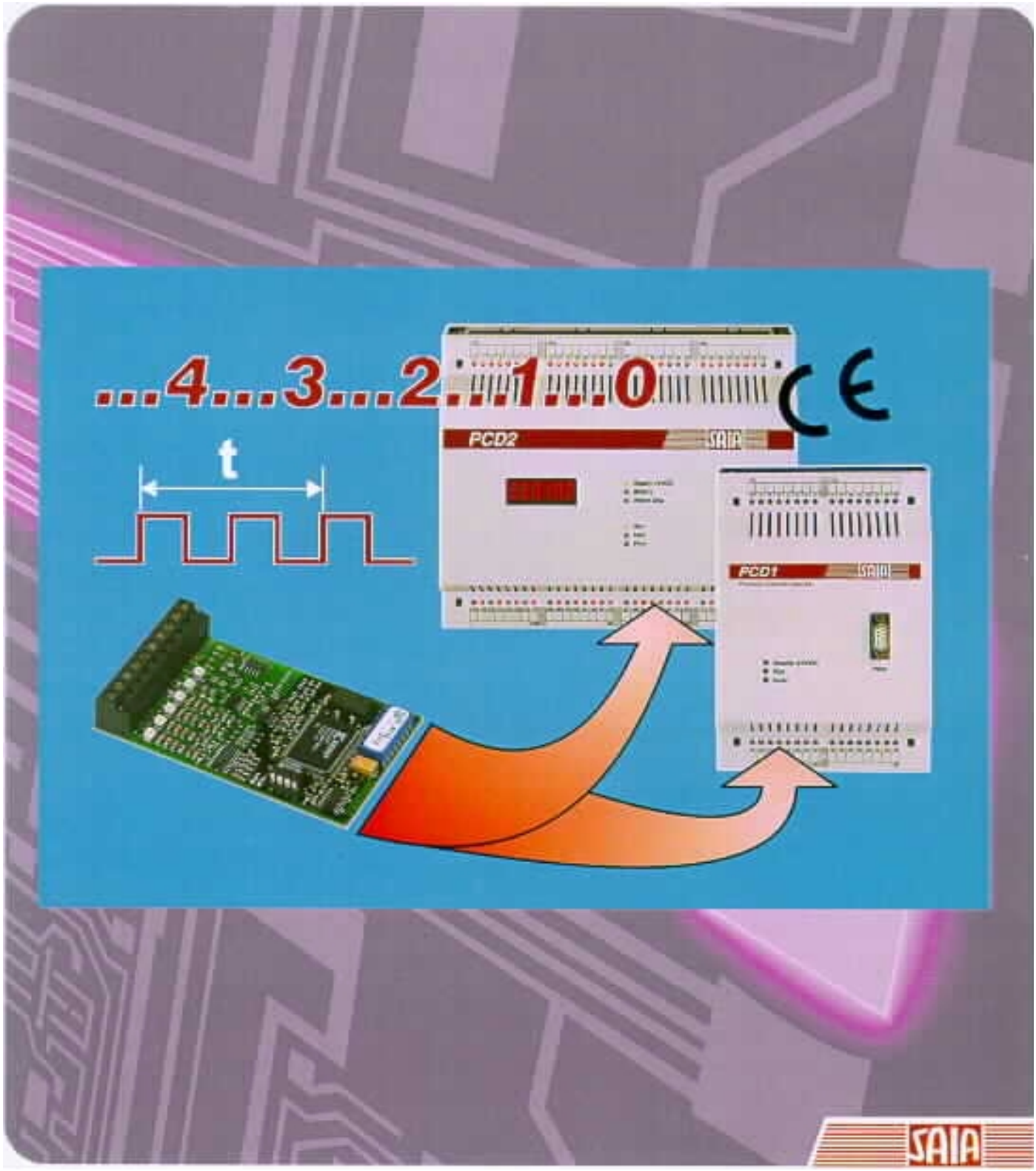
SAIA-Burgess Electronics

SWITCHES • MOTORS • CONTROLLERS

SAIA® PCD

Process Control Devices

PCD2.H110
Universelles Zähl- und Messmodul



Deutsche Ausgabe 26/755 D2

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

GB: Electronic Controllers Telefon 026 / 672 72 72
Telefax 026 / 672 74 99

SAIA-Burgess Gesellschaften

Schweiz
SAIA-Burgess Electronics AG
Freiburgstrasse 33
CH-3280 Murten
☎ 026 672 77 77, Fax 026 670 19 83

Deutschland
SAIA-Burgess Electronics GmbH
Daimlerstrasse 1k
D-63303 Dreieich
☎ 06103 89 060, Fax 06103 89 06 66

Österreich
SAIA-Burgess Electronics Ges.m.b.H.
Schallmooser Hauptstrasse 38
A-5020 Salzburg
☎ 0662 88 49 10, Fax 0662 88 49 10 11

Italien
SAIA-Burgess Electronics S.r.l.
Via Cadamosto 3
I-20094 Corsico MI
☎ 02 48 69 21, Fax 02 48 60 06 92

Frankreich
SAIA-Burgess Electronics Sàrl.
10, Bld. Louise Michel
F-92230 Gennevilliers
☎ 01 46 88 07 70, Fax 01 46 88 07 99

Niederlande
SAIA-Burgess Electronics B.V.
Hanzeweg 12c
NL-2803 MC Gouda
☎ 0182 54 31 54, Fax 0182 54 31 51

Belgien
SAIA-Burgess Electronics Belgium
Avenue Roi Albert 1er, 50
B-1780 Wemmel
☎ 02 456 06 20, Fax 02 460 50 44

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

Dänemark
Malthe Winje Automation AS
Håndværkerbyen 57 B
DK-2670 Greve
☎ 70 20 52 01, Fax 70 20 52 02

Norwegen
Malthe Winje Automasjon AS
Haukelivn 48
N-1415 Oppegård
☎ 66 99 61 00, Fax 66 99 61 01

Schweden
Malthe Winje Automation AB
Truckvägen 14A
S-194 52 Upplands Våsby
☎ 08 795 59 10, Fax 08 795 59 20

Suomi/ Finnland
ENERGEL OY
Atomitie 1
FIN-00370 Helsinki
☎ 09 586 2066, Fax 09 586 2046

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

Spanien
Tecnosistemas Medioambientales, S.L.
Poligono Industrial El Cabril, 9
E-28864 Ajalvir, Madrid
☎ 91 884 47 93, Fax 91 884 40 72

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

Polen
SABUR Ltd.
ul. Druzynowa 3A
PL-02-590 Warszawa
☎ 22 844 63 70, Fax 22 844 75 20

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

Issue : 22.11.2000

SAIA® Process Control Devices

Universelles Zähl- und Messmodul

PCD2.H110

Ausgabe 26/755 D2 - 04.99

Technische Änderungen vorbehalten

Anpassungen

Handbuch: PCD2.H110 - Universelles Zähl- und Messmodul - Ausgabe D2

Datum	Abschnitt	Seite	Beschreibung

Inhalt

Seite

Seite

Notizen

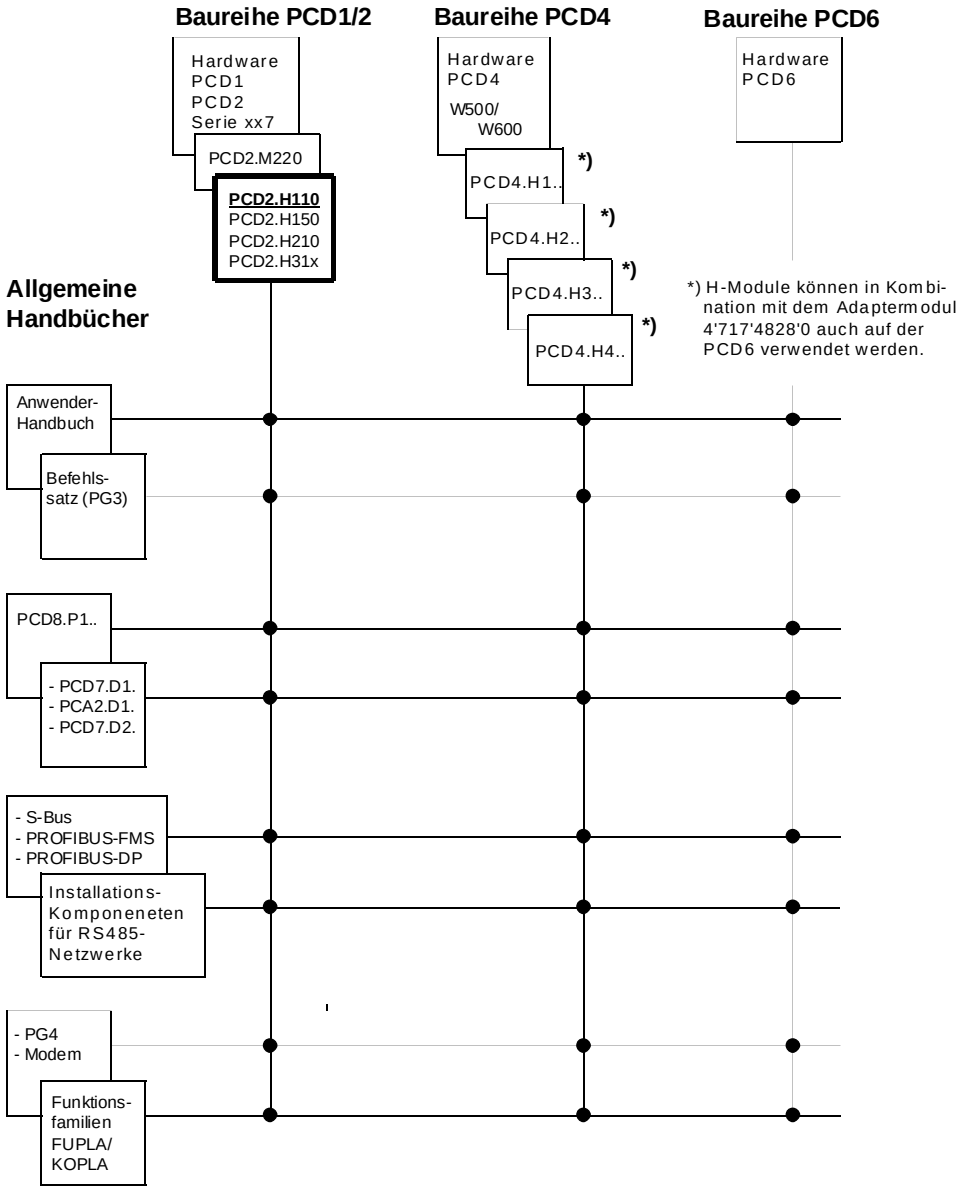
Wichtiger Hinweis:

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, United Laboratories (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

Die Prozess-Steuergeräte SAIA® PCD offerieren bereits in der Grundausführung 1600 Zählregister zu 31 Bit, mit welchen jedoch nur Frequenzen bis ca. 20 Hz erfasst werden können. Über die Interrupt-Eingänge werden 1 kHz und mit dem Zählmodul ..Hl00 bis 20 kHz erreicht

Das neue Zählmodul ..H110 erweitert nicht nur den Frequenzbereich bis 100 kHz, sondern erlaubt auch die genaue Messung von Frequenzen bis 100 kHz sowie der Dauer von Perioden und Impulsen bis zu einer Stunde.

Dank der beiden Zähleingänge A und B wird die Drehrichtung von Inkrementaldrehgebern erkannt, was das Modul ..H110 auch zur Achsensteuerung befähigt, sofern vom Modul keine geregelte Bewegung gefordert wird. Für die geregelte Steuerung von Servomotoren mit Anfahr- und Bremsrampen wird das Positioniermodul PCD2.H3.. empfohlen.

Das neue Zähl- und Messmodul ..H110 verwendet einen modernen FPGA-Baustein (Field Programmable Gate Array), welcher mittels steckbarem PROM auch für andere OEM-Aufgaben gezielt programmiert werden kann. Dazu sind nach aussen 4 Eingänge, 4 Ausgänge und 2 x 4 LED vorhanden.

