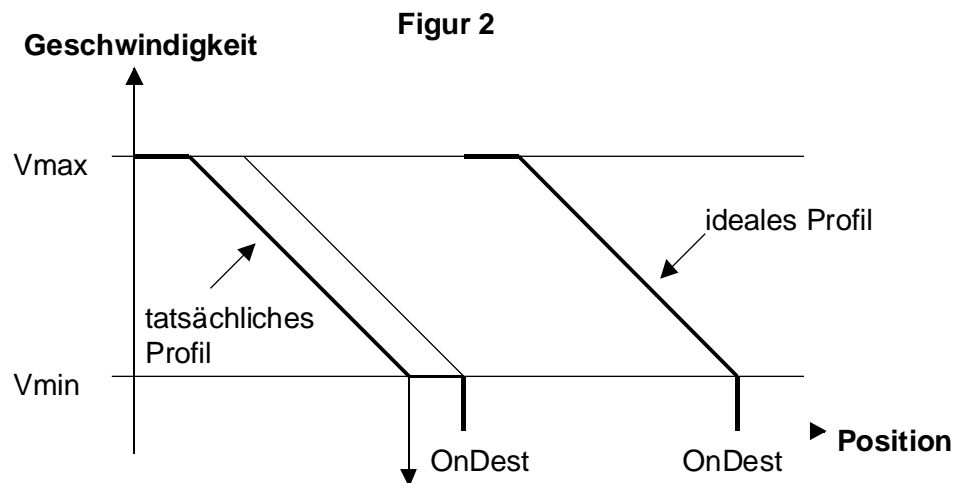
wobei für 'x' die Nummer des Moduls einzusetzen ist.

Die 2 nicht spezifisch verwendeten Ausgänge 'A2' und 'A3' können über das Anwenderprogramm mittels FBs angesteuert und frei verwendet werden (EXEC - SetOut2, EXEC - ResOut2, EXEC - SetOut3, EXEC - ResOut3). Ein Zurücklesen dieser Ausgänge ist nicht möglich.

5.3 Ergänzende Information: Das Frequenz-Profil



Anzahl Schritte für Beschleunigung = Anzahl Schritte für Abbremsung



Ergänzung der Bewegung mit einigen Schritten mit Vmin

Die nachfolgend gezeigte Gleichung erlaubt die Bestimmung der maximalen Zeit zur Ergänzung der Bewegung.

$$\text{max. Zeit} = \frac{V_{\text{max}}}{V_{\text{min}}} \times T_{\text{acc}}$$

wobei:

- Vmax** = maximale erreichte Geschwindigkeit während der Bewegung
- Vmin** = minimale programmierte Geschwindigkeit (Start-Stop Frequenz)
- Tacc** = Beschleunigungszeit: $n * 62.5 \mu\text{s}$ ($1 \leq n \leq 255$)

Zur Beachtung: Die gezeigte Formel entspricht dem schlechtesten Fall (Worst Case). Durch kleine Veränderungen der verschiedenen Parameter Vmax, Vmin und Tacc kann die Bewegung bzw. die Ergänzungszeit optimiert werden.

5.4 Ergänzende Information: Referenzfahren (FB Home)

Mit dem FB 'Home' kann die Referenzfahrt eigenständig ausgeführt werden. Zur Definition der Referenzfahrt werden 9 Parameter benötigt. Der Aufruf des FB 'Home' ist im Abschnitt 7.1.2. beschrieben.

Die zu referenzierende Achse muss initialisiert sein (FB Init).
(Die nachfolgende Beschreibung bezieht sich auf Bild 6.07).

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 den in den Parametern 6 und 7 definierten Geschwindigkeiten sowie der in Parameter 8 definierten Beschleunigung durchgeführt. Im Parameter 3 wird die Suchrichtung definiert. Wird der Referenzschalter nicht gefunden und trifft die Achse auf einen Endschalter (HW oder SW), wird die Suchrichtung invertiert.
2. Ist der Referenzschalter gefunden, beginnt das Freifahren. Die Richtung des Freifahrens ist in Parameter 4 definiert, die Geschwindigkeit für das Freifahren ist V_{min} (Parameter 6).
3. Nachdem der Referenzschalter abgefallen ist, stoppt die Achse und als Absolutposition wird die in Parameter 2 definierte Position geladen.
4. Das Modul wird mit den ursprünglichen Einstellungen (aus dem FB Init) konfiguriert und der FB Home wird verlassen.

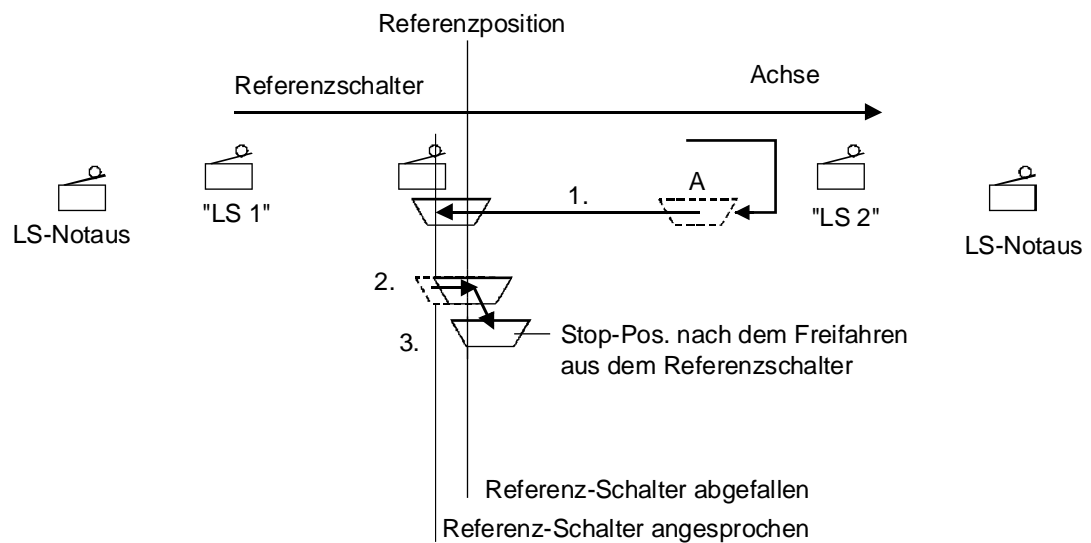
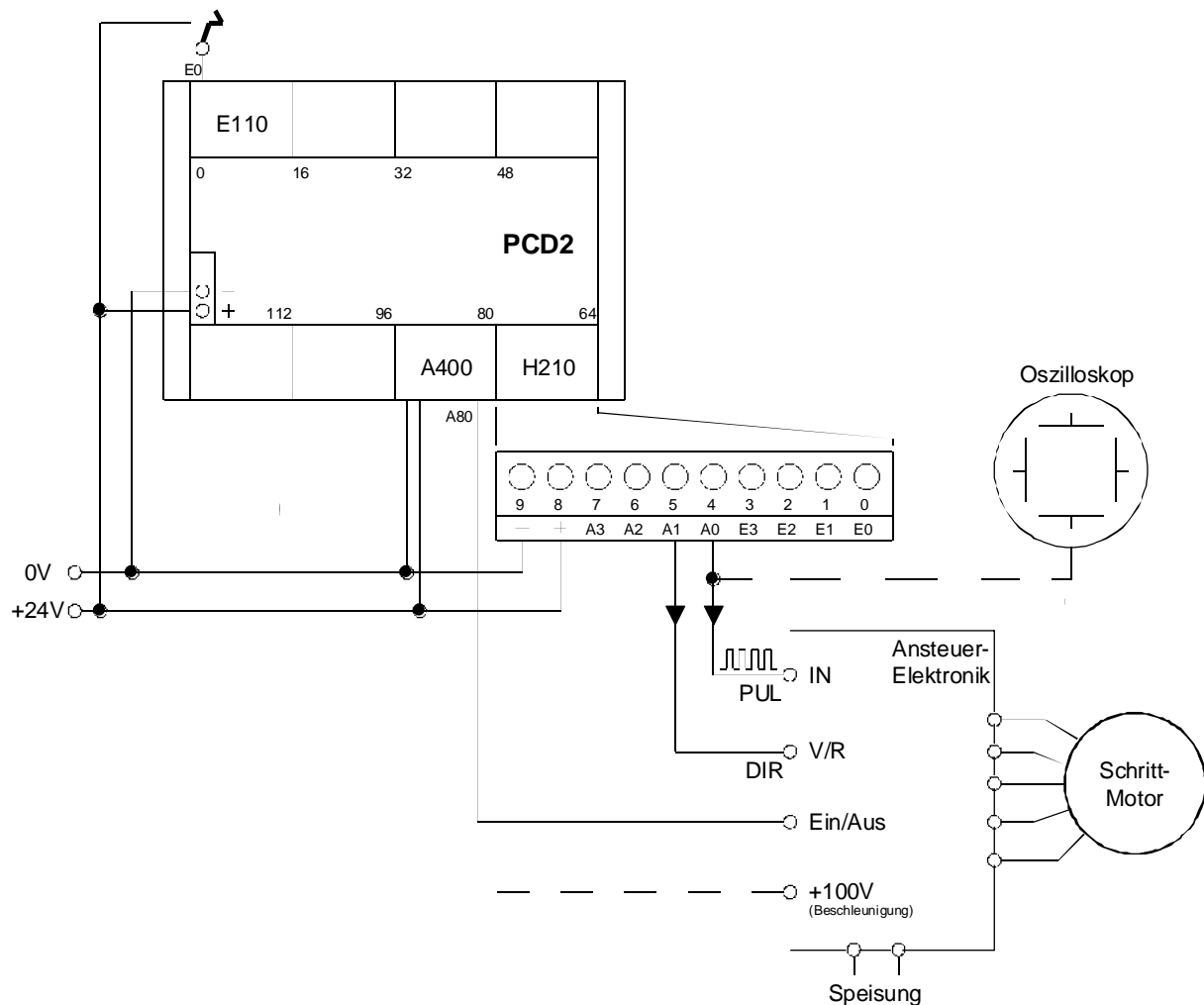


Bild 6.07

Hinweise:

- Es können mehrere Module gleichzeitig auf der gleichen PCD referenziert werden.
- Als Schalter müssen Öffner-Kontakte verwendet werden.
- Während der Referenzfahrt sind die Limit-Switches intern ausgeschaltet. Diese werden nur als Richtungsumschalter benützt.
- Der Notstop ist nur aktiv, fallsdieser im FB 'Init' konfiguriert wurde.
- Wird kein Referenzschalter gefunden (was nach zweimaliger Richtungsumkehr bei einem LS der Fall ist), wird das Fehlerflag 'fHomeErr_x' (x = Modul Nr.) gesetzt, die Absolutposition auf 0 gebracht und der FB 'Home' verlassen.
- Das Flag 'fEndHome_x' muss vor einer Referenzfahrt im Anwenderprogramm zurückgesetzt werden. Dieses Flag wird am Ende der Referenzfahrt automatisch 'H' gesetzt.
- Der FB 'Home' wird verlassen, wenn die Referenzfahrt erfolgreich ausgeführt oder ein Fehler erkannt wurde, kann der FB mit einem Timeout (Parameter 9) abgebrochen werden. Der Wert entspricht der Zeit in Sekunden nach welcher der FB 'Home' abgebrochen wird. Zusätzlich zum Fehlerflag 'fHomeErr_x' wird in diesem Fall das Diagnoseregister 'rDiag' mit dem Code 9 (für Parameter 9) im 3. Byte (für FB Home) geladen. Der Parameter Check erfolgt wie in Kapitel 8 beschrieben.

6. Der schnelle Einstieg



Minimal-Anordnung für den Betrieb eines Schrittmotors, welcher mit dem PCD2.H210 gesteuert werden soll.

Die einzelnen Elemente sind:

- PCD1 oder PCD2 mindestens bestückt mit

1 PCD2.H210
1 PCD2.E110
1 PCD2.A400
- Geeignete Ansteuer-Elektronik
- Schrittmotor, ev. mit Schlitten
- Speisegerät
- Oszilloskop (nicht unbedingt benötigt)

Die Parameter für 'Vmin', 'Vmax' und 'Acc' (Beschleunigung) sind entsprechend dem eingesetzten Antrieb individuell zu wählen und können **nicht** von den nachfolgend gezeigten Beispielen übernommen werden.

6.1 Einstieg mit Programmierung in IL (PG4)

Um auf einfachste Weise einen Schrittmotor in Betrieb zu nehmen, sei das nachfolgend gezeigte Minimal-Programm vorgeschlagen:



Ein sauberes Anwenderprogramm soll keine Warteschleifen enthalten. Das vorliegende Beispiel wurde, zum Aufzeigen der wesentlichen Befehle zur Ansteuerung eines PCD2.H210, 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 Kapitel 9).

Der Schrittmotor habe die folgenden Daten:

Start-Stop Frequenz:	40 Hz
Maximal Frequenz:	1000 Hz
Beschleunigung:	2000 Hz/sek
Schritte/Umdrehung:	48

Aufgabe: Es soll nach dem Einschalten des PCD-Eingangs 'E 0' ein Schrittmotor jeweils 4800 Schritte ausführen.

Die einzelnen Parameter und die Einstellung der Basisadressen sind dem nachfolgenden Kapitel 7.1 zu entnehmen. Das Anwenderprogramm kann dann z.B. wie folgt gestaltet werden:

(Ein ähnliches Beispiel, jedoch mit Vor- und Rückwärts-Bewegung und unter Verwendung der absoluten und der relativen Positionsprogrammierung, ist auf der FB-Diskette als "H210_EX1.SRC" zu finden).

Die FBs (IL für PG4) sind auf der Diskette PCD9.H21E. Um die FBs auf den PC zu installieren, ist den Anweisungen in der Datei README.TXT zu folgen. Diese Datei liegt auch auf dieser Diskette.

Die Anzahl Module (1) und die Adresse des PCD2.H210-Modules (Adresse 64) muss in die Datei 2D2H210_B.MBA eingetragen werden:

NbrModules EQU 1 ; No. of H210 modules used (0...16)

BA_1 EQU 64 ; Base address of module 1

Diese Datei (2D2H210_B.MBA) muss im Projektverzeichnis dieses Beispiels liegen, d.h. diese Datei ist ab der Diskette von Hand ins aktuelle Projektverzeichnis zu kopieren.


```

#include d2h210_b.equ
$group H210

    xob          16

    cfb          init          ; Initialisierung
                k 1          ; Modul-Nummer
                0           ; Frequenzbereich 0
                0           ; Not-Stop: nein
                0           ; End-Schalter: nein
                4           ; Vmin (38 Hz)
                105         ; Vmax (954 Hz)
                77          ; Beschleunigung (1986 Hz/s)

    ld           r 1          ; PCD-Register
                4800         ; Anzahl Schritte

    exob

; -----

    cob          0
                0

st1:   sth      i 0          ; Start ?
        jr      l st1

    cfb          exec         ; Lade destination relative
                k 1          ; Modul-Nummer
                LdDestRel    ; Befehl: load dest. rel.
                r 1          ; Reg. mit Anzahl Schritten

    cfb          exec         ; Start
                k 1          ; Modul-Nummer
                start        ; Befehl: Start
                rNotused     ; leeres Register

wait:   sth      OnDest_1    ; finished ?
        jr      l wait

st2:   sth      i 0          ; start ?
        jr      h st2

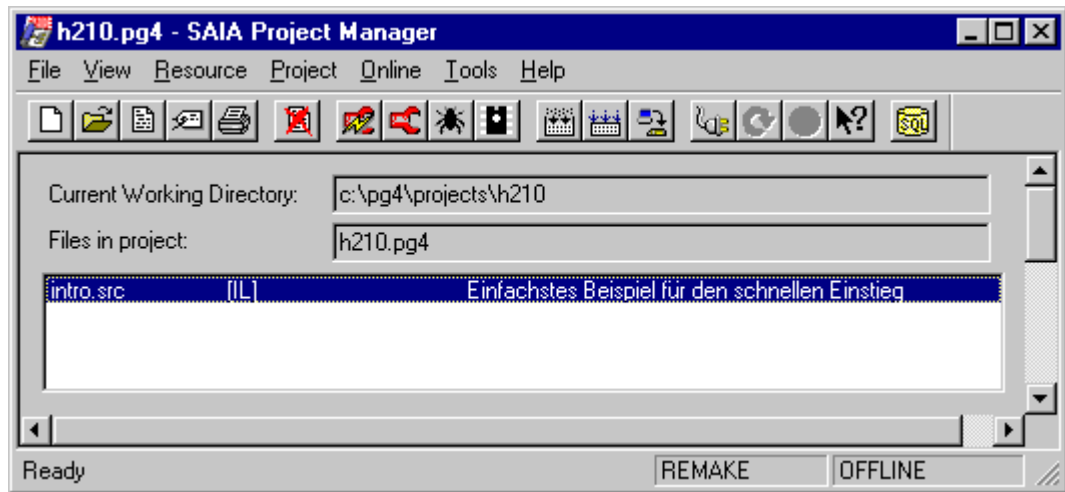
    ecob

$endgroup
; -----

```

Wie bereits erwähnt soll ein sauberes Anwenderprogramm keine Warteschleifen enthalten. Das vorliegende Beispiel wurde, zum Aufzeigen der wesentlichen Befehle zur Ansteuerung eines PCD2.H210, trotzdem mit Warteschleifen gestaltet. In der Praxis ist für diese Art von Programmen immer eine GRAFTEC-Struktur zu wählen, siehe Kapitel 9.

Das Programm hat (beispielsweise) den Namen "intro.src" und liegt im Projekt "H210" im PG4.



Mit 'Project' - 'Build' wird das Programm verarbeitet und in die PCD geladen.

Soll die Position dauernd, z.B. an einem Displaymodul PCD7.F5x0 angezeigt werden, wird am einfachsten ein neuer COB zum bereits bestehenden COB 0 angehängt und folgendes editiert:

```

        cob      0
            0

        ecob
; -----

$group H210

        cob      1
            0

        cfb      exec      ; Lese Position
                k 1        ; Modul-Nummer
                RdPosition ; Befehl: RdPosition
                r 99       ; Display-Register

        dsp      r 99      ; Anzeige Display-Register

        ecob

$endgroup
; -----

```

Es müssen für diesen Fall im COB 0 in allen Warteschleifen je ein Befehl 'Next COB' (NCOB) eingebaut werden, da sonst das Programm den COB 1 nur sehr selten erreicht. Hierbei wird die Problematik der Spaghetti-Programmierung wieder einmal offensichtlich...

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". (Für die Verwendung des älteren Programmierwerkzeuges "PG3" sind eigene FBs vorhanden).

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.H21E 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.H21E. Die Diskette enthält die folgenden Verzeichnisse:

- APPSDIR : enthält alle Helps
- FB : enthält die .SRC- und .EQU-Dateien des H210
- FBOX : enthält die FBoxen zum H210
- 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.H21E 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:

D2H210_B.SRC	FB-Quell-Code	read-only Datei
D2H210_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
D2H210_B.HLP	FB-Help-Datei

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

Die Datei **D2H210_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 '**D2H210_B.MBA**' ist nachfolgend gezeigt:

Datei: **D2H210_B.MBA** (MBA = Modul Basis Adresse)

```

;
; This file can be modified by the user
;
; Basis addresses defined by the user
; -----
$group H210
NbrModules      EQU      1          ; No. of H210 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.H210-Module anzugeben. danach sind die Hardware-Basisadressen der verwendeten PCD2.H210-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. H210-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 H210 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      208        ;Base address of module 2
BA_3            EQU      112        ;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-H2" und das eigentliche Anwenderprogramm den Namen "move-01.sfc".

```
C:\pg4 \FB          \D2H210_b.equ
          \D2H210_b.src
          \...
          \FBOX          \...
          \GALEP3        \...
          \PROJECTS     \FUP_E          (Demo Beispiel PG4)
                              \GRAF_E         (Demo Beispiel PG4)
                              \TEST-H2       \D2H210_b.mba
                                          \move-01.sfc
          \...
          \D2H210_b.hlp
```

Das Anwenderprogramm für den H210-Teil präsentiert sich folgendermassen :

```
$include D2H210_b.equ
$group H210

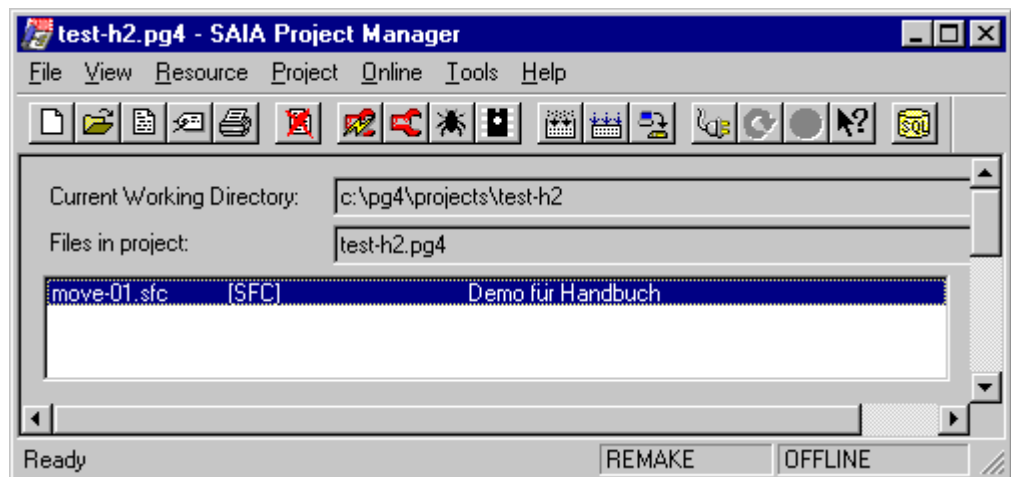
XOB      16

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 7 Parametern
- EXEC Ausführen FB mit 3 Parametern
- HOME Grundstellung FB mit 9 Parametern

Der Aufruf des FB "INIT" präsentiert sich immer wie folgt:

```
CFB      init
k        1      ; Par. 1: Modul-Nummer (k 1 - k 16)
          1      ; Par. 2: Frequenzbereich (0 - 3)
          0      ; Par. 3: Notstop (0=Aus/1=Ein)
          0      ; Par. 4: Endschalter (0=Aus/1=Ein)
          1      ; Par. 5: Vmin (1 - 255)
          100    ; Par. 6: Vmax (1 - 255)
          20     ; Par. 7: Beschleunigung (1 - 255)
```

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
r        777    ; Par. 3: Wert (aus Quell-Register)
```

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

```
CFB      exec
k        1      ; Par. 1: Modul-Nummer (k 1 - k 16)
          RdPosition ; Par. 2: Funktion
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.