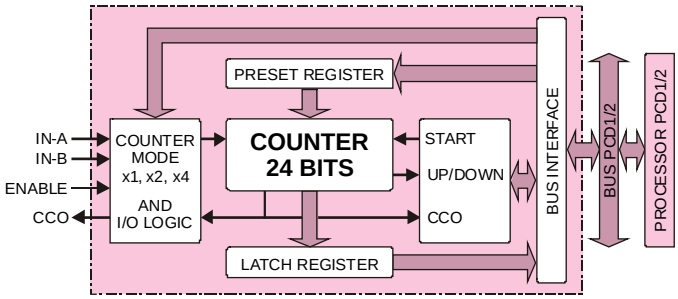
Für die Standardfunktionen des ..H110 stehen dem Anwender Funktionsbausteine als FB und FBoxen zur Verfügung.

1.2 Funktion und Anwendung

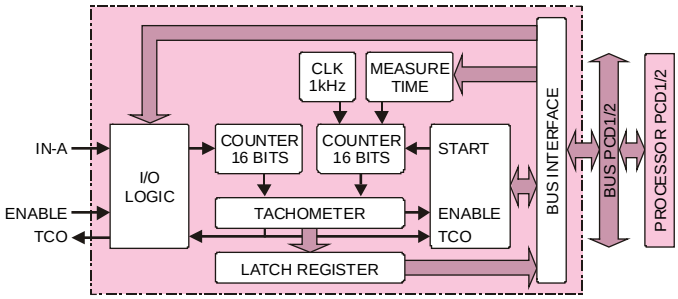
Dieses Low-Cost Modul lässt sich an jedem beliebigen E/A-Steckplatz einer PCD1 oder PCD2 einstecken.

Das Modul kann in verschiedenen Modi betrieben werden:

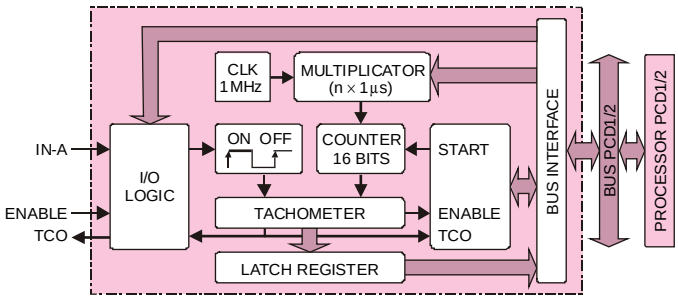
Blockschema als Zählmodul



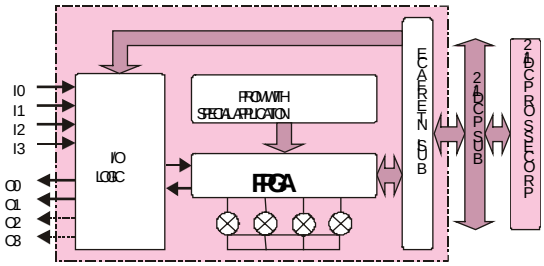
Blockschema als Frequenzmessung



Blockschema zur Messung der Perioden- oder Impulsdauer



Blockschema für spezielle OEM-Ausführungen



1.3 Die wichtigsten Eigenschaften

Bis zu 16 Module PCD2.H110 lassen sich in einer PCD2, bis zu 4 in einer PCD1, im Parallelbetrieb einsetzen.

Die Funktionen Zählen und Messen können im selben Modul gleichzeitig genutzt werden.

als Zählmodul

- Zählfrequenz bis 100 kHz
- Zählbereich 0 ... 16 777 215 (24 Bit)
- Vorwahlwert 0 ... 16 777 215 (24 Bit)
- Vor- oder Rückwärtszählen bis zum Vorwahlwert
- 2 digitale Eingänge A und B mit Drehrichtungserkennung
- 1 direkter Zählerausgang CCO
- wählbare Zählmodi

zur Frequenzmessung

- Frequenzbereich 500 Hz bis 100 kHz
- Messbereich 0 ... 65 535 (16 Bit)
- Genauigkeit >1‰. (abhängig von der Messzeit)
- Der schnelle Ausgang TCO kann bei Ende der Messung z. B. zum Auslösen eines Interrupts verwendet werden.

zur Messung von Perioden- oder Impulsdauer

- Frequenzbereich 0,27 mHz bis 500 Hz
- Perioden- oder Impulsdauer 2 ms bis 1h
- Der schnelle Ausgang TCO kann bei Ende der Messung z. B. zum Auslösen eines Interrupts verwendet werden.

1.4 Typische Einsatzgebiete

Für die kleinen Basissteuerungen PCD1 und PCD2 erweitert sich der Anwendungsbereich durch den Einsatz der neuen H-Module erheblich. Das ..H110 ermöglicht insbesondere:

- Schnelles Zählen von Impulsen proportional zu Mengen (Stücke, Energie-Einheiten usw.) bei geringer Belastung der Basis-CPU
- Ungeregelte Achsensteuerung beliebiger Antriebe mit Inkrementaldrehgebern
- Quarzgenaues Ermitteln von Geschwindigkeit, Drehzahl, Durchflussmenge usw.

Anwendung bei:

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

1.5 Programmierung

Dank praxisbezogener Funktionsbausteine (FBs und FBoxen) müssen für die verschiedenen Zähl- und Messmodi 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). (Die Verwendung des älteren Programmierwerkzeuges "PG3" ist nur noch eingeschränkt möglich, da neue Assemblerdirektiven verwendet werden, welche vom PG3 nicht verstanden werden).

Initialisierungsbefehl

INIT
- Wahl der Modul-Nummer
- Counter Konfigurierung
- Counter Preset
- Register Preset
- Enable Counter Konfigurierung
- CCO Konfigurierung
- IN-A Konfigurierung
- IN-B Konfigurierung
- Messung Konfigurierung
- Messung Wert
- Enable Messung Konfigurierung
- TCO Konfigurierung

Ausführungsbefehl

EXEC
- Wahl der Modul-Nummer
- eigentlicher Befehl
- Register für Ladewert bzw. Resultat

Die eigenlichen Befehle:

- LdCtPres Laden des Counter-Vorwahlwertes
- LdRegPres Laden des Vergleichswertes
- ModMsConf Konfigurierung des Mess-Modus'
- LdMsVal Messbereich laden
- RdCt Counter lesen
- RdMsImp Lesen des Messresultats bei Perioden-, Impulsdauer- oder Frequenzmessung als Zählwert
- RdMsUnit Lesen des Messresultats bei Perioden-, Impulsdauer- oder Frequenzmessung in Einheiten
- StartCt Start des Counters
- StartMs Start einer Messung
- StopMs Stoppen einer Messung
- RdIdent Lesen der Modul-Identifikation

Notizen

2. Technische Daten

2.1 Technische Daten der Hardware

Digitale Eingänge

Anzahl	4
Nennspannung:	24 V
Bereich "Low":	- 30 ... +5 V
Bereich "High":	+15 ... 30 V
Nur Quellbetrieb (positive Logik)	
Eingangsstrom (typisch)	6,5 mA
Eingangsfilter	150 kHz
Schaltungsart	galvanisch verbunden

Digitale Ausgänge

Anzahl	2
Strombereich	5 bis 500 mA (Leckstrom max.: 1 mA) (Lastwiderstand min.: 48Ω im Spannungsbereich von 5 bis 24 V)
Kurzschluss-Schutz	nein
Frequenz	≤ 100 kHz
Spannungsbereich	5 bis 32 V (externe Speisung)
Schaltungsart	galvanisch verbunden, nicht kurzschlussfest, der Plus wird geschaltet
Spannungsabfall (typisch)	< 0.5V bei 500 mA
Ausgangsverzögerung	kleiner als 1 µs, bei induktiver Last länger, als Folge der Freilaufdiode

Stromversorgung

Interne Speisung	
ab PCD1/2-Bus	5 VDC, max. 90 mA
Extern durch Anwender	24VDC (10 ... 32 VDC), max. 2A**,**
für alle Ausgänge	Restwelligkeit max. 10%

Betriebsbedingungen

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

LED-Anzeigen

Anzahl	6
LED 0:	Status des Eingangs "A"
LED 1:	Status des Eingangs "B"
LED 2:	Status des Eingangs "EnableC"
LED 3:	Status des Eingangs "EnableM"
LED 4:	Status des Ausgangs "CCO"
LED 5:	Status des Ausgangs "TCO"

Programmierung

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

Bestellangaben

PCD2.H110	Zähl- und Messmodul bis 100 kHz
PCD9.H11E	Software-Bibliothek mit Funktions-bausteinen

2.2 Elektrische Spezifikationen

Interne Leistungsaufnahme

+5V:	max. 90 mA
Uext	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

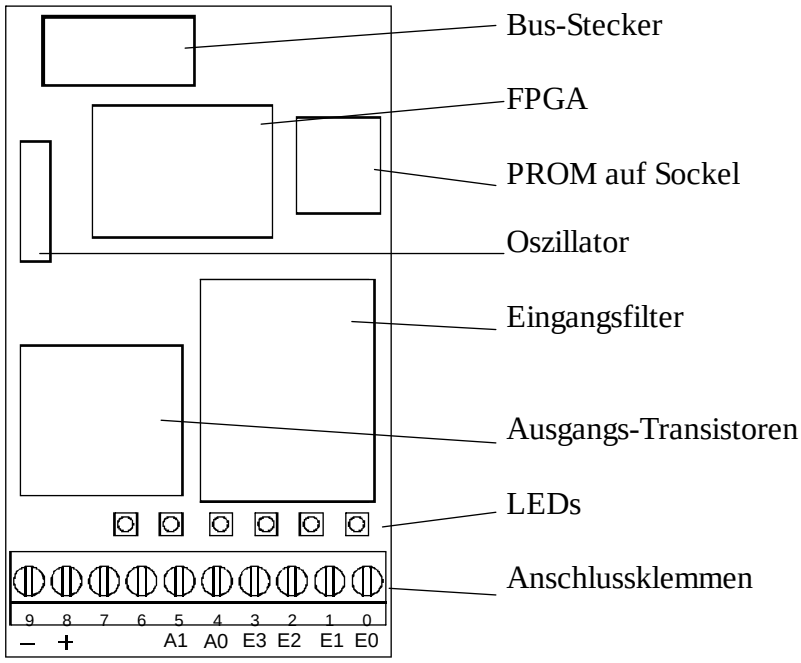
2 digitale Ausgänge (A0 und A1)
(siehe Abschnitt 2.1)

2.3 Funktionsspezifische Daten

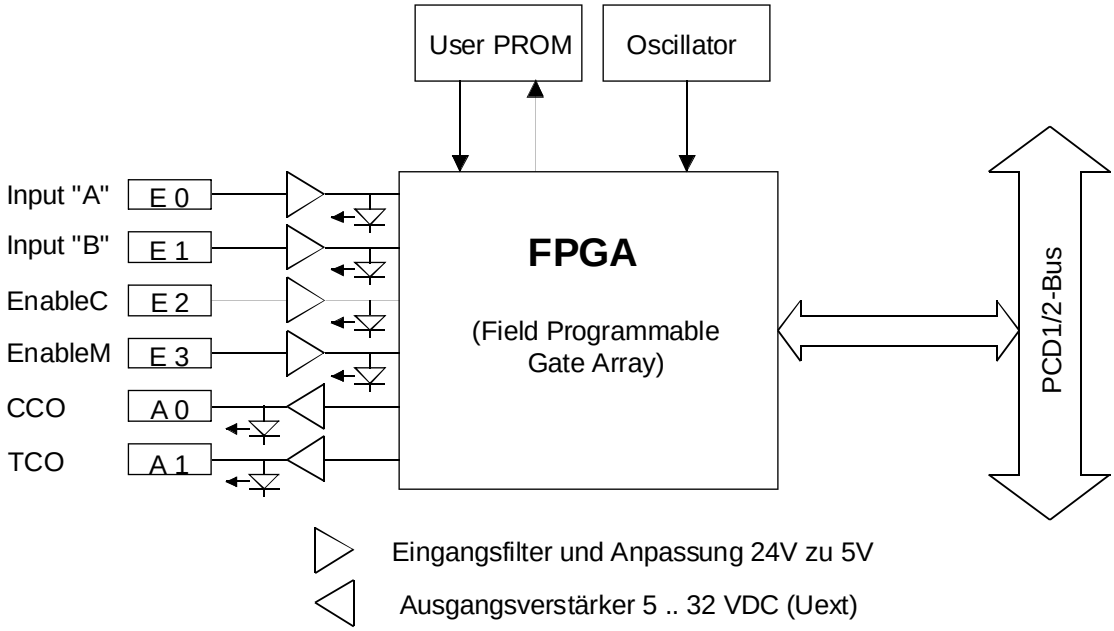
Anzahl Systeme	1	
Zählbereich		0 bis 16'777'215 (24 Bit)
Zählfrequenz		max. 100 kHz
Datensicherung		Alle Daten dieses Moduls sind flüchtig (nicht flüchtige PCD-Register stehen zur Verfügung)

3. Präsentation

Bestücktes Modul



Einfaches Blockschaltbild

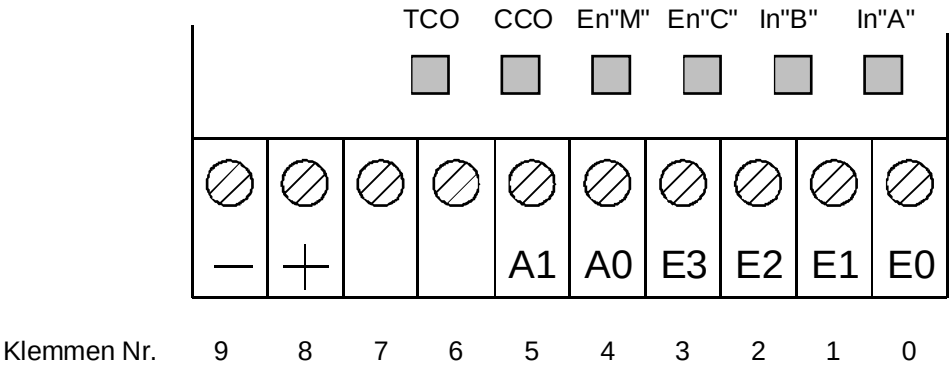


Notizen

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 numeriert (von rechts nach links)



Es sind auch die Blockschaltbilder im Abschnitt 1.2 zu konsultieren.

Eingänge:

Anzahl		4
Klemme 0 =	E 0:	Eingang "A" für Zählung und als Messwerteingang
Klemme 1 =	E 1:	Zähleingang "B" nur für Zählung
Klemme 2 =	E 2:	Eingang "Enable C" bei Verwednung des Moduls als Zähler
Klemme 3 =	E 3:	Eingang "Enable M" bei Verwendung des Moduls für Messungen

Ausgänge:

Anzahl		2
Klemme 4 =	A 0:	Ausgang "CCO" für Zähler
Klemme 5 =	A 1:	Ausgang "TCO" Messfunktionen

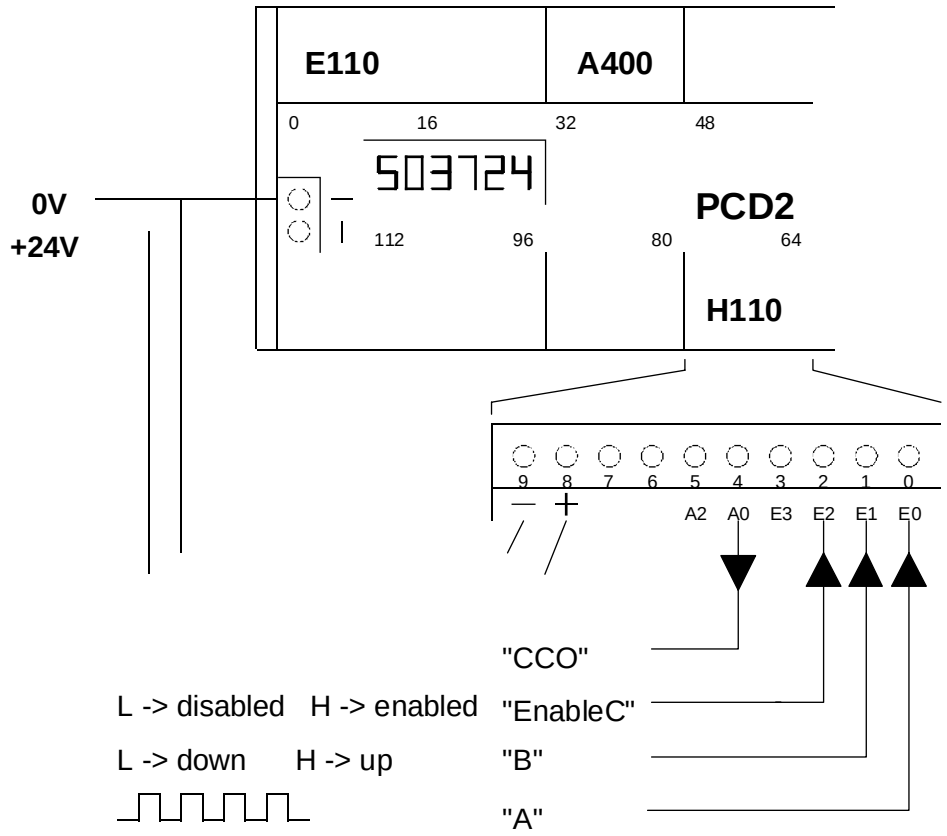
Speisung:

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

LED-Anzeigen

Anzahl	6
LED 0:	Status des Eingangs "A"
LED 1:	Status des Eingangs "B"
LED 2:	Status des Eingangs "EnableC"
LED 3:	Status des Eingangs "EnableM"
LED 4:	Status des Ausgangs "CCO"
LED 5:	Status des Ausgangs "TCO"

5. Der schnelle Einstieg



Minimal-Anordnung für die Verwendung des PCD2.H110 als Up-Down-Zähler.

Die einzelnen Elemente sind:

- PCD2 (oder PCD1) mindestens bestückt mit 1 PCD2.H110
  (1 PCD2.F510/530)
  (1 PCD2.E110)
  (1 PCD2.A400)
- Prellfreier Impulsgeber
- Speisegerät 24 VDC geglättet

Es kann auch ein 2-phasiger Inkrementalgeber (24V) an die Eingänge "A" und "B" angeschlossen werden. Es wird dann, je nach Drehrichtung, automatisch "up" oder "down" gezählt.

5.1 Einstieg mit Programmierung in IL

Um auf einfachste Weise ein Zähl- und Messmodul PCD2.H110 als Zähler 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.H110, trotzdem mit Warteschleifen gestaltet. In der Praxis ist für diese Art von Programmen immer eine GRAFTEC- oder in Zukunft eine FUPLA-Struktur zu wählen.

Aufgabe: Es soll nach dem Einschalten der PCD der Zähler auf 1000 geladen werden. Signale am Eingang "A" sollen, je nach gewählter Zählrichtung am Eingang "B" auf- oder abgezählt werden. Der Zählerstand soll am Displaymodul angezeigt oder im Debugger verfolgt werden (R 777).

Das Programm liegt im Projekt "h110" und hat den Namen "count.src".

Die einzelnen Parameter und die Einstellung der Basisadressen sind dem nachfolgenden Kapitel 6 zu entnehmen. Das Anwenderprogramm kann dann z.B. wie folgt gestaltet werden:

(Ausführliche, korrekt strukturierte Beispiele sind den einzelnen Kapiteln dieses Handbuchs zu entnehmen und sind auch auf der Diskette PCD9.H11E enthalten).

Die FBs (IL für PG4) sind auf der Diskette PCD9.H11E. 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.

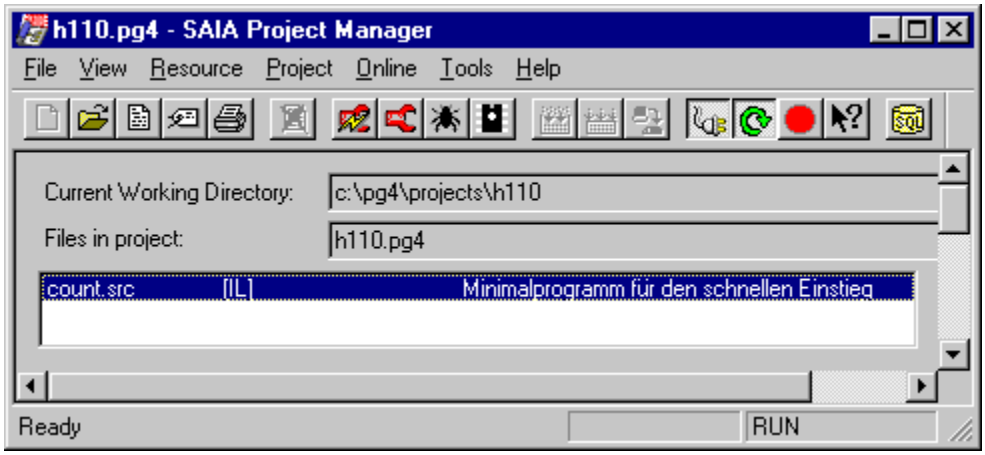
Die Anzahl Module (1) und die Adresse des PCD2.H110-Moduls (Adresse 64) muss in die Datei 2D2H110_B.MBA eingetragen werden:

```
NbrModules   EQU   1      ; No. of H110 modules used (0...16)

BA_1         EQU   64     ; Base address of module 1
```

Diese Datei (2D2H110_B.MBA) muss im Projektverzeichnis dieses Beispiels liegen, d.h. diese Datei ist aus dem Verzeichnis 'PG4_FB' der Diskette von Hand ins aktuelle Projektverzeichnis zu kopieren.

```
$include d2h110_b.equ
$group h110

xob       16

ld      r 999      ; PCD-Register mit
          1000     ;  Startwert für Counter
ld      r 998      ; PCD-Register mit Wert
          0        ;  für Vergleichs-Register

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1    Modul-Nummer
          0        ;  Par 2    Zähl-Modus "x1"
        r 999      ;  Par 3    Startwert für Counter
        r 998      ;  Par 4    Wert für Vergleichs-Register
          0        ;  Par 5    EnableC "statisch-normal"
          0        ;  Par 6    CCO "statisch-normal"
          0        ;  Par 7    Input A "normal"
          0        ;  Par 8    Input B "normal"
          0        ;  Par 9    nicht verwendet für Counter
          0        ;  Par 10   nicht verwendet für Counter
          0        ;  Par 11   nicht verwendet für Counter
          0        ;  Par 12   nicht verwendet für Counter

cfb       exec     ; Start Counter
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

exob
; ----------------------------------------

cob       0        ; eigentliches Anwender-Programm
          0

cfb       exec     ; Read Counter
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  Register 777

dsp     r 777      ; Anzeige an Displaymodul

$endgroup

ecob
```

Wie bereits erwähnt, hat das Programm den Namen "count.src" und liegt im PG4 im Projekt "h110".



Mit 'Project' - 'Build' wird das Programm verarbeitet, in die PCD geladen und in "Run" geschalten. Der aktuelle Zählerstand wird unmittelbar am Displaymodul angezeigt. Ist kein Displaymodul vorhanden, kann der Wert im PCD-Register R 777 im Debugger verfolgt werden. (Display Register 777 <Space> Refresh <CR>)

5.2 Einstieg mit Programmierung in FUPLA

In Vorbereitung

Notizen

6. Die 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.H11E 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. Anwederprogramme können jedoch auch in reinem BLOCTEC oder in reinem FUPLA geschrieben werden.

6.1 Programmierung in IL mit FBs

6.1.1 Das IL-Paket (Installation der FB)

Die Bestellnummer der Diskette lautet PCD9.H11E. Die Diskette enthält die folgenden Verzeichnisse:

- APPSDIR : enthält alle Helps
- FB : enthält die .SRC- und .EQU-Dateien des H110
- FBOX : enthält die FBoxen zum H110
- 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. Für alle andern PG4-Versionen ist die Datei 'Readme' zu konsultieren. (Das Paket enthält auch die 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.H11E 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:

D2H110_B.SRC	FB-Quell-Code	read-only Datei
D2H110_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
D2H110_B.HLP	FB-Help-Datei

Diese 2 Dateien liegen im Verzeichnis A:\APPSDIR und werden im PG4 ins Verzeichnis ...\PG4 kopiert.