Befehle (Funktionen) für den FB 'Exec':

```
LdVmin      ; load min. frequency
LdVmax      ; load max. frequency
LdAcc       ; load acceleration

LdDestRel   ; load Destination relative

LdDestAbs   ; load Destination absolute

Start       ; initialise and start motion
Stop        ; stop motion
Continue     ; continue motion after stop
```

```

SetOut2          ; set      Output 2
ResOut2          ; reset   Output 2
SetOut3          ; set      Output 3
ResOut3          ; reset   Output 3

RdPosition      ; read remaining Position
RdIdent         ; read module identification
    
```

Ein FB 'SetZero', zur Definition irgendeiner Position als Null, existiert nicht. Es kann für diesen Fall das Register 'rPosAbs_x' verwendet werden, indem die Null-Position als

```

Ld      rPosAbs_x
        0
    
```

geladen wird. Achtung: Die Achse muss zur Durchführung dieses Befehls stillstehen!

Der Aufruf des FB 'Home' präsentiert sich immer wie folgt:

CFB	HOME		
k	1	; Par. 1:	Modul-Nummer (k 1 - k 16)
r	888	; Par. 2:	Ref. Position (R)
	1	; Par. 3:	Such Richtung (0 = ab/1 = auf)
	0	; Par. 4:	Ref. Schalter Verlassrichtung (0 = ab/1 = auf)
	2	; Par. 5:	Frequenzbereich (0 - 3)
	10	; Par. 6:	Vmin (1 - 255)
	100	; Par. 7:	Vmax (1 - 255)
	20	; Par. 8:	Beschleunigung (1 - 255)
	30	; Par. 9:	Timeout in Sek. (1 - 65535)
			0 = kein Timeout)

Generelle Definitionen:

```

rDiag          ; error flag if parameter value wrong
fPar_Err
    
```

Modulspezifische Definitionen:

```

fHomeErr_x      ; Error during home procedure
fEndHome_x      ; Low during homing

rPosAbs_x       ; absolute position (wird durch
                ;   FB Exec RdPosition aufgefrischt)

EmergTrig_x     ; Emergency stop Triggered
OnDest_x        ; Destination OK
LSxTrig_x       ; Limit Switch 1 or 2 Triggered
Emerg_x         ; Emergency stop
LS1_x           ; Limit Switch 1
REF_x           ; REFerence switch
LS2_x           ; Limit Switch 2
    
```

"_x" entspricht der Modul-Nummer

Die effektiven Adressen der Register und Flag sind in der Datei 'Projektname.MAP' aufgelistet (für Debug Zwecke).

7.1.3 Bestimmung der Parameter

Die Werte für 'Vmin', 'Vmax' und 'Beschleunigung' können den nachfolgenden Tabellen entnommen werden. Es können für diese Parameter je Werte von 1 bis 255 gewählt werden. Es stehen 4 Frequenzbereiche 0 bis 3 zur Verfügung. Die Werte für die Parameterierung einer Aufgabe sind immer ein- und demselben Bereich zu entnehmen.

Werte für "Vmin" und "Vmax" (Vmin und Vmax in Hz)

Wert	Bereich 0	Bereich 1	Bereich 2	Bereich 3
1	9.5	19	38	76
2	19	38	75	152
3	28	57	114	229
4	38	76	153	305
5	48	95	191	381
6	57	114	229	458
8	76	153	305	610
10	95	191	381	763
15	143	286	572	1144
20	190	381	763	1526
25	238	477	954	1907
30	286	572	1144	2289
40	381	763	1526	3052
50	477	964	1907	3814
60	572	1144	2289	4577
70	667	1335	2670	5340
80	763	1526	3052	6103
90	858	1717	3433	3866
100	954	1907	3815	7629
110	1049	2098	4196	8392
120	1144	2289	4577	9155
130	1240	2480	4959	9918
140	1335	2670	5340	10680
150	1430	2861	5722	11443
160	1526	3052	6103	12206
170	1621	3243	6485	12969
180	1716	3433	6866	13732
190	1812	3624	7248	14495
200	1907	3815	7629	15258
210	2002	4006	8010	16021
220	2098	4196	8392	16784
230	2193	4387	8773	17546
240	2288	4578	9155	18309
250	2384	4769	9536	19072
255	2431	4864	9727	19454

Beispiel: siehe nächste Seite

Werte für die Beschleunigung (in Hz/sek)

Wert	Bereich 0	Bereich 1	Bereich 2	Bereich 3
1	152939	305911	611821	1223642
2	76470	152955	305911	611821
3	50980	101970	203940	407881
4	38235	76478	152955	305911
5	30588	61182	122364	244728
6	25489	50985	101970	203940
8	19117	38239	76478	152955
10	15294	30991	61162	122364
15	10196	20394	40788	81776
20	7647	15296	30591	61162
25	6118	12236	24473	48946
30	5098	10197	20394	40788
40	3823	7648	15296	30591
50	3059	6118	12236	24473
60	2549	5099	10197	20394
70	2185	4370	8740	17481
80	1912	3824	7648	15296
90	1699	3399	6798	13596
100	1529	3059	6118	12236
110	1390	2781	5562	1124
120	1274	2549	5099	10197
130	1176	2353	4706	9413
140	1092	2185	4370	8740
150	1020	2039	4079	8158
160	956	1920	3824	7648
170	900	1800	3599	7198
180	850	1700	3399	6798
190	805	1610	3220	6440
200	765	1530	3059	6118
210	728	1457	2913	5826
220	695	1391	2781	5562
230	665	1330	2660	5320
240	637	1275	2549	5099
250	612	1224	2447	4895
255	600	1200	2400	4800

Beispiel: Es sind die Werte für $V_{min} = 100$ Hz, $V_{max} = 3000$ Hz und eine Beschleunigung von 6000 Hz/sek zu bestimmen.

Es wird immer der Bereich mit den tiefsten noch erreichbaren Frequenzen gewählt, für unsern Fall "Bereich 1".

für V_{min} → Wert 5 (95 Hz)
 für V_{max} → Wert 150 - 160 = ca. 157
 für Beschleunigung → Wert ca. 50

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 D2H210_b.MBA werden während dem Assemblieren überprüft:

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

"Remark: No H210 used (NbrModules = 0 in D2H210_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 H210 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 'H210.RdIdenti'"

(wobei der Ausdruck 'H210' vom \$group h210 generiert wird)

- Falls die Definition \$group H210 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 (z.B. Frequenz Bereich = 0, 1, 2, 3). Falls ein Parameter ausserhalb eines Bereichs liegt, wird dieser auf den minimalen Wert (Ausnahme: auf 255 für die Beschleunigung), 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 rsp. 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 der Frequenzbereich (Parameter 5) im FB Home von Module 2 falsch definiert (>3), so wird 'rDiag' mit dem Wert 00 03 05 02 hex geladen.

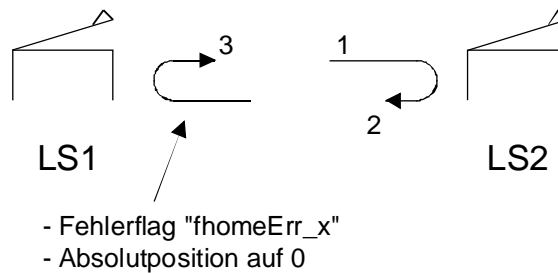
Das Diagnose-Register wird bei jedem falschen Parameter überschrieben und enthält immer den letzten Fehler. Es sollte deshalb ausgewertet werden sobald das Flag 'fPar_Err' einen Bereichsfehler signalisiert. Die absoluten Adressen von 'rDiag' und 'fPar_Err' sind in der Datei 'project.MAP' ersichtlich. 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 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 9 als Parameter-Nr. geladen (Timeout ist Parameter 9).

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
      .
      .
      .