Die Datei **D2H110_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 '**D2H110_B.MBA**' ist nachfolgend gezeigt:

Datei: **D2H110_B.MBA** (MBA = Modul Basis Adresse)

```
;
; This file can be modified by the user
;
; Basis addresses defined by the user
; -----------------------------------
$group H110
NbrModules      EQU     1       ; No. of H110 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
```

Bei 'NbrModules EQU x' ist die Anzahl PCD2.H110-Module anzugeben. Danach sind die Hardware-Basisadressen der verwendeten PCD2.H110-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 numerieren. Werden z.B. 3 Stk. H110-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 H110 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-H11" und das eigentliche Anwenderprogramm den Namen "count-01.sfc".

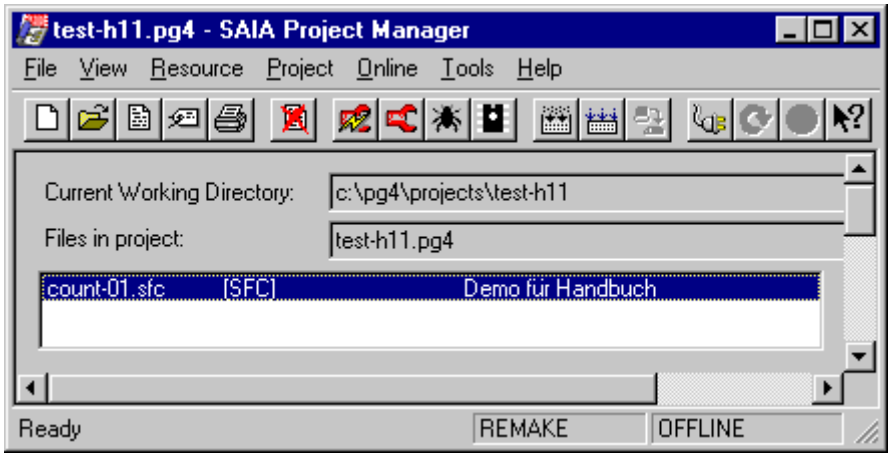
```
C:\PG4 \FB          \D2H110_b.equ
                    \D2H110_b.src
       \...
       \FBOX        \...
       \GALEP3      \...
       \PROJECTS    \FUP_E            (Demo Beispiel PG4)
                    \GRAF_E           (Demo Beispiel PG4)
                    \TEST-H11    \D2H110_b.mba
                                 \count-01.sfc
       \...
       \D2H110_b.hlp
```

Das Anwenderprogramm für den H110-Teil präsentiert sich folgendermassen :

```
$include D2H110_b.equ
$group H110

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.



6.1.2 Die einzelnen FBs

Das ganze Paket besteht grundsätzlich aus 2 FBs mit Parametern:

- INIT Initialisierung FB mit 12 Parametern
- EXEC Ausführung FB mit 3 Parametern

Der Aufruf des FB 'INIT' präsentiert sich z.B. wie folgt:

```
CFB    init
     k 1           ; Par. 1:   Modul-Nummer     (k 1 - k 16)
       0           ; Par. 2:   Counter Konfigurierung (0 - 4)
     r 100         ; Par. 3:   Counter Preset (0 - 16777215)
     r 101         ; Par. 4:   Register Preset (0 - 16777215)
       0           ; Par. 5:   Enable Counter Konfig. (0 - 3)
       0           ; Par. 6:   CCO Konfigurierung (0 - 3)
       0           ; Par. 7:   Input A Konfigurierung (0 - 3)
       0           ; Par. 8:   Input B Konfigurierung (0, 1)
       0           ; Par. 9:   Messung Konfigurierung (0 - 5)
       0           ; Par. 10:  Messung Wert (0 - 65535)
       0           ; Par. 11:  Enable Messung Konfig. (0 - 3)
       0           ; Par. 12:  TCO Konfigurierung (0 - 3)
```

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)
       RdCt        ; Par. 2:   Befehl: Lesen des aktuellen Wertes
     r 555         ; Par. 3:   Ziel-Register

CFB    exec
     k 1           ; Par. 1:   Modul-Nummer     (k 1 - k 16 )
       LdCtPres    ; Par. 2:   Befehl: Laden des Counters
     r 0           ; Par. 3:   nicht verwendet (leeres Register)
```

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

```
LdCtPres          ; Laden des Counter-Vorwahlwertes
LdRegPres         ; Laden des Vergleichsregisters
ModMsConf         ; Konfigurierung des Messmoduls
LdMsVal           ; Laden des Messbereichs
RdCt              ; Lesen des Wertes im Counter
RdMsImp           ; Lesen des Messresultats
RdMsUnit          ; Lesen des Messres. in Einheiten
StartCt           ; Start des Counters
StartMs           ; Start einer Messung
StopMs            ; Stoppen einer Messung
RdIdent           ; Lesen der Modul-Identifikation
```

Es müssen immer 3 Parameter angegeben werden, auch wenn für einige Funktionen nur 2 verlangt sind. Als 3. Parameter kann 'rNotUsed' oder irgend ein (leeres) Register angegeben werden.

Für den Anwender abfragbare Elemente:

Element	**Adresse**	**Beschreibung**
Cstart_x	8 + BA_x	Wenn Counter läuft, Element = H. ('EnableC' = H und 'Start-Counter' erfolgt)
CCO_x	9 + BA_x	Abbild des Ausgangs 'CCO' (in dynamischen Modi nicht sichtbar, zu kurz))
UpDown_x	10 + BA_x	Wenn Counter up zählt, Element = H.
EnableM_x	12 + BA_x	Abbild des Eingangs 'EnableM'
TCO_x	13 + BA_x	Abbild des Ausgangs 'TCO' (in dynamischen Modi nicht sichtbar, zu kurz)
OverFlow_x	14 + BA_x	Wenn Overflow im Mess-Modus, Element = H.
EndMes_x	15 + BA_x	Wenn Messung beendet, Element = H. Das Element wird am Ende eines Read-Befehls (RdMsImp oder RdMsUnit) zurückgesetzt.
fPar_Err	siehe Resour-cenliste	H, wenn Parameter im FB 'Init' ausserhalb eines erlaubten Bereichs liegen.
fError_x	siehe Resour-cenliste	H, bei einer Division durch Null in der Einheiten-Konvertierung eines Messresultats.
rDiag	siehe Resour-cenliste	Siehe Kapitel 7: Fehlerbehandlung und Diagnose.

'_x' entspricht der Modul-Nummer

Die effektiven Adressen der Elemente sind der Resourcenliste zu entnehmen (für Debug-Zwecke).

6.2 Programmierung in FUPLA mit FBoxen

in Vorbereitung

6.3 Programmierung in GRAFTEC mit FBoxen

in Vorbereitung

7. Fehlerbehandlung und Diagnose

7.1 Definitionsfehler durch Assembler geprüft

Folgende Definitionsfehler in der Datei D2H110_b.MBA werden während dem Assemblieren überprüft:

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

  "Remark: No H110 used (NbrModules = 0 in D2H110_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 H110 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 'H110.RdIdenti'"
  (wobei der Ausdruck 'H110' vom $group h110 generiert wird)

- Falls die Definition $group H110 fehlt, meldet der Assembler:

  "Symbol not defined"

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

7.2 Fehlerbehandlung in RUN

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.

Im FB 'Init' werden alle Parameter auf deren Wert überprüft, ob diese im zulässigen Bereich liegen (z.B. Bereich des Counter-Modus = 0, 1, 2, 3, 4). 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 rsp. im Initial-Step (IST) 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)
```

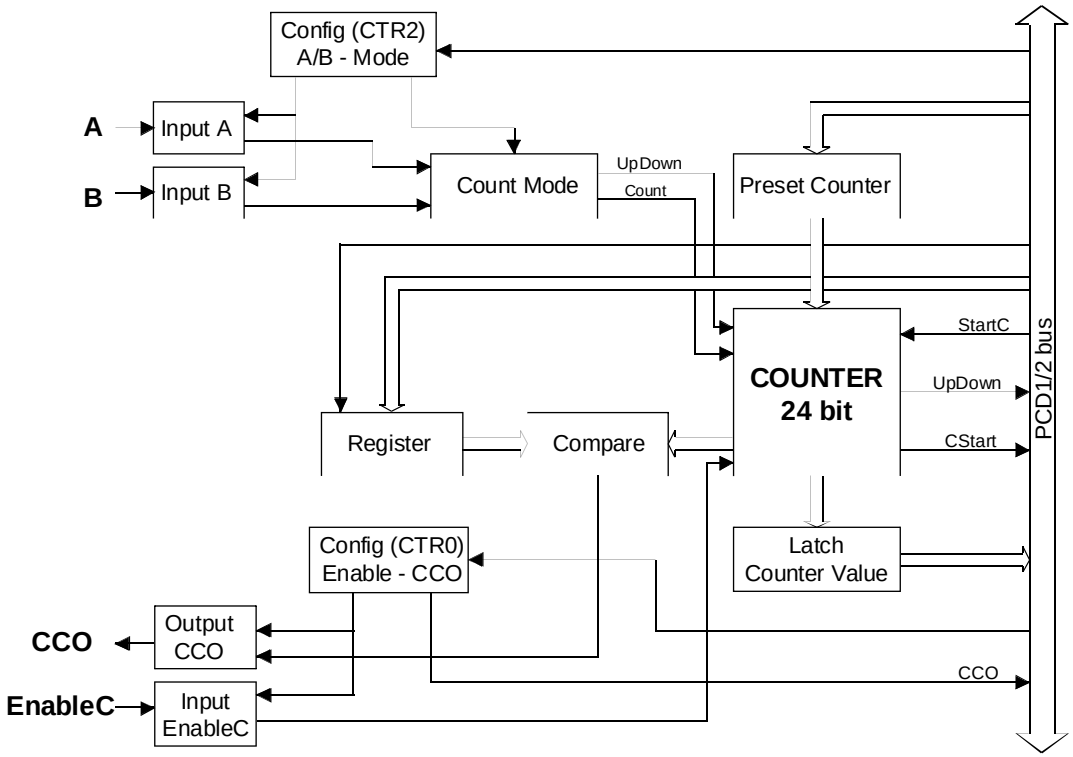
Beispiel: Wird die Konfigurierung des CCO (Parameter 6) im FB 'Init' des Moduls 2 falsch definiert, so wird 'rDiag' mit dem Wert 00 01 06 02 hex geladen.

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. 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. Das PCD2.H110 für Zähl- und Positionieraufgaben

8.1 Blockschaltbild des Zählers



8.2 Beschreibung des Zählers

Der Kern der Schaltung ist der 24-Bit Zähler (Counter). Das Laden des Counters geschieht via den Preset-Counter. Der Preset-Counter selbst wird vom Anwenderprogramm in 3 Paketen zu 8 Bit via ein PCD-Register entweder bei der Konfigurierung oder mit dem Befehl 'LdCtPres' geladen. Mit dem Befehl 'StartCt' wird der Wert des Preset-Counters in den Counter geladen und damit auch der Counter aktiviert.

Das Laden des Registers geschieht in ähnlicher Weise. Es wird vom Anwenderprogramm ein 24-Bit Wert in 3 Paketen via ein PCD-Register in das Register des H110 geladen. Nachdem sowohl die Eingänge 'A' und 'B' sowie der direkte Zählerausgang 'CCO' und der Eingang 'EnableC' konfiguriert wurden (Beschreibung dazu erfolgt etwas später) kann bei Übereinstimmung von Register und Counter der CCO-Ausgang gemäss der Konfigurierung geschaltet und der Prozess sowie das Anwenderprogramm gesteuert werden.

Der Status der Elemente 'CCO' (CCO_x), 'Enable' (CStart_x) und 'Up-Down' (UpDown_x) können im Anwenderporgramm bei Bedarf ausgewertet werden.

8.3 Die Konfigurierung der Eingänge 'EnableC', 'CCO', 'A', 'B' sowie des Zählmodus'

8.3.1 Die Konfigurierung des Eingangs 'EnableC'

Die Konfigurierung des Eingangs '**EnableC**' erlaubt folgende Modi:

Standard: "statisch / normal" → Init-Parameter 5 = 0

Während 'H' am Eingang 'EnableC' liegt, ist die Zählung freigegeben.
Während 'L' am Eingang 'EnableC liegt, ist die Zählung blockiert.

Weitere Möglichkeiten:

"statisch / invertiert" → Init-Parameter 5 = 1

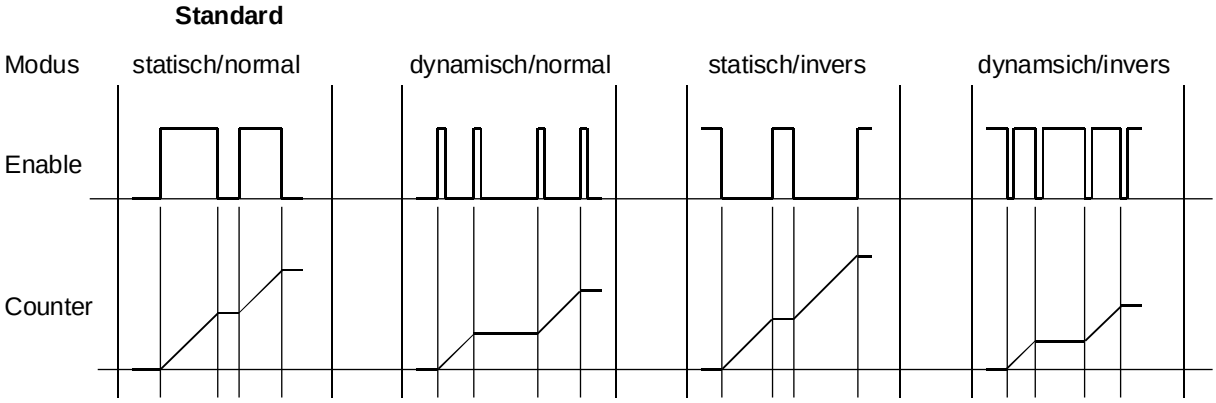
Während "L" am Eingang 'EnableC' liegt, ist die Zählung freigegeben.
Während "H" am Eingang 'EnableC' liegt, ist die Zählung blockiert.

"dynamisch / normal" → Init-Parameter 5 = 2

Es liegt "L" am Eingang 'EnableC'. Eine 1. positive Flanke (H) schaltet den 'EnableC' ein, die nächste wieder aus, usw.

"dynamisch / invertiert" → Init-Parameter 5 = 3

Es liegt "H" am Eingang 'EnableC'. Eine 1. negative Flanke (L) schaltet den 'EnableC' ein, die nächste wieder aus, usw.



8.3.2 Die Konfigurierung des Ausgangs 'CCO'

Die Konfigurierung des Ausgang **'CCO'** erlaubt die folgenden Modi:

"statisch / normal" → Init-Parameter 6 = 0

Der 'CCO' wird vom Anwenderprogramm aktiviert und wird oder bleibt = L. Bei Gleichheit zwischen Register und Counter wird der 'CCO' auf H geschaltet und bleibt = H, bis vom Anwenderprogramm ein neuer Befehl zum Aktivieren kommt.

"statisch / invers" → Init-Parameter 6 = 1

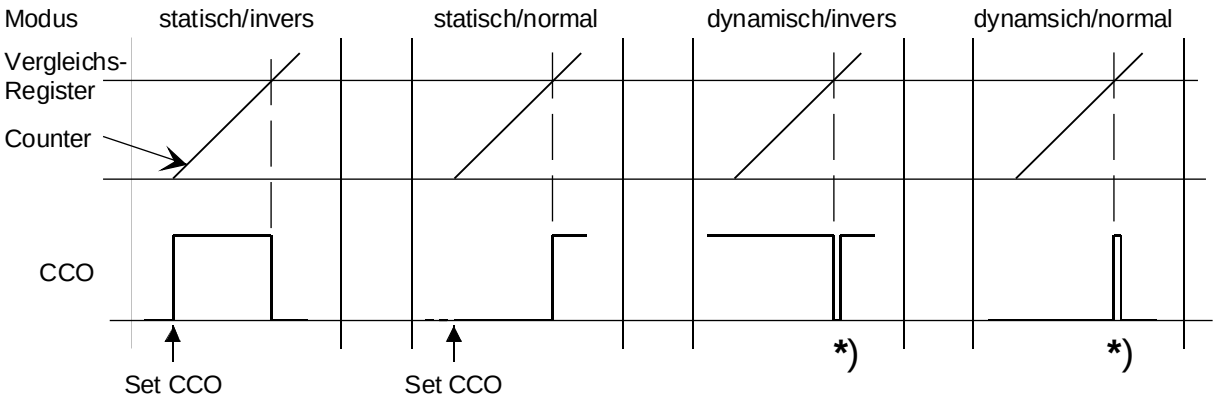
Der 'CCO' wird vom Anwenderprogramm eingeschaltet und wird = H. Bei Gleichheit zwischen Register und Counter wird der 'CCO' auf L geschaltet und bleibt = L, bis vom Anwenderprogramm ein neuer Befehl zum Einschalten kommt.

"dynamisch / normal" → Init-Parameter 6 = 2

Der 'CCO' wird vom Anwenderprogramm aktiviert und wird oder bleibt = L. Bei Gleichheit zwischen Register und Counter wird der CCO für 25 .. 100 µs = H *). Bei jeder fogenden Übereinstimmung von Register und Counter wiederholt sich das Verhalten des 'CCO', ohne dass vom Anwenderprogramm ein neuer Befehl kommt.

"dynamisch / invers" → Init-Parameter 6 = 3

Der 'CCO' wird vom Anwenderprogramm eingeschaltet und wird = H. Bei Gleichheit zwischen Register und Counter wird der 'CCO' für 25 .. 100 µs = L *). Bei jeder fogenden Übereinstimmung von Register und Counter wiederholt sich das Verhalten des 'CCO', ohne dass vom Anwenderprogramm ein neuer Befehl kommt.



*) Die Auswertung dieses kurzen Impulses erfolgt via die Interrupteingänge der PCD1/2 und den XOB 20 bzw. 25.

8.3.3 Die Konfigurierung des Eingänge 'A' und 'B'

Die Eingänge 'A' und 'B' können einzeln invertiert werden.

	Counter	**Messung**	**Parameter 7**
Eingang 'A'	normal	normal	0
Eingang 'A'	invertiert	normal	1
Eingang 'A'	normal	invertiert	2
Eingang 'A'	invertiert	invertiert	3

	Counter		**Parameter 8**
Eingang 'B'	normal	-	0
Eingang 'B'	invertiert	-	1

Das Invertieren eines einzelnen Eingangs ('A' oder 'B') beim Einsatz des Moduls als Counter, ergibt physikalisch eine Umkehr der Drehrichtung eines Antriebs.

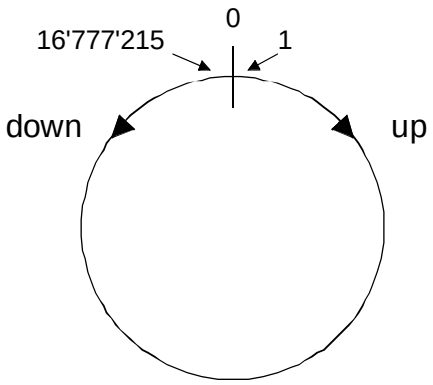
8.3.4 Die Konfigurierung des Zählmodus'

(ohne Verwendung von 2-phasigen Inkrementalgebern)

Modus 'x1' → Init-Parameter 2 = 0 oder 1

Mit dem Modus 'x1' (ohne Inkrementaldrehgeber) werden einfache Zählaufgaben gelöst:

- Die zu zählenden Signale werden an den Eingang 'A' angelegt
- Ist Eingang 'B' = L, wird abwärts (down) gezählt, wenn Par. 8 = 0.
  Ist Eingang 'B' = H, wird aufwärts (up) gezählt, wenn Par. 8 = 0.
  Ist Par. 8 = 1, wird die Zählrichtung invertiert.

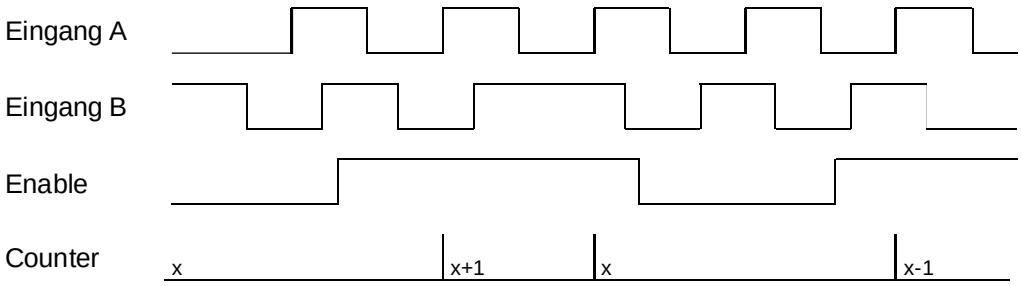


Der Zählbereich ist 0 ... 16'777'215 (0 ... 2^{24} - 1)

Wird von 0 an "up" gezählt → 0, 1, 2 ...
Wird von 0 an "down" gezählt → 0, 16'777'215, 16'777'214 ...

Es gibt keine negativen Werte und keinen Overflow

Modus 'x1' → Init-Parameter 2 = 0 oder 1



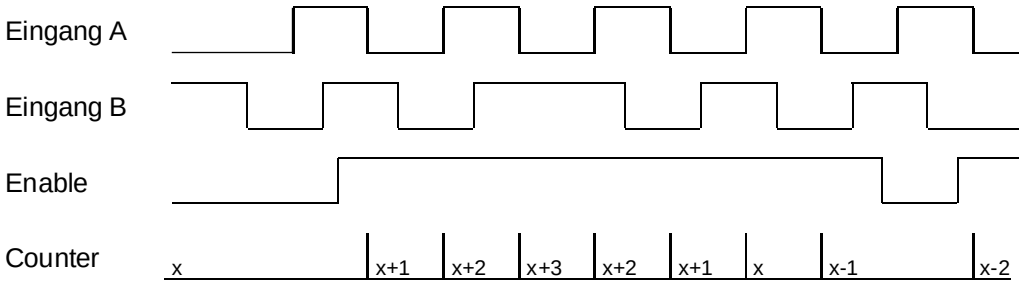
Nur die ansteigende Flanke des Signals 'A' wird ausgewertet. Das um 90° phasenverschobene Signal 'B' definiert die Zählrichtung.

Wichtig: Es dürfen im Modus 'x1' **keine** Inkrementaldrehgeber eingesetzt werden, da in gewissen Situationen Signale richt korrekt ausgewertet werden.

Wird mit 2-phasigen Inkrementaldrehgebern gearbeitet, gelten die folgenden Modi:

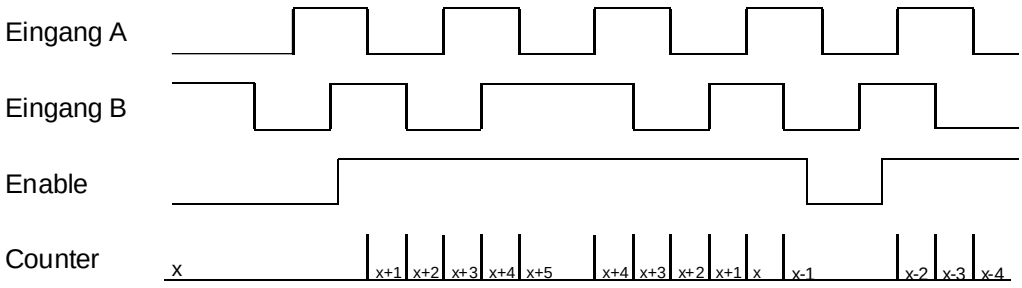
Modus 'x2' → Init-Parameter 2 = 2

Es werden die ansteigenden und die abfallenden Flanken des Sinals 'A' ausgewertet. Das um 90° phasenverschobene Signal 'B' definiert die Zählrichtung.



Modus 'x4' → Init-Parameter 2 = 4

Es werden die ansteigenden und abfallenden Flanken beider Signale 'A' und 'B' ausgewertet. Das um 90° phasenverschobene Signal 'B' definiert auch noch die Zählrichtung.



Ein Modus 'x3' (Init-Parameter 2 = 3) kann auch gewählt werden, hat aber keine praktische Bedeutung und wird nicht beschrieben.

8.4 Programmierbare Funktionen des Counters

- Konfigurierung der Funktionen 'EnableC' und 'CCO'.
- Konfigurierung des Zählmodus'.
- Konfigurierung der Eingänge 'A' und 'B'
- Initialisierung des Zählers, d.h. Übernahme der Definitionen des 'EnableC', des 'CCO' und des Zählmodus.
- Laden des Zählerwertes in das PCD-Register.
- Übertragen des Zählerwertes aus einem PCD-Register in den Preset-Counter.
- Übertragen des Wertes aus dem Preset-Counter in den Counter.
- Laden des Vorwahlwertes in ein PCD-Register.
- Übertragen des Vorwahlwertes aus einem PCD-Register in das Vorwahl-Register.
- Start des Zählers und aktivieren des 'CCO'.

- Lesen das Zählerwertes.
- Lesen des Status des Ausgangs 'CCO' (H = ein, L = aus)
- Lesen des Status des Eingangs 'EnableC' (H = ein, L = aus)
- Lesen der Zählrichtung (H = up, L = down)

Die Programmierung wird auf den folgenden Seiten an einigen Beispielen erläutert.

Anstelle einer mehrseitigen Beschreibung, wird **das Prinzip der Programmierung** an einem unstrukturierten Beispiel gezeigt. Das Programm ist lauffähig und kann, z.B. zum Test eines PCD2.H110, verwendet werden.

Da es sich bei Zählaufgaben immer um sequentielle Programmabläufe handelt:

- Moduleigenschaften und Zählaufgabe definieren
- Ende der Zählung abwarten
- Zählung auswerten

sollen Anwenderprogramme konsequent in GRAFTEC programmiert werden.

Die 3 typischen Beispiele, welche auf das Prinzip-Beispiel folgen, sind deshalb in GRAFTEC editiert.

Die Programme wurden im "PG4" erstellt (GRAFTEC und IL).

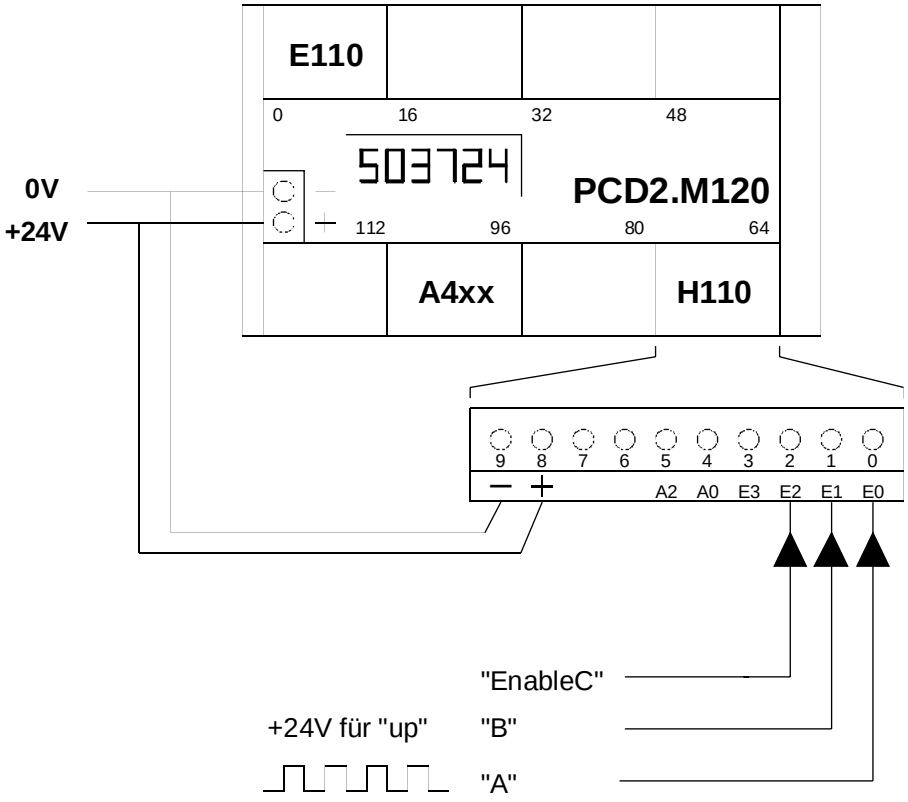
8.5 Prinzip der Programmierung

Es soll an einem einfachen Beispiel die Methodik zur Programmierung des Zählers des PCD2.H110-Moduls gezeigt werden.

Aufgabe: Es soll der Counter auf den Startwert 500 und das Vorwahlregister auf 900 geladen werden. Der CCO-Ausgang soll "statisch-normal" und der Enable-Eingang "statisch-invers" konfiguriert werden. Es sollen der Counter-Modus 'x1' und die Eingänge 'A' und 'B' als "normal" konfiguriert werden. Nach dem Anlegen von Impulsen an den Eingang 'A' soll der Counter bis zum Erreichen der Registerwertes (900) aufwärts zählen. Beim Erreichen des Wertes des Vorwahlregisters soll ein digitaler PCD-Ausgang komplementiert und der Counter wieder auf den Startwert (500) geladen werden usw. Die Anordnung entspricht einem Frequenz-Teiler.

Das Programm hat den Namen "prinzip.src" und liegt im Projekt " d2-h110". In dieses Projekt ist auch die .mba-Datei mit der Anzahl H110-Module (1) und der Basisadresse des H110-Moduls (64) zu kopieren.

Anordnung für die Verwednung des PCD2.H110 als Zähler im gezeigten Beispiel.



Die einzelnen Elemente sind:

PCD1/2.M1x0 mindestens bestückt mit
- 1 PCD2.H110
- 1 PCD2.A4xx
- 1 PCD2.F510/530