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
...
  
```

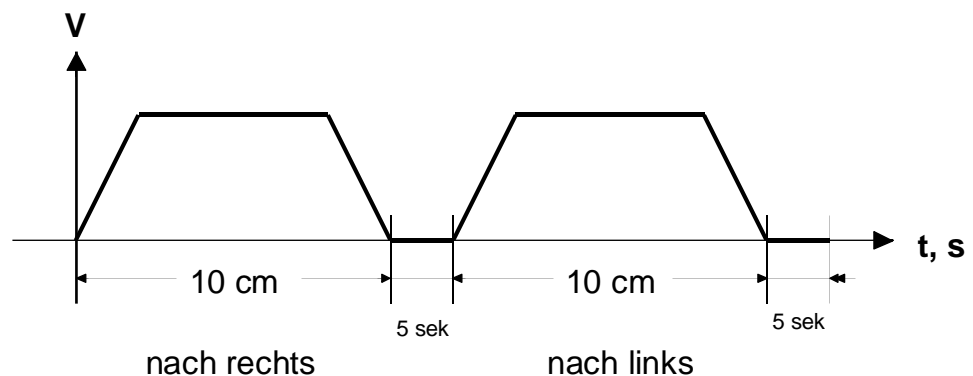
Notizen

9. Anwendungsbeispiele

9.1 Typisches Anwendungsbeispiel in GRAFTEC und IL

Basierend auf dem Beispiel in Kapitel 6, "Der schnelle Einstieg", soll ein Schlitten, welcher von einem Schrittmotor angetrieben wird nach dem Einschalten des PCD-Eingangs 0 (Start) aus einer Startposition um 10 cm nach rechts und nach einer Pause von 5 sek wieder in die Ausgangsstellung zurück fahren.

Bleibt der Eingang "Start" eingeschaltet, soll das Spiel nach weiteren 5 sek Pause wieder von Neuem beginnen.



Als Variante soll mit dem PCD-Eingang 1 ("Stop") der Lauf des Schlittens unterbrochen und mit dem PCD-Eingang 2 ("Weiter") weiter gelaufen lassen werden, wobei die Anfahr- und Bremsrampen beibehalten werden sollen.

Einige Daten zum Laufmodell:

Schritte/Umdrehung:	48
Steigung der Spindel:	1 mm/U
Vmin (Start-Stop):	40 Hz
Vmax:	1000 Hz
Beschleunigung:	2000 Hz/sek

Das Projekt erhält den Namen "TEST-H2"

Das eigentliche Anwenderprogramm wird in GRAFTEC geschrieben und erhält den Namen "H210_Ex2.SFC".

Die Parameter für 'Vmin', 'Vmax' und 'Acc' (Beschleunigung) sind entsprechend dem eingesetzten Antrieb individuell zu wählen und können **nicht** von den nachfolgend gezeigten Beispielen übernommen werden.

Es sind die folgenden Schritte auszuführen:

- Die Datei "D2H210_b.mba" ins Projektverzeichnis "test-h2" kopieren und die Basisadresse des PCD2.H210-Moduls definieren, z.B. 64.
- Bestimmen der Parameter für Geschwindigkeiten und Beschleunigung:

Gemäss Abschnitt 7.1.3 wird Bereich 1 gewählt.

Vmin → Wert 4
 Vmax → Wert ca. 105
 Beschleunigung → Wert ca. 75

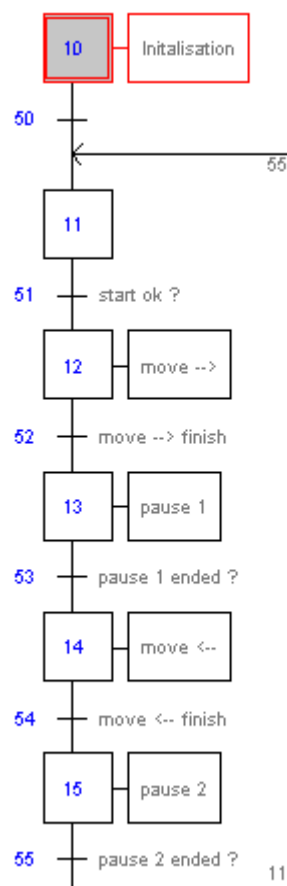
- Bestimmen der Anzahl Schritte

Anzahl Schritte: 100 mm x 48 Schritte = 4800 Schritte

Wird im Modus "relativ" gearbeitet wird für die 1. Bewegung (nach rechts) 4800 und für das Zurückfahren -4800 eingegeben.

Wird im Modus "absolut" gearbeitet wird für die 1. Bewegung ebenfalls 4800 eingegeben, für das Zurückfahren wird jetzt aber 0 (Null) programmiert.

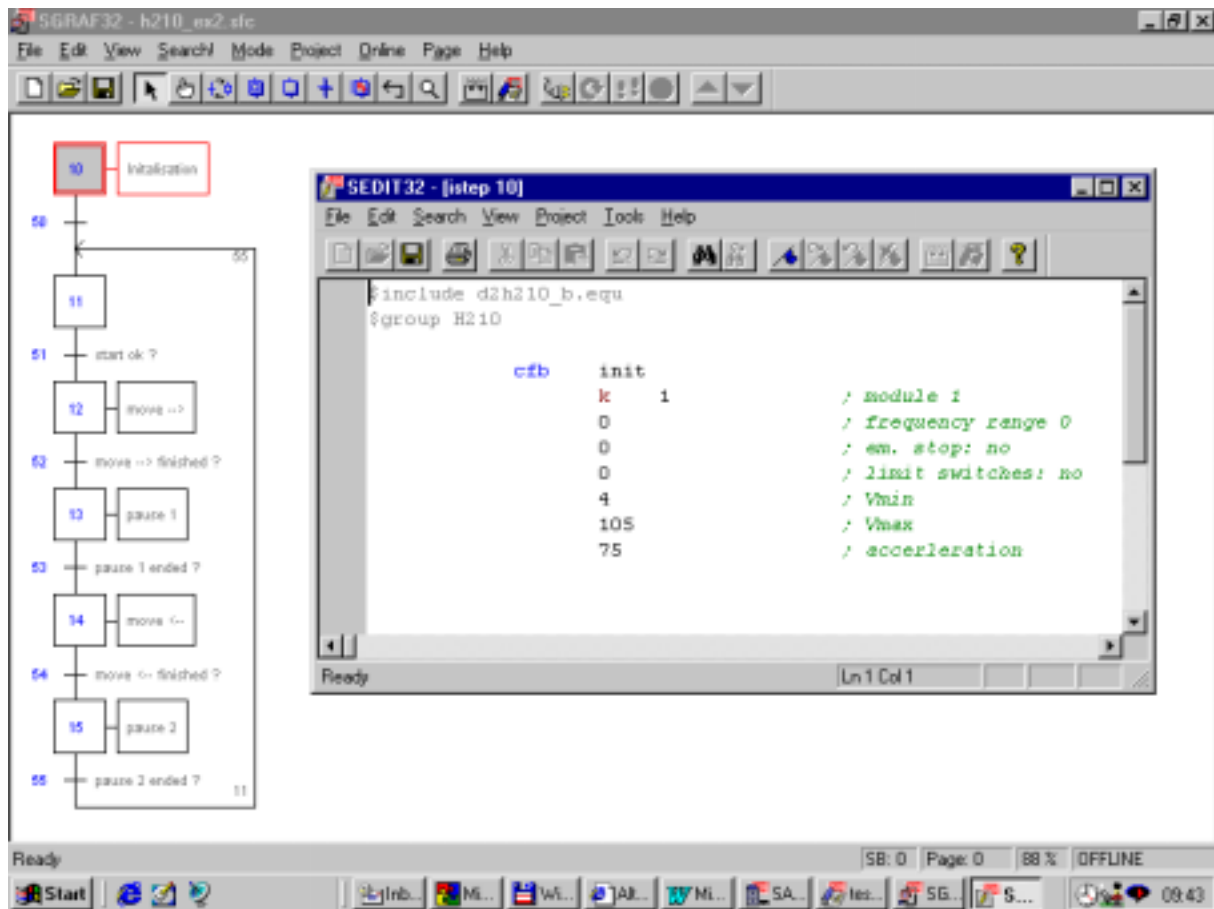
- Erstellen der GRAFTEC-Struktur mit Kommentaren



Codieren der einzelnen Steps und Transitionen mit SEDITWIN



(Das Editieren mit SFUP wird im Abschnitt 9.2 behandelt).



Der Code der einzelnen Steps und Transitionen wird in gleicher Weise editiert.

Die Datei "H210_Ex2.SFC" kann z.B. im Notepad-Editor aufgerufen und ausgedruckt werden. Siehe die folgenden Seiten. Zusammen mit dem Ausdruck der GRAFTEC-Struktur bildet der Ausdruck der Code-Datei die Dokumentation des Programms.

(Um die nachfolgend gezeigte Darstellung zu erhalten, ist die Datei "h210_ex2.sfc" in "h210_ex2.src" umzubenennen).

```

SB      0
;-----
IST     10          ;Initalisation
        O 50
$include d2h210_b.equ
$group H210

        cfb        init
        k          1      ; module 1
        0          0      ; frequency range 0
        0          0      ; em. stop: no
        0          0      ; limit switches: no
        4          4      ; Vmin
        105        105    ; Vmax
        75          75    ; accerleration

EST     ;10
;-----
ST      11
        I 50
        I 55          ;pause 2 ended ?
        O 51          ;start ok ?
EST     ;11
;-----
ST      12          ;move -->
        I 51          ;start ok ?
        O 52          ;move --> finished ?
ld      r          1
        4800

        cfb        exec
        k          1      ; module 1
        lddestrel
        r          1

        cfb        exec
        k          1      ; module 1
        start
        rnotused

EST     ;12
;-----
ST      13          ;pause 1
        I 52          ;move --> finished ?
        O 53          ;pause 1 ended ?
ld      t          1
        50          ; 5 sec
EST     ;13
;-----
ST      14          ;move <--
        I 53          ;pause 1 ended ?
        O 54          ;move <-- finished ?
ld      r          1
        -4800

        cfb        exec
        k          1      ; module 1
        lddestrel
        r          1

        cfb        exec
        k          1      ; module 1
        start
        rnotused

EST     ;14

```

```

;-----
ST    15          ;pause 2
      I 54          ;move <-- finished ?
      O 55          ;pause 2 ended ?
ld    t          1
      50          ; 5 sec
EST   ;15
;-----
TR    50
      I 10          ;Initalisation
      O 11
ETR   ;50
;-----
TR    51          ;start ok ?
      I 11
      O 12          ;move -->
sth   i          0
ETR   ;51
;-----
TR    52          ;move --> finished ?
      I 12          ;move -->
      O 13          ;pause 1
cfb   exec
      k 1          ; module 1
      rdposition
      r 0
dsp   r          0    ; display module
sth   ondest_1
ETR   ;52
;-----
TR    53          ;pause 1 ended ?
      I 13          ;pause 1
      O 14          ;move <--
cfb   exec
      k 1          ; module 1
      rdposition
      r 0
dsp   r          0    ; display module
stl   t          1
ETR   ;53
;-----
TR    54          ;move <-- finished ?
      I 14          ;move <--
      O 15          ;pause 2
cfb   exec
      k 1          ; module 1
      rdposition
      r 0
dsp   r          0    ; display module
sth   ondest_1
ETR   ;54

```

```

;-----
TR    55          ;pause 2 ended ?
      I 15          ;pause 2
      O 11

cfb   exec
      k    1      ; module 1
      rdposition
      r    0

dsp   r    0      ; display module

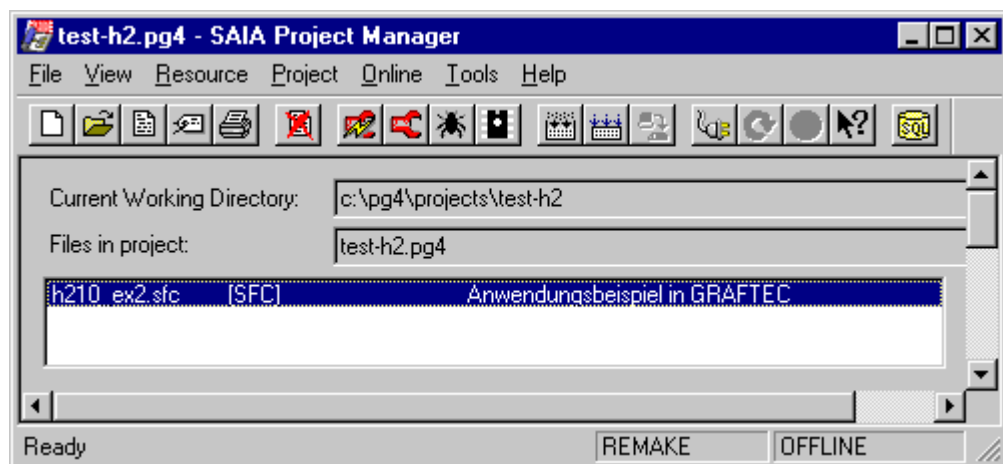
stl   t    1

$endgroup
ETR   ;55

ESB   ;0

```

Ist alles soweit gedingen, kann aus dem Project-Manager mit den Funktionen 'Project' -'Build' das Programm verarbeitet und in die PCD geladen werden.



Ist ein Displaymodul PCD7.F510 oder F520 bestückt, kann neben der Bewegung der Achse auch der Zählerstand direkt verfolgt werden.

Andere, ähnliche Beispiele sind der Diskette PCD9.H21E zu entnehmen.

Variante mit Stop- und Weiterlaufmöglichkeit: (H210_Ex3.SFC)

Soll, wie in der Aufgabenstellung angedeutet, der Lauf der Bewegung mit dem PCD-Eingang 1 gestoppt und mit dem PCD-Eingang 2 wieder weiter laufen gelassen werden, sind die Transitionen 52 (move →) und 54 (move ←) wie folgt zu programmieren:

```

;-----
TR    52                                ;move --> finished ?
      I 12                                ;move -->
      O 13                                ;pause 1

cfb   k      exec
      k      1                            ; module 1
      r      rdposition
      r      0

dsp   r      0                            ; display module

sth   i      1                            ; stop
dyn   f      1
set   f      10                           ; stop-store flag
cfb   h      exec
      k      1                            ; module 1
      stop
      rnotused

sth   i      2                            ; continue
dyn   f      2
res   f      10                           ; stop-store flag
cfb   h      exec
      k      1                            ; module 1
      continue
      rnotused

sth   ondest_1
anl   f      10                           ; stop-store flag
ETR   ;52

```

In den TR 52 und 54 des Beispiels wird im Prinzip nur das Ende der Bewegung abgewartet (sth ondest_1). Eine Eigenheit des GRAFTEC ist, dass nach jeder **nicht** erfüllten TR das Programm zum aufrufenden COB zurückkehrt und beim nächsten Programmzyklus die nicht erfüllte TR wieder **vollständig** abgearbeitet wird. Dies wird im Programm dazu verwendet, den Display dauernd aufzufrischen.

Soll nun die Impulsausgabe während einer Bewegung unterbrochen (gestoppt) werden, muss dies auch in der TR, welche das Ende der Bewegung abwartet, geschehen. Wird der Stop-Befehl aktiviert, wird auch das "ondest_x"-Flag = H und die TR würde als erfüllt betrachtet. Der Stop-Befehl wird deshalb in einem Hilfs-Flag (stop-store flag) zwischengespeichert und der logische Zustand dieses Hilfs-Flags wird zur Entscheidung, ob wirklich der ganze Weg gefahren wurde, herangezogen. Mit der Funktion "Weiter" (continue) wird dieses Hilfs-Flag wieder zurückgeschaltet.

Es kann auf diese Weise, im Bedarfsfall, eine Bewegung sauber, d.h. mit Brems- und Anfahrtsrampe, unterbrochen werden ohne dass Schritte verloren gehen.

Anwendung des Referenz- und der Endschalter

Dieses Beispiel liegt auf der Diskette PCD9.H21E.

```
; *****
; * Programm-Beispiel für FB's H210 PG4 *
; *****
;
; FileName :      H210_Ex4.SRC
; Autor :       Nguyen T.D.  03.01.99
;
; System-Anforderungen:
; 1 Eingangs-Modul auf Adresse 0,
; 1 H210-Module auf der Adresse, welche in der Datei
;   D2H210_B.MBA definiert wurde.
;
; Beschreibung:
;
; - Dieses Programm beschreibt die "Homing"-Prozedur.
;   Ist Eingang I3 = H, wird die Homing-Prozedur abgearbeitet.
;   Die Endschalter (LS) sollten für die Homing-Prozedur
;   verdrahtet sein
;   Falls keine LS vorhanden sind, gilt für die Simulation:
;   E1 = I5 = LS1, E2 = I6= LSRef, E3 = I7= LS2.
;   Sind LS vorhanden, sind LS1, LSRef und LS2 korrekt
;   anzuschliessen.
; - Dieses Programm kann entweder zum Beispiel H210_Ex2.SFC
;   oder H210_Ex3.SFC gelinkt werden.
;   Um das Motion-Programm in Betrieb zu nehmen, sind die
;   Eingänge der entsprechenden Programme zu übernehmen, z.B.
;   I0 = start, I1 = stop, I2 = continue und es sind die
;   Beschreibungen im Kapitel 7 des Handbuchs zu beachten.
; - Der Anwender hat die Parameter im FB INIT im IST der
;   Beispiele H210_Ex2.SFC oder H210_Ex3.SFC anzupassen:
;   Set Emergency stop = 1 und Limit switch = 1.
; - Der Arbeitsbereich ist mit LS1 und LS2 begrenzt. Ausserhalb
;   dieses Bereichs sind das Flag LS1Trig1 und der Ausgang A3
;   dieses Moduls gesetzt. Der Ausgang A3 ist auch gesetzt,
;   wenn der Notaus aktiviert ist (E0 = 0). In der Praxis
;   kann im Anwenderprogramm ein PCD-Ausgang für Alarm- und
;   allgemeine Warnungen gesetzt werden.
; - Der Notstop-Eingang ist mit E0 = I4 simuliert.
;
```

```

$include D2H210_B.EQU      ; for PG4 V2.0 the complete path
                           ; is no more necessary
$group H210

XOB          16            ; startup block

CFB          Exec         ; FB Exec
              K 1         ; module number
              ResOut3     ; reset output A3
              rNotUsed    ; empty register

EXOB

;-----

COB          0            ; cyclic block
              0

LD           R 100        ; Load Offset position for REF point
              0

STH          I 3          ; Start HOMING Procedure
DYN          F 1000
CFB          H HOME      ; FB Home
              K 1         ; module number
              R 100        ; offset register
              1           ; search direction (0=Neg.,1=Pos.)
              0           ; free travel direction
(0=Neg.,1=Pos)
              1           ; frequency range
              16          ; Vmin
              80          ; Vmax
              100         ; acceleration
              400         ; time out (Max. time in sec before
                           ; stop occurs

STH          LSxTrig_1    ; LS1 or LS2 overdrived ?
ORH          EmergTrig_1 ; or Emergency Stop activated
CFB          H Exec       ; FB Exec
              K 1         ; module number
              SetOut3     ; command: set output A3
              rNotUsed    ; empty register