```
; *********************************************************
; Basis-Anwenderprogramm für das PCD2.H110-Modul
; als Zähler: prinzip.src
; *********************************************************

$include d2h110_b.equ
$group h110

xob       16

ld      r 999      ; PCD-Register mit
          500      ;  Startwert für Counter
ld      r 998      ; PCD-Register mit
          900      ;  Wert für Vergleichs-Register
ld      r 0        ; Hilfs-Register
          0        ;  leer

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1    Modul-Nummer
          1        ;  Par 2    Zähl-Modus 'x1'
        r 999      ;  Par 3    Startwert für Counter
        r 998      ;  Par 4    Wert für Vergleichs-Register
          1        ;  Par 5    EnableC "statisch-invers"
          0        ;  Par 6    CCO "statisch-normal"
          0        ;  Par 7    Input A "normal"
          0        ;  Par 8    Input B "normal"
          0        ;  Par 9    nicht verwendet für Counter
          0        ;  Par 10   nicht verwendet für Counter
          0        ;  Par 11   nicht verwendet für Counter
          0        ;  Par 12   nicht verwendet für Counter

cfb       exec     ; Start Counter, setze CCO
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

exob
; ---------------------------------------------------------

cob       0        ; eigentliches Anwender-Programm
          0

cfb       exec     ; Read Counter
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert -> R 777

dsp     r 777      ; Anzeige an Displaymodul

sth       cco_1    ; Abfrage CCO
cpb     h 25       ; wenn CCO = H, PB 25 aufrufen

ecob
; ---------------------------------------------------------
```

```
pb        25       ; PB für Neustart des Counters

com     o 100      ; invertiert bei jedem R = C

cfb       exec     ; Lade Counter mit Vorwahlwert
        k 1        ;  Modul-Nummer
          LdCtPres ;  Befehl: Laden des Vorwahlwertes
        r 999      ;  mit Wert aus Register 999

cfb       exec     ; Start Counter, setze CCO
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