ECOB

$endgroup
;-----

```

Notizen

9.2 Typisches Anwendungsbeispiel in reinem FUPLA

In Vorbereitung

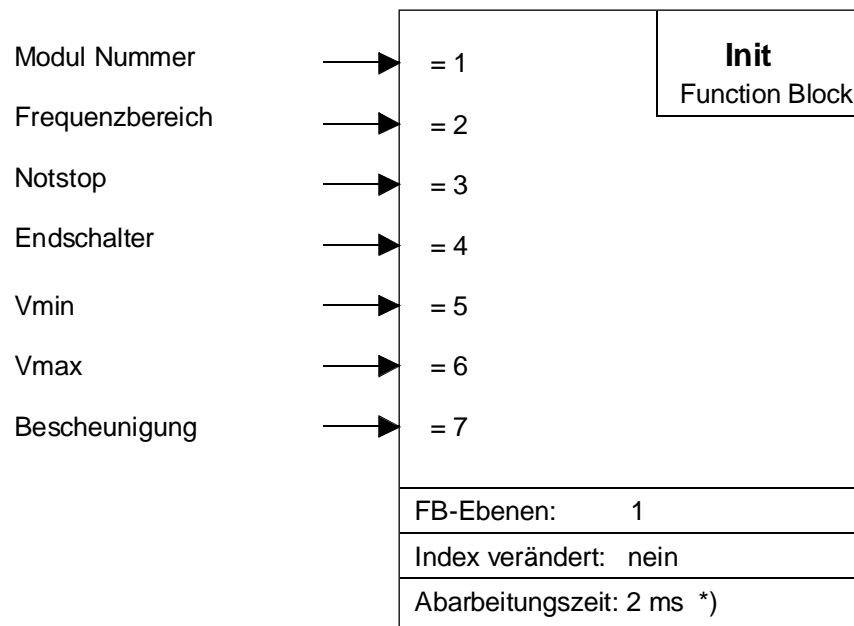
9.3 Typisches Anwendungsbeispiel in GRAFTEC und FUPLA

In Vorbereitung

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

Der Funktionsblock 'Init'

Init **FB:** Initialisierung eines H210 Moduls



*) mit PCD2.M120 gemessen

Funktionsbeschreibung:

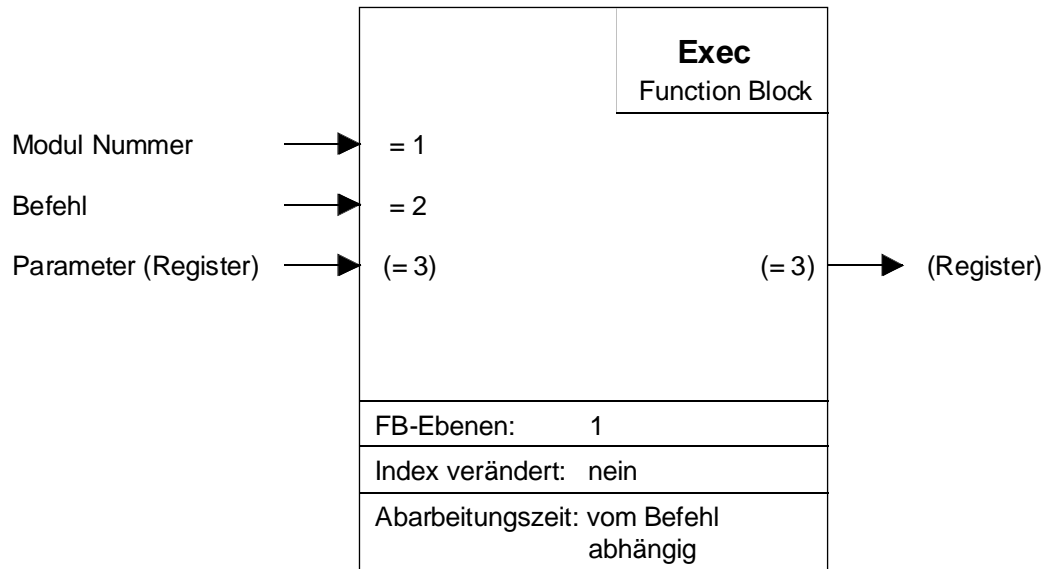
Mit diesem FB werden die Einstellungen des PCD2.H210-Moduls definiert und die Basisadresse aus der Datei D2H210_B.MBA gelesen. Die Werte für Vmin, Vmax und Acc können jederzeit geändert werden und gelten nicht als Maximalwerte.

Parameter 1 muss als Konstante, alle anderen Parameter als Zahl übergeben werden.

Param.	Bezeichnung	Typ	Format	Wert	Bemerkung
= 1	Modul Nr.	K	K n	1 – 16	
= 2	Frequenzbereich		ganzzahl	0 - 3	0 = 9.5 – 2431 Hz 1 = 19 – 4864 Hz 2 = 38 – 9727 Hz 3 = 76 – 19'454 Hz
= 3	Notstop Konfigurierung		ganzzahl	0 / 1	0 = aus / 1 = ein
= 4	Endschalter Konfig.		ganzzahl	0 / 1	0 = aus / 1 = ein
= 5	Vmin (Start-Stop Freq.)		ganzzahl	1 - Vmax	Eingabe kontrolliert
= 6	Vmax (Frequenz nach der Hochlauframpe)		ganzzahl	Vmin - 255	Eingabe kontrolliert
= 7	Acc (Beschleunigung)		ganzzahl	255 - 1	Eingabe kontrolliert

Der Funktionsblock 'Exec'

Exec: **FB:** Ausführung eines Befehls für das H210-Modul



Funktionsbeschreibung:

Mit diesem FB werden die Ausführungsbefehle zum PCD2.H210-Modul geschickt.

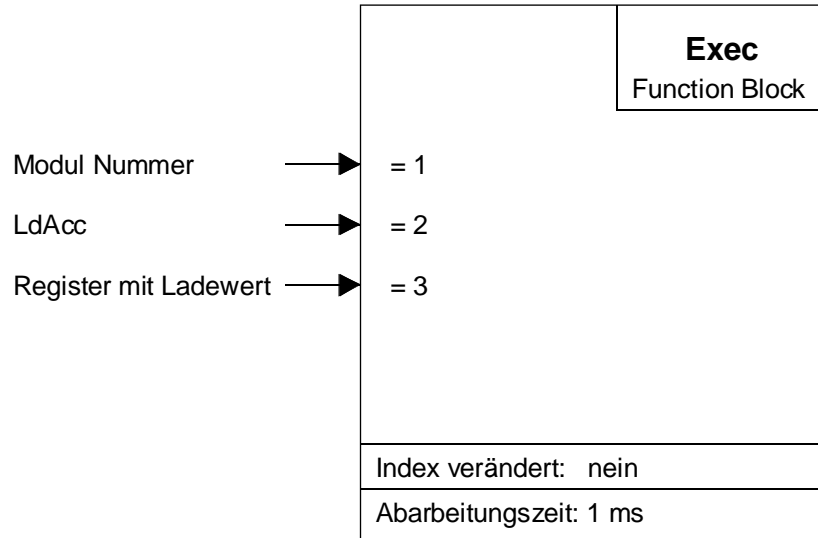
Die Modul-Nr. (Parameter 1) muss eine Konstante sein (k 1...k 16).
Die Basisadresse wird in der Datei 'D2H210_B.MBA' definiert.
Die FBs unterstützen max. 16 PCD2.H210-Module pro PCD-System.

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

Der Parameter eines Befehls (z.B. der Beschleunigungswert beim Befehl LdAcc) 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.H210 (FB-Parameter)

LdAcc **Befehl:** Laden der Beschleunigung



Funktionsbeschreibung:

Mit diesem Befehl wird die Beschleunigung ins Eingangsregister geladen. Diese wird beim nächsten Befehl 'Start' übernommen. Die Beschleunigungs- und Bremsrampen sind symmetrisch. Die Beschleunigung ist vom gewählten Frequenzbereich abhängig (Siehe Tabelle in Abschnitt 7.1.3). Zur Berechnung der effektiven Beschleunigung kann folgende Formel verwendet werden:

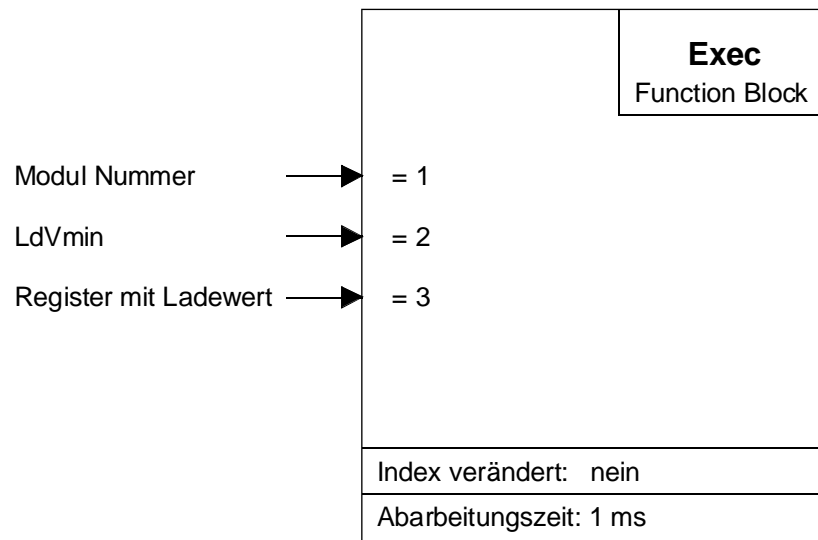
$$\text{eff. Beschl. (Frequenzbereich = 3)} = \frac{76.30 \text{ [Schritte/s]} * 16000 \text{ [1/s]}}{\text{Beschleunigungswert}} \text{ [Schritte/s}^2\text{]}$$

usw. mit 38.15 und 19.08 [Schritte/s] bis,

$$\text{eff. Beschl. (Frequenzbereich = 0)} = \frac{9.54 \text{ [Schritte/sec]} * 16000 \text{ [1/s]}}{\text{Beschleunigungswert}} \text{ [Schritte/s}^2\text{]}$$

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdAcc				
= 3	Beschleunigung	R	ganzzahl	255 - 1	255 = min ; 1 = max Werte ausserhalb des Bereichs werden nicht abgefangen

LdVmin**Befehl:** Laden der Start-Stop Frequenz**Funktionsbeschreibung:**

Mit diesem Befehl wird die Start-Stop Frequenz (Vmin) ins Eingangsregister geladen. Diese wird beim nächsten Befehl 'Start' übernommen.

Die Abstufung der Frequenz, d.h. die Schritte von einer Frequenzstufe zur nächsten ist vom gewählten Bereich abhängig:

Für den Bereich 0 (9.5 bis 2431 Hz) beträgt die Auflösung 9.54 Hz

Für den Bereich 1 (19 bis 4864 Hz) beträgt die Auflösung 19.08 Hz

Für den Bereich 2 (38 bis 9727 Hz) beträgt die Auflösung 38.15 Hz

Für den Bereich 3 (76 bis 19'454 Hz) beträgt die Auflösung 76.30 Hz.

Siehe auch Tabelle im Abschnitt 7.1.3

Berechnung der effektiven Start-Stop Frequenz:

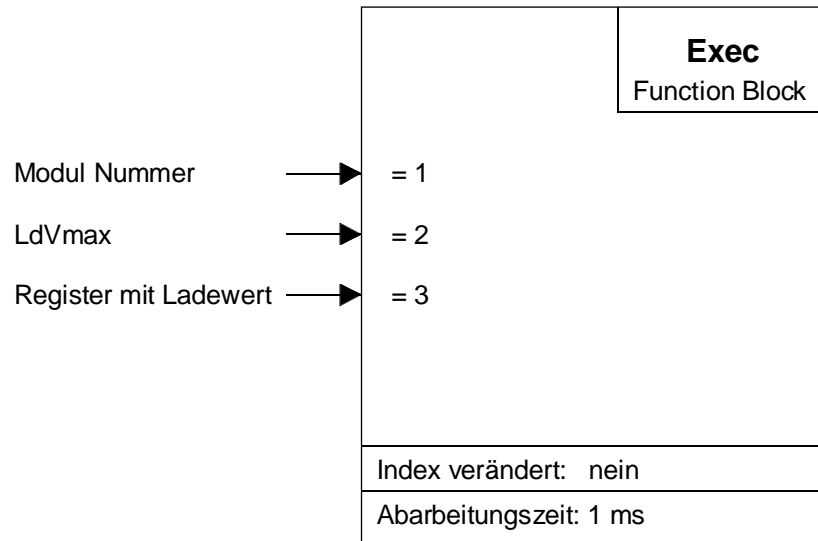
Für Bereich = 0: Start-Stop Frequenz = 9.54 Hz * Wert Vmin

usw. für 19.08 Hz und 38.15 Hz bis

für Bereich = 3: Start-Stop Frequenz = 76.30 Hz * Wert Vmin

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdVmin				