epb

$endgroup
; ------------------------------------------------------------
```

Beschreibung des Programms

Am Anfang des Anwenderprogramms ist die Datei 'd2h110_b.equ' mit '$include' einzubinden. Mit der Assembler-Direktive '$group' wird der nachfolgende Programmteil bis zu '$endgroup' als Programmteil für das PCD2.H110.Modul deklariert. (Da die anderen H-Module die gleichen Befehle verwenden, wird auf diese Art der Modultyp beim Assemblieren des Anwender-Programms unterschieden).

In der Kaltstart-Routine 'XOB 16' wird als erstes ein PCD-Register mit dem Startwert des Counters geladen (z.B. R 999). Ist der Startwert = 0, wird dieses PCD-Register mit 0 geladen. Das nächste PCD-Register wird mit dem Wert des H110 Vergleichs-Registers geladen (z.B. R 998). Dieses Register muss auch dann definiert werden, wenn das H100-Register in der Anwendung nicht verwendet wird. Es kann noch ein leeres Register (z.B. R 0) für Hilfsfunktionen vorbereitet werden.

Die eigentliche Konfigurierung des PCD2.H110-Moduls geschieht mit dem Aufruf des FB 'Init'. Diesem FB werden 12 Parameter mitgegeben, deren Bedeutung im Programm selbst und im "Anhang A" dieses Handbuchs aufgelistet ist. Es wird für Parameter 5: EnableC "statisch-invers" gewählt. Die Klemme 2 braucht dann nicht an '+' gelegt zu werden. Der Parameter 6: 'CCO' wird als "statisch-normal" definiert. Bei Gleichheit von Counter und Register wird der CCO-Ausgang = H.

Nach der Abarbeitung des FB 'Init' wird der Counter mit dem Aufruf des FB 'Exec' mit dem Parameter 2 als 'StartCt' gestartet und der CCO aktiviert.

Im Hauptteil des Anwenderprogramms, dem COB, wird in diesem Beispiel der Counterwert dauernd ausgelesen und dem Display zugeführt sowie der CCO softwaremässig abgefragt. Wird der CCO = H, d.h. "Counter = Register", wird der PB 25 aufgerufen und darin der digitale PCD-Ausgang komplementiert sowie der Counter wieder mit dem Initialwert geladen, neu gestartet und der CCO wieder neu aktiviert.

Das Programm kann nun mit 'Project' - 'Build' verarbeitet, in die PCD geladen und in 'Run' gebracht werden. Die Funktion der Anordnung kann am Display und am angesteuerten digitalen PCD-Ausgang verfolgt werden.

Es wäre auch denkbar, den CCO dynamisch zu betreiben, d.h. dass der CCO bei jedem "Counter = Register" für ca. 100 µs = H würde. Dieser Impuls könnte dann zum 'INB1'-Interrupt-Eingang geführt werden. Bei jedem Impuls (Counter = Register) würde danach ein XOB 20 aufgerufen, worin der Counter wieder neu geladen und gestartet werden könnte.

Diese dynamische Methode sollte nur von Spezialisten in Erwägung gezogen werden, da sich Konflikte zwischen den H110-FBs im COB und im XOB 20 ergeben können.

Andere (realistischere) Anwendungen werden in den nachfolgenden Beispielen gezeigt.

8.6 Anwendungsbeispiel Nr. 1: Zähler in GRAFTEC

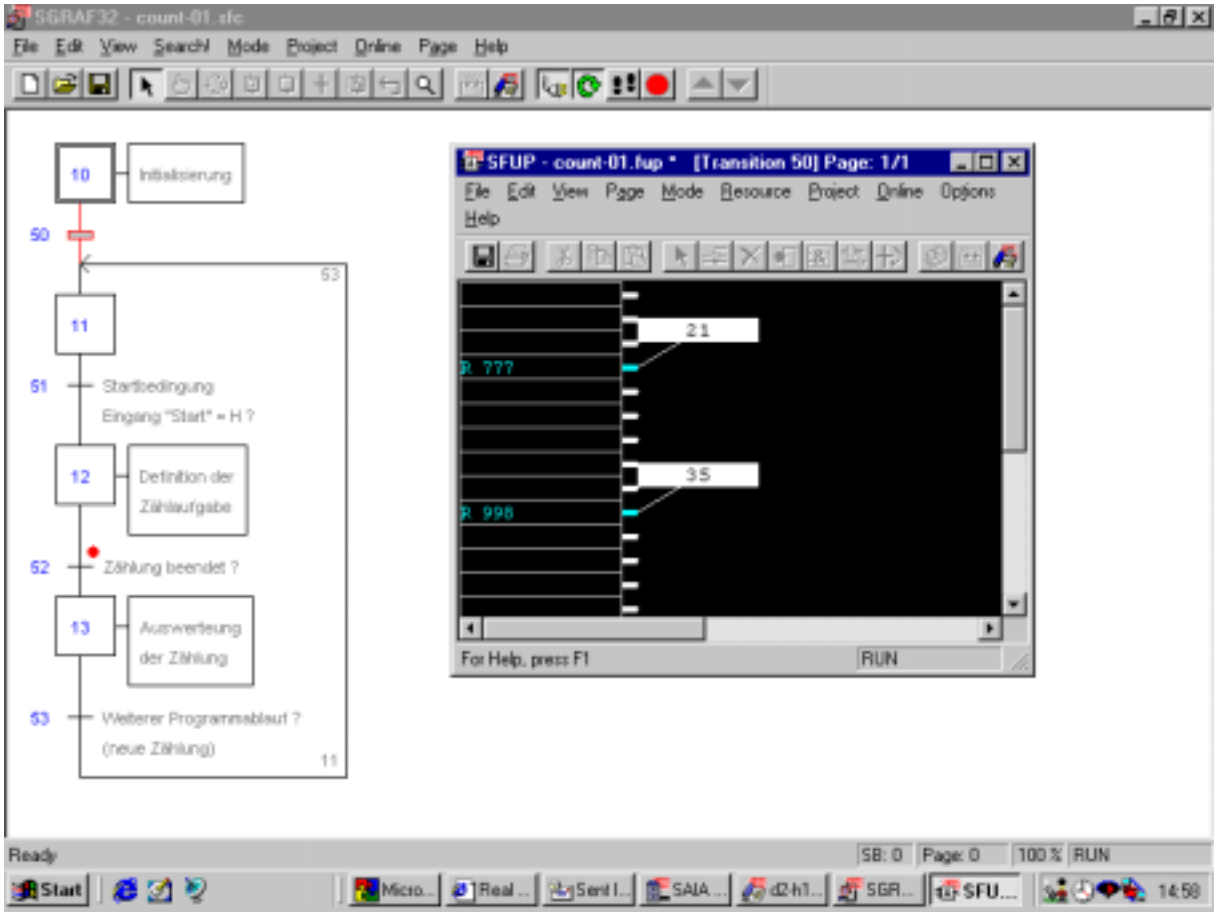
Nach dem Einschalten des Eingangs "Start" soll der Counter auf 0 und das Register auf einen Wert, welcher am 2-stelligen BCD-Schalter, welcher an die Eingänge 16 bis 23 verdrahtet ist, geladen werden.

Der CCO-Ausgang ist so zu definieren, dass dieser am Anfang jeder neuen Zählung = H geschaltet wird. Erreicht der Counter den vorgegebenen Registerwert, soll der CCO = L werden und = L bleiben, bis eine neue Zählung gestartet wird.

Der Zählerstand soll an jeder Stelle des Programms im Displayfenster angezeigt werden. Der Ablauf des Programms sowie die Werte des Registers und des Counters sollen zusätzlich online am Bildschirm des Programmiergerätes sichtbar sein.

Das Anwenderprogramm wird in GRAFTEC editiert. Das Programm erhält innerhalb des Projekts "D2-H110" den Namen "COUNT-01.SFC".

Das fertige Programm zeigt sich etwa folgendermassen:



Der Code des Programms "count-01.sfc"

(Um diese Darstellung zu erhalten, ist die Datei "count-01.sfc" in "count-01.src" umzubenennen).

```
SB        0

;-------------------------------
IST       10       ;Initialisierung
    O 50
$include d2h110_b.equ
$group h110

ld      r 999      ; Startwert für Counter
          0
ld      r 998      ; Wert für Vergleichs-Register
          0
ld      r 0        ; leeres Hilfsregister
          0

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1  Modul-Nummer
          1        ;  Par 2  Zähl-Modus 'x1'
        r 999      ;  Par 3  Startwert für Counter
        r 999      ;  Par 4  Wert für Vergleichs-Register
          1        ;  Par 5  Enable "statisch-invers"
          1        ;  Par 6  CCO "statisch-invers"
          0        ;  Par 7  Input A "normal"
          0        ;  Par 8  Input B "normal"
          0        ;  Par 9  nicht verwendet
          0        ;  Par 10 nicht verwendet
          0        ;  Par 11 nicht verwendet
          0        ;  Par 12 nicht verwendet

EST               ; 10

;-------------------------------
ST        11
    I 50
    I 53          ; Weiterer Programmablauf ?
    O 51          ; Startbedingung: Eingang "Start" = H ?

EST               ; 11
```

```
;-------------------------------
ST        12       ; Definition der Zählaufgabe
    I 51           ; Startbedingung: Eingang "Start" = H ?
    O 52           ; Zählung beendet ?

digir     2        ; Lesen BCD-Wert
        i 16
        r 998

cfb       exec     ; Laden des Vergleichs-Registers
        k 1        ;  Modul-Nummer
          LdRegPres;  Befehl: Laden des Vergleichs-Reg.
        r 998      ;  Reg. mit Ladewert

cfb       exec     ; Laden des Counters
        k 1        ;  Modul-Nummer
          LdCtPres ;  Befehl: Laden des Counters
        r 998      ;  Reg. mit Ladewert

cfb       exec     ; Start des Counters, aktivieren des CCO
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

EST                ; 12

;-------------------------------
ST        13       ; Auswertung der Zählung
    I 52           ; Zählung beendet ?
    O 53           ; Weiterer Programmablauf ?

com     o 101      ; stellvertretend für Prozess

EST                ; 13

;-------------------------------
TR        50
    I 10           ; Initialisierung
    O 11

;; SFUP
__TR00050

ETR                ; 50

;-------------------------------
TR        51       ; Startbedingung: Eingang "Start" = H ?
    I 11
    O 12           ; Definition der Zählaufgabe

cfb       exec     ; Read Counter
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert in R 777
dsp     r 777      ; Anzeige an Displaymodul
sth     i 0        ; PCD-Eingang "Start"

ETR                ; 51
```

```
;-------------------------------
TR        52       ; Zählung beendet ?
    I 12           ; Definition der Zählaufgabe
    O 13           ; Auswerteung der Zählung

cfb       exec     ; Read Counter
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert in R 777
dsp     r 777      ; Anzeige an Displaymodul
stl       cco_1    ; Abfrage des CCO

 ETR               ; 52

;-------------------------------
TR        53       ; Weiterer Programmablauf ?
    I 13           ; Auswerteung der Zählung
    O 11

cfb       exec     ; Read Counter
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert in R 777
dsp     r 777      ; Anzeige an Displaymodul
stl     i 0        ; PCD-Eingang "Start"

$endgroup

ETR                ; 53

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 H110-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 H110-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.

In der normalerweise leeren TR 50 können im FUPLA z.B. die beiden Werte für den aktuellen Stand des Counters und der BCD-Vorwahlwert editiert und danach online eingesehen werden. Der Code wird vom PG4 automatisch erzeugt und darf nicht verändert werden.

Im ST 12 werden der aktuelle BCD-Wert ins Vorwahlregister des H110 geladen, der Counter zurück gestellt (mit Null geladen), der Counter gestartet und damit auch der CCO aktiviert.

In TR 52 wird softwaremässig das Ende der Zählung abgefragt, um den weiteren Programmablauf frei zu geben. Der Prozess selbst wird mit dem hardwarmässigen CCO-Ausgang direkt gesteuert. Vor der Abfrage der Weiterschaltbedingung (stl cco_1) wird der Counterinhalt gelesen und zum Display gebracht. Dies gilt auch für TR 51 und TR 53.

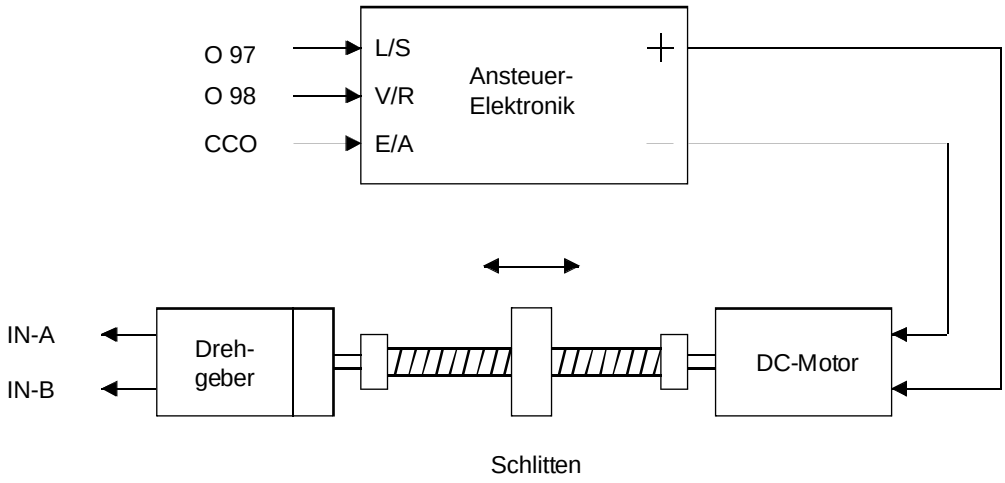
8.7 Anwendungsbeispiel Nr. 2: Positionierung mit Inkrementaldrehgeber

Der Schlitten eines Laufmodells (Gleichstrommotor, Spindel, Schlitten, Inkrementaldrehgeber und dazugehöriger Ansteuerelektronik) ist aus einer Grundstellung in eine andere Position und, nach einer Pause, wieder in die Ausgangstellung zurück zu fahren. Es soll mit der vollen Geschwindigkeit bis zu einem Vorabschaltwert und von da an, mit reduzierter Geschwindigkeit, bis zur Zielposition gefahren werden.

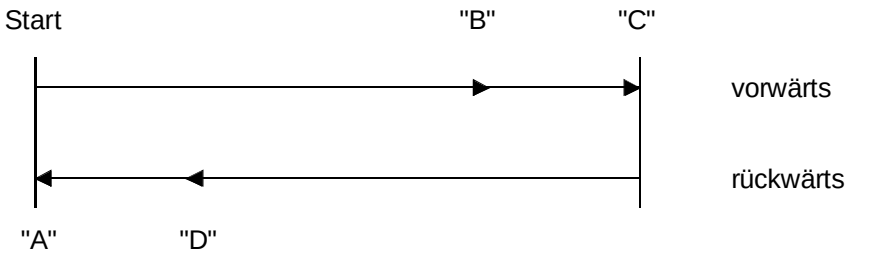
Einige Daten des Laufmodells (V-PCX 10):

Gleichstrommotor mit Getriebe:	ca. 1200 U/min. bei 24 VDC
Inkrementaldrehgeber:	500 Impulse/U, 2-phasig, 90° verschoben
Steigung der Spindel	1.0 mm
Eingänge der Elektronik: (V-PCX 11)	vorwärts/rückwärts (V/R): H = vorwärts; L = rückwärts langsam/schnell (L/S): L = langsam; H = schnell ein/aus (E/A) L = Motor aus, kurzgeschlossen H = Motor ein, entsprechend vorwärts/rückwärts, langsam/schnell
Ausgänge der Elektronik:	Motor (Polung beachten)
Speisung der Elektronik:	24 VDC geglättet

Anordnung und Verdrahtung der Geräte:

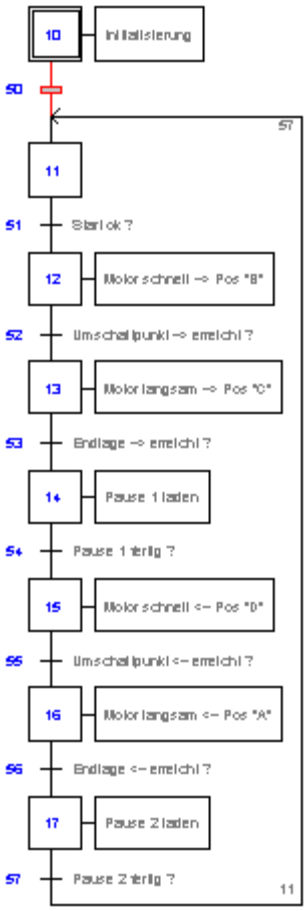


Bewegung:



0 cm	10 mm	90 mm	100 mm	Weg
0	10'000	90'000	100'000	Impulse (Modus x2)

Die GRAFTEC-Struktur sieht etwa folgendermassen aus:



Der Code des Programms "move-01.sfc"

(Um diese Darstellung zu erhalten, ist die Datei "move-01.sfc" in "move-01.src" umzubenennen).

```
SB        0

;-------------------------------
IST       10       ;Initialisierung
    O 50
$include d2h110_b.equ
$group h110

ld      r 999      ; Startwert für Counter
          0
ld      r 998      ; Wert für Vergleichs-Register
          0
ld      r 0        ; leeres Hilfsregister
          0
ld      r 995      ; Hilfsregister für negative Werte
          16000000
ld      r 996      ; Hilfsregister für negative Werte
          777215

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1  Modul-Nummer
          2        ;  Par 2  Zähl-Modus "x2"
        r 999      ;  Par 3  Startwert für Counter
        r 998      ;  Par 4  Wert für Vergleichs-Register
          1        ;  Par 5  Enable "statisch-invers"
          1        ;  Par 6  CCO "statisch-invers"
          0        ;  Par 7  Input A "normal"
          0        ;  Par 8  Input B "normal"
          0        ;  Par 9  nicht verwendet
          0        ;  Par 10 nicht verwendet
          0        ;  Par 11 nicht verwendet
          0        ;  Par 12 nicht verwendet
EST                ; 10

;-------------------------------
ST        11
    I 50
    I 57           ; Pause 2 fertig ?
    O 51           ; Start ok ?
EST                ; 11

;-------------------------------
ST        12       ; Motor schnell --> Pos "B"
    I 51           ; Start ok ?
    O 52           ; Umschaltpunkt --> erreicht ?

ld      r 998      ; Laden Wert für "B"
          8640
cfb       exec     ; Laden des Vergleichs-Registers
        k 1        ;  Modul-Nummer
          LdRegPres;  Befehl: Laden des Vergleichs-Reg.
        r 998      ;  PCD-Register mit Vergleichswert
cfb       exec     ; Start des Counters, aktivieren des CCO
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

set     o 97       ; Motor "schnell"
set     o 98       ; Motor "vorwärts"

EST                ; 12
```

```
;--------------------------------
ST        13       ; Motor langsam --> Pos "C"
    I 52           ; Umschaltpunkt --> erreicht ?
    O 53           ; Endlage --> erreicht ?

ld      r 998      ; Laden Wert für "C"
          9600

cfb       exec     ; Laden des Vergleichs-Registers
        k 1        ;  Modul-Nummer
          LdRegPres;  Befehl: Laden des Vergleichs-Reg.
        r 998      ;  PCD-Register mit Vergleichswert

cfb       exec     ; Start des Counters, aktivieren des CCO
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

res     o 97       ; Motor "langsam"
set     o 98       ; Motor "vorwärts"

EST                ; 13

;--------------------------------
ST        14       ;Pause 1 laden
    I 53           ;Endlage --> erreicht ?
    O 54           ;Pause 1 fertig ?

ld      t 0        ; Pause 1
          50       ;  5 Sekunden

EST                ; 14

;--------------------------------
ST        15       ; Motor schnell <-- Pos "D"
    I 54           ; Pause 1 fertig ?
    O 55           ; Umschaltpunkt <-- erreicht ?

ld      r 998      ; Laden Wert für "C"
          960

cfb       exec     ; Laden des Vergleichs-Registers
        k 1        ;  Modul-Nummer
          LdRegPres;  Befehl: Laden des Vergleichs-Reg.
        r 998      ;  PCD-Register mit Vergleichswert

cfb       exec     ; Start des Counters, aktivieren des CCO
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

set     o 97       ; Motor "schnell"
res     o 98       ; Motor "rückwärts"

EST                ; 15
```

```
        ;--------------------------------
        ST        16       ; Motor langsam <-- Pos "A"
            I 55           ; Umschaltpunkt <-- erreicht ?
            O 56           ; Endlage <-- erreicht ?

        ld      r 998      ; Laden Wert für "A"
                  0

        cfb       exec     ; Laden des Vergleichs-Registers
                k 1        ;  Modul-Nummer
                  LdRegPres;  Befehl: Laden des Vergleichs-Reg.
                r 998      ;  PCD-Register mit Vergleichswert

        cfb       exec     ; Start des Counters, aktivieren des CCO
                k 1        ;  Modul-Nummer
                  StartCt  ;  Befehl: Start Counter
                r 0        ;  leeres Register

        res     o 97       ; Motor "langsam"
        res     o 98       ; Motor "rückwärts"

        EST                ; 16

        ;--------------------------------
        ST        17       ; Pause 2 laden
            I 56           ; Endlage <-- erreicht ?
            O 57           ; Pause 2 fertig ?

        ld      t 0        ; Pause 1
                  50       ;  5 Sekunden

        EST                ; 17

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

        ;--------------------------------
        TR        51       ; Start ok ?
            I 11
            O 12           ; Motor schnell --> Pos "B"

        cfb       exec     ; Read Counter
                k 1        ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                r 777      ;  gelesener Wert in R 777

        cmp     r 777      ; ¦
                r 995      ; ¦ Test, ob Counterwert
        jr      n next     ; ¦
        sub     r 777      ; ¦   kleiner als Null.
                r 995      ; ¦
                r 777      ; ¦   Wenn ja, Anzeige
        sub     r 777      ; ¦
                r 996      ; ¦ des negativen Wertes.
                r 777      ; ¦

next:   dsp     r 777      ; Anzeige an Displaymodul

        sth     i 0        ; PCD-Eingang "Start"

        ETR                ; 51
```

```
        ;-------------------------------
        TR        52       ; Umschaltpunkt --> erreicht ?
            I 12           ; Motor schnell --> Pos "B"
            O 13           ; Motor langsam --> Pos "C"

        cfb       exec     ; Read Counter
                k 1        ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                r 777      ;  gelesener Wert in R 777

        cmp     r 777      ; ¦
                r 995      ; ¦ Test, ob Counterwert
        jr      n next     ; ¦
        sub     r 777      ; ¦   kleiner als Null.
                r 995      ; ¦
                r 777      ; ¦   Wenn ja, Anzeige
        sub     r 777      ; ¦
                r 996      ; ¦ des negativen Wertes.
                r 777      ; ¦

next:   dsp     r 777      ; Anzeige an Displaymodul

        stl       cco_1    ; Abfrage CCO

        ETR                ; 52

        ;-------------------------------
        TR        53       ; Endlage --> erreicht ?
            I 13           ; Motor langsam --> Pos "C"
            O 14           ; Pause 1 laden

        cfb       exec     ; Read Counter
                  k 1      ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                  r 777    ;  gelesener Wert in R 777

        dsp       r 777    ; Anzeige an Displaymodul

        stl       cco_1    ; Abfrage CCO

        ETR                ; 53

        ;-------------------------------
        TR        54       ; Pause 1 fertig ?
            I 14           ; Pause 1 laden
            O 15           ; Motor schnell <-- Pos "D"

        cfb       exec     ; Read Counter
                k 1        ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                r 777      ;  gelesener Wert in R 777

        dsp     r 777      ; Anzeige an Displaymodul

        stl     t 0        ; Abfrage Timer

        ETR                ; 54
```

```
        ;-------------------------------
        TR        55       ; Umschaltpunkt <-- erreicht ?
            I 15           ; Motor schnell <-- Pos "D"
            O 16           ; Motor langsam <-- Pos "A"

        cfb       exec     ; Read Counter
                k 1        ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                r 777      ;  gelesener Wert in R 777

        dsp     r 777      ; Anzeige an Displaymodul
        stl       cco_1    ; Abfrage CCO
        ETR                ; 55

        ;-------------------------------
        TR        56       ; Endlage <-- erreicht ?
            I 16           ; Motor langsam <-- Pos "A"
            O 17           ; Pause 2 laden

        cfb       exec     ; Read Counter
                k 1        ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                r 777      ;  gelesener Wert in R 777

        cmp     r 777      ; ¦
                r 995      ; ¦ Test, ob Counterwert
        jr      n next     ; ¦
        sub     r 777      ; ¦   kleiner als Null.
                r 995      ; ¦
                r 777      ; ¦   Wenn ja, Anzeige
        sub     r 777      ; ¦
                r 996      ; ¦ des negativen Wertes.
                r 777      ; ¦

next:   dsp     r 777      ; Anzeige an Displaymodul
        stl       cco_1    ; Abfrage CCO

        ETR                ; 56

        ;-------------------------------
        TR        57       ; Pause 2 fertig ?
            I 17           ; Pause 2 laden
            O 11

        cfb       exec     ; Read Counter
                k 1        ;  Modul-Nummer
                  RdCt     ;  Befehl: Read Counter
                r 777      ;  gelesener Wert in R 777

        cmp     r 777      ; ¦
                r 995      ; ¦ Test, ob Counterwert
        jr      n next     ; ¦
        sub     r 777      ; ¦   kleiner als Null.
                r 995      ; ¦
                r 777      ; ¦   Wenn ja, Anzeige
        sub     r 777      ; ¦
                r 996      ; ¦ des negativen Wertes.
                r 777      ; ¦

next:   dsp     r 777      ; Anzeige an Displaymodul
        stl     t 0        ; Abfrage Timer

        $endgroup
        ETR                ; 57

        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 H110-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 H110-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.

Der Counter wird nur beim Einschalten, d.h. im IST mit Null geladen und danach im Programm nicht mehr beeinflusst. Danach werden alle vom Inkrementaldrehgeber kommenden Signale gezählt. Im Counter liegt also immer das genaue Abbild der Schlittenposition, d.h. dass auch Überläufe der Endpositionen oder das Drehen der Spindel von Hand genau erfasst werden. Der Befehl 'Start Counter' in den ST 12, 13, 15 und 16 dient jeweils der Aktivierung des CCO. Das Vergleichs-Register im H110 wird vor jeder neuen Schlittenbewegung mit der entprechenden nächsten Position geladen.

Die softwaremässige Abfrage des CCO in den TR 52, 53, 55 und 56 dient nur der Weiterschaltung im GRAFTEC. Die Steuerung des Prozesses selbst, im vorliegenden Fall der Steuerung des Motors des Schlittens, geschieht direkt via den CCO-Ausgang.

Die Programmteile "Test, ob Counterwert kleiner als Null..." in den meisten TR dienen dazu, die Anzeige für Werte kleiner Null als negative Werte anzuzeigen. Diese Programmteile haben mit dem PCD2.H110 selbst nichts zu tun. Es könnte auch die Nullposition bei z.B. 1000 definiert werden (Offset). So könnte die Problematik der Werte "unter Null" umgangen werden.

Der geneigte Leser wird vielleicht feststellen, dass das Beispiel nicht ganz sauber ausprogrammiert ist. Beim Umschalten von "schnell" auf "langsam" wird der Motor kurzzeitig ausgeschaltet bzw. von der Ansteuerung kurzgeschlosen, da der CCO-Ausgang beim Erreichen der Positionen "B" bzw. "D" ausgeschaltet und nach dem Laden des neuen Wertes wieder eingeschaltet wird. Richtigerweise müsste diese Umschaltung mit einem normalen Ausgang überbrückt werden. Es geht hier aber nur ums Prinzip und dieses dürfte ersichtlich sein.