= 3	Start-Stop Frequenz	R	ganzzahl	1-Vmax	Werte ausserhalb des Bereichs werden nicht abgefangen

LdVmax**Befehl:** Laden der Maximalgeschwindigkeit**Funktionsbeschreibung:**

Mit diesem Befehl wird die Maximal-Frequenz (Vmax), welche nach einer Anfahrtsrampe erreicht werden kann, ins Eingangsregister geladen. Sie wird beim nächsten Befehl Start übernommen.

Die Abstufung der Frequenz, d.h. die Schritte von einer Frequenzstufe zur nächsten ist vom gewählten Bereich abhängig:

Für den Bereich 0 (9.5 bis 2431 Hz) beträgt die Auflösung 9.54 Hz

Für den Bereich 1 (19 bis 4864 Hz) beträgt die Auflösung 19.08 Hz

Für den Bereich 2 (38 bis 9727 Hz) beträgt die Auflösung 38.15 Hz

Für den Bereich 3 (76 bis 19'454 Hz) beträgt die Auflösung 76.30 Hz.

Siehe auch Tabelle im Abschnitt 7.1.3

Berechnung der effektiven Frequenz:

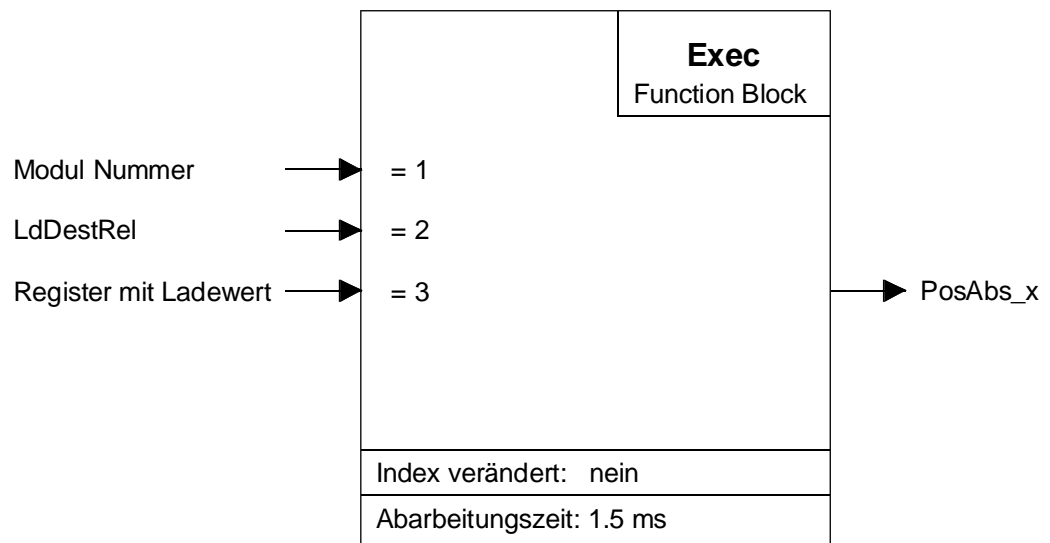
Für Bereich = 0: Frequenz = 9.54 Hz * Wert Vmax

usw. für 19.08 Hz und 38.15 Hz bis

für Bereich = 3 : Frequenz = 76.30 Hz * Wert Vmax

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdVmax				
= 3	Maximal-Frequenz	R	ganzzahl	Vmin - 255	Werte ausserhalb des Bereichs werden nicht abgefangen

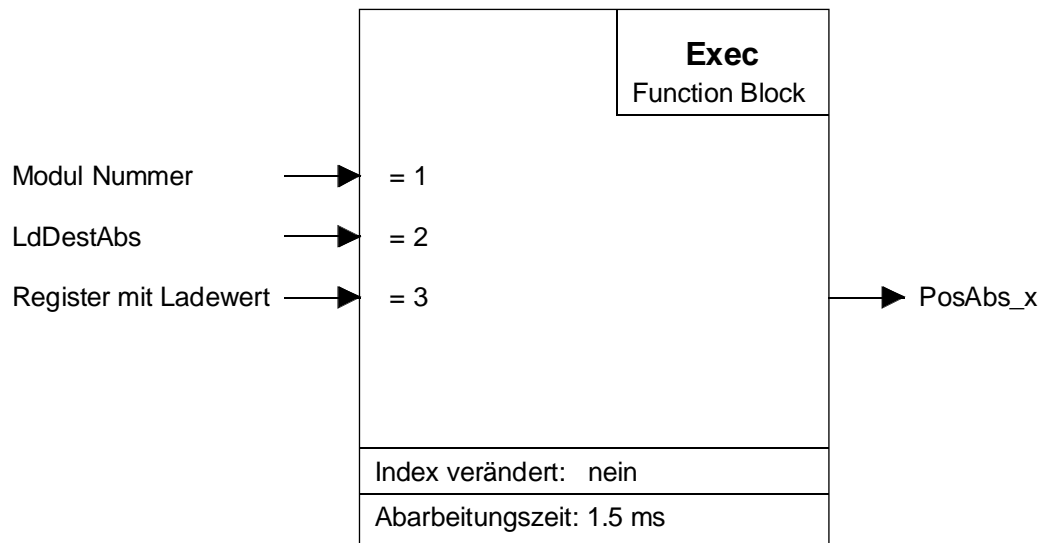
LdDestRel**Befehl:** Laden des relativen Verfahrenweges**Funktionsbeschreibung:**

Mit diesem Befehl wird die Anzahl Schritte für die nächsten Bewegung ins Eingangsregister geladen. Diese wird beim nächsten Befehl 'Start' übernommen. Dieser Befehl arbeitet immer im Relativ-Modus. Der Eingabewert ist immer ganzzahlig (positiv oder negativ) und darf 2^{24} nicht überschreiten.

Dieser Befehl berechnet auch die absolute Position (Register 'PosAbs_x'), welche nach der Ausführung der Bewegung erreicht wird.

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdDestRel				
= 3	Ziel (relativ)	R	ganzzahl	24 Bit	± 16777215
PosAbs_x	Absolute Position	R	ganzzahl	32 Bit	$-2^{31} .. +2^{31} - 1$

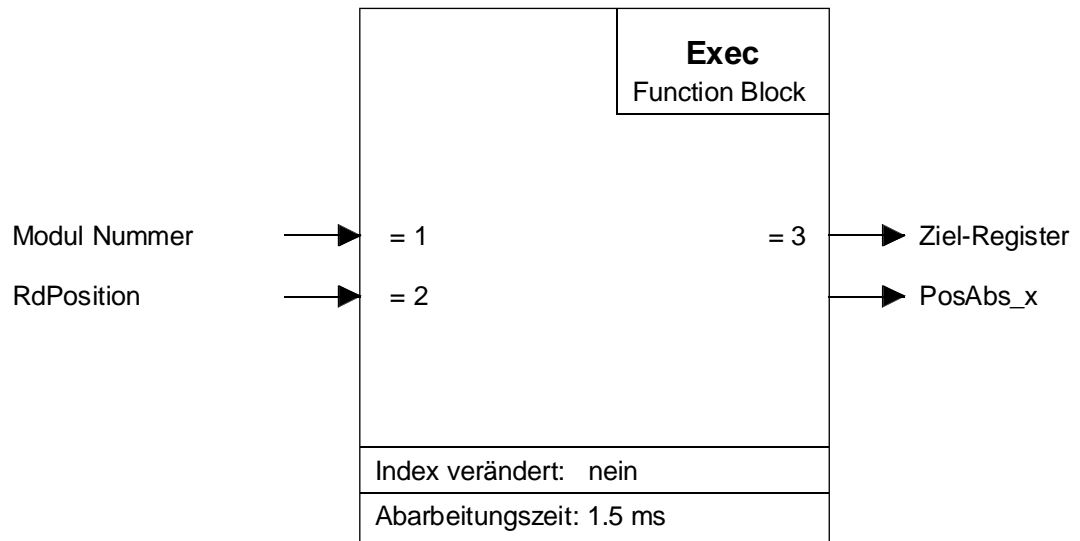
LdDestAbs**Befehl:** Laden der absoluten Endposition**Funktionsbeschreibung:**

Mit diesem Befehl wird die Zielposition für die nächste Bewegung ins Eingangsregister geladen. Diese wird beim nächsten Befehl 'Start' übernommen. Der Befehl berechnet die relative Bewegung zur letzten Position. Der Wert dieser relativen Bewegung darf 2^{24} nicht überschreiten. Der Eingabewert ist immer ganzzahlig und positiv.

Dieser Befehl berechnet auch die absolute Position (Register 'PosAbs_x'), welche nach der Ausführung der Bewegung erreicht wird.

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdDestAbs				
= 3	Ziel (absolut)	R	ganzzahl	32 Bit	$-2^{31} .. +2^{31} - 1$ aber max. 24 Bit als relative Bewegung
PosAbs_x	Absolute Position	R	ganzzahl	32 Bit	$-2^{31} .. +2^{31} - 1$

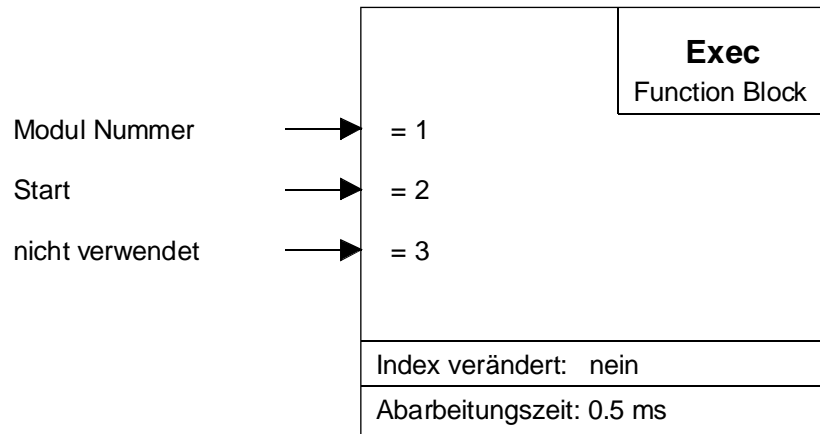
RdPosition**Befehl:** Lesen der aktuellen (noch verbleibenden) Position**Funktionsbeschreibung:**

Mit diesem Befehl wird der Zähler des Moduls gelesen. Der Zählerstand gibt an, wie viele Impulse bis zum Ende der Bewegung noch auszugeben sind. Das Resultat ist immer positiv.

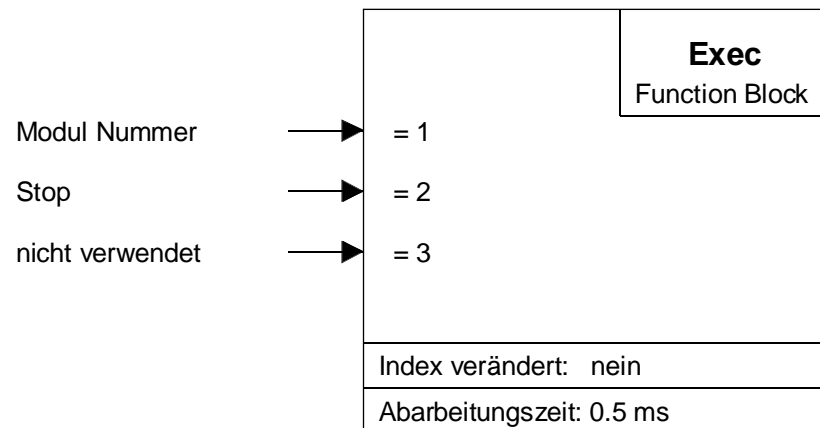
Dieser Befehl berechnet auch die aktuelle Absolutposition, welche im Register 'PosAbs_x' aufgefrischt wird.

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdPosition				
= 3	PCD-Register mit Anzahl Impulsen für den noch verbleibenden Weg.	R	ganzzahl	24 Bit	0 .. 16777215
PosAbs_x	Absloute Position	R	ganzzahl	32 Bit	$-2^{31} .. +2^{31} - 1$

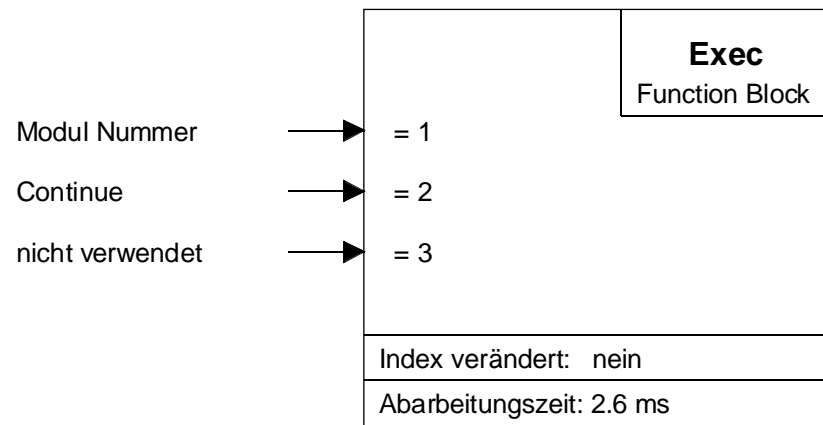
Start**Befehl:** Starten der Bewegung**Funktionsbeschreibung:**

Mit diesem Befehl werden die Bewegungsparameter 'Vmin', 'Vmax', 'Acc' und 'Dest' in die Arbeitsregister übertragen und die Bewegung gestartet. Dieser Befehl darf nur ausgeführt werden, wenn die vorherige Bewegung beendet ist (OnDest_x = H).

Stop**Befehl:** Stoppen einer Bewegung**Funktionsbeschreibung:**

Mit diesem Befehl kann eine laufende Bewegung mit der normalen Bremsrampe gestoppt werden. Sobald die Bremsrampe beendet ist und der Motor stillsteht, wird der Eingang 'OnDest_x' = H. Eine unterbrochene Bewegung kann mit dem Befehl 'Continue' beendet werden. Das Modul ist nach einem Stop nicht blockiert und kann auch durch einen neuen 'Start'-Befehl auf eine neue Position verfahren werden. Siehe Beispiel Seite 9-7.

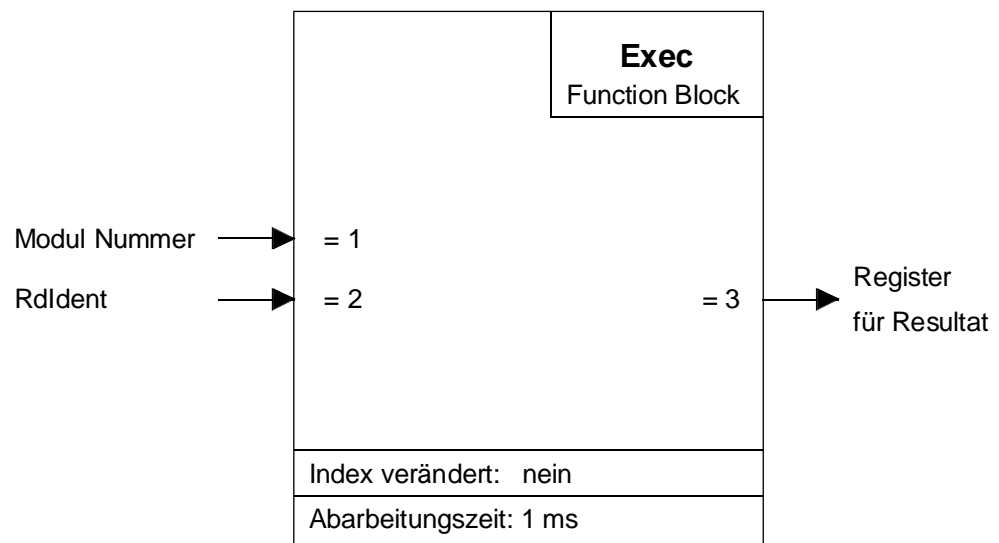
Continue **Befehl:** Weiterlaufen lassen einer gestoppten Bewegung



Funktionsbeschreibung:

Dieser Befehl lässt eine mit 'Stop'unterbrochene Bewegung weiter laufen. Zuerst wird die verbleibende Position gelesen, dann der verbleibende Weg geladen und die Bewegung gestartet.

Der Befehl 'Continue' darf nur gesendet werden, wenn der Motor zum Stillstand gekommen ist (Ondest_x = H), da sonst die Endposition nicht korrekt angefahren wird.

RdIdent**Befehl:** Lesen der Modulidentifikation**Funktionsbeschreibung:**

Mit diesem Befehl kann die korrekte Funktion des Moduls PCD2.H210 überprüft sowie die FPGA-Version ausgelesen werden. Funktioniert das Modul korrekt, wird der Wert 32xx zurückgegeben. Siehe Tabelle unten. Wenn das Modul defekt sein sollte (oder falsch adressiert), wird der Wert 0 gelesen.

Beschreibung der beteiligten Ein- und Ausgabeelemente:

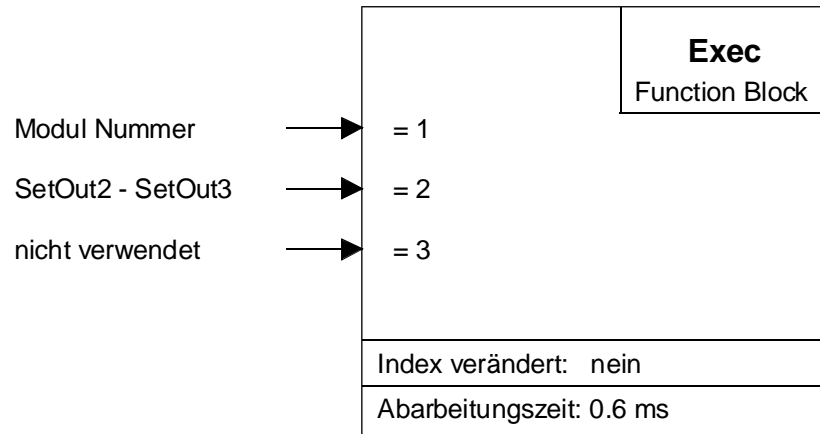
Par.	Bez./Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 -16	
= 2	Befehl: RdIdent				
= 3	Modulidentifikator H210	R	ganzzahl	12 Bit	0 → defekt

Tabelle der gültigen Identifikationen:

Wert	FPGA-Version
3200	Version HD0
3201	Version HD1
3202	Version HD2
...	...
3215	Version HDF

SetOut2 - SetOut3

Befehl: Ausgang 2 resp. 3 setzen

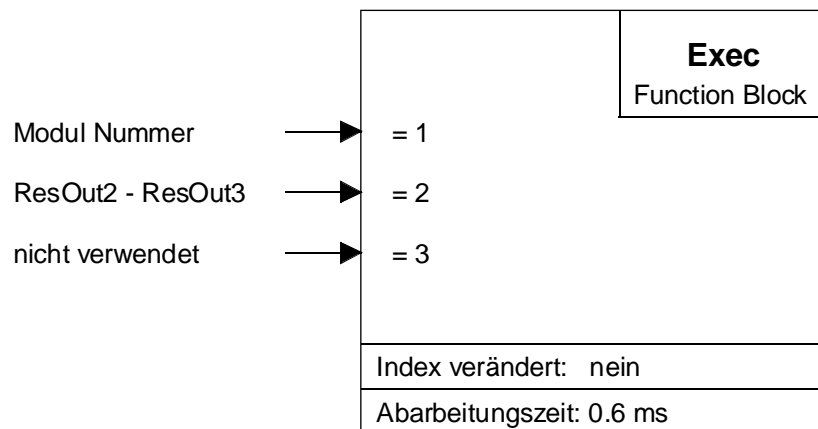


Funktionsbeschreibung:

Mit diesen Befehlen werden die digitalen Ausgänge 'A2' bzw. 'A3' gesetzt. Da die Ausgänge nicht direkt vom PCD1/2-Bus adressiert werden, können diese nicht zurückgelesen und von den Debugger-Befehlen 'Clear Outputs' resp. 'Clear All-Elements' auch nicht beeinflusst werden.

ResOut2 – ResOut3

Befehl: Ausgang 2 resp. 3 rücksetzen

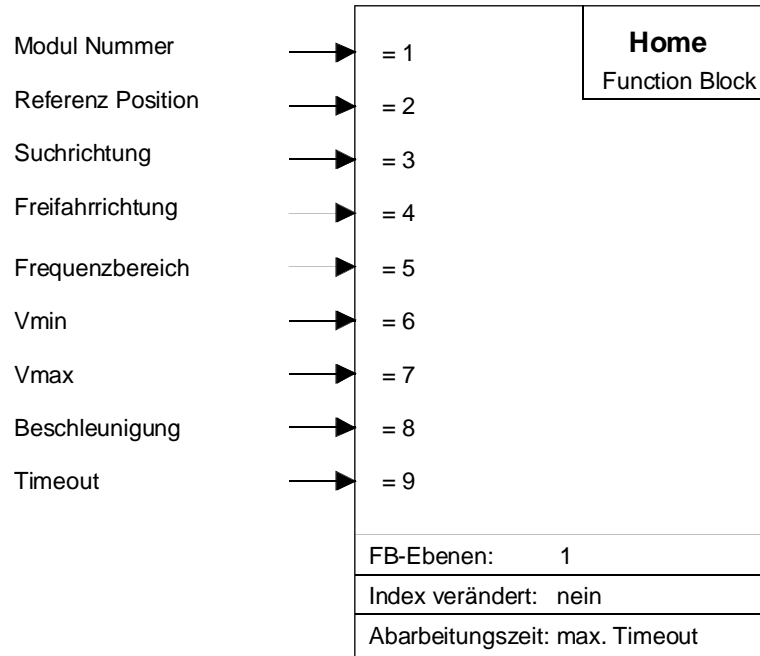


Funktionsbeschreibung:

Mit diesen Befehlen werden die digitalen Ausgänge 'A2' bzw. 'A3' rückgesetzt. Da die Ausgänge nicht direkt vom PCD1/2-Bus adressiert werden, können diese diese nicht zurückgelesen und von den Debugger Befehlen 'Clear Outputs' resp. 'Clear All Elements' auch nicht beeinflusst werden.

Der Funktionsblock 'Home'

Home **FB:** Suche die Referenzposition



Funktionsbeschreibung:

Mit diesem FB wird die Referenzfahrt definiert und gestartet. Der FB wird erst verlassen, wenn der Referenzschalter gefunden wurde resp. die Timeout Zeit erreicht wird. Das Modul muss initialisiert sein (FB Init), bevor der FB Home ausgeführt werden kann. Das Flag 'OnDest_x' zeigt das Ende der Referenzfahrt an. Für zusätzliche Erklärungen der Referenzfahrt siehe Kap. 5.4.

Par.	Bezeichnung	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K	K n	1 – 16	
= 2	Referenzposition *)	R	ganzzahl	32 Bit	Absolutposition
= 3	Suchrichtung		ganzzahl	0 / 1	0 = neg. / 1 = pos.
= 4	Freifahrrichtung		ganzzahl	0 / 1	0 = neg. / 1 = pos.
= 5	Frequenzbereich		ganzzahl	0 – 3	Siehe FB Init
= 6	Vmin für Home		ganzzahl	1 – Vmax	Eingabe kontrolliert
= 7	Vmax für Home		ganzzahl	Vmin – 255	Eingabe kontrolliert
= 8	Acc für Home		ganzzahl	255 – 1	Eingabe kontrolliert
= 9	Timeout (max. Zeit in sek. nach der abgebrochen wird)		ganzzahl	0 – 65535 [s]	0 = keine Begrenzung

*) Dieser Parameter ermöglicht die Definition eines Offset zur Referenzposition. (Siehe Seite 7-6 für die Definition eines beliebigen Punktes als Null-Position).

Notizen

Anhang B: Zusammenfassung aller Software-Elemente 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.H210

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: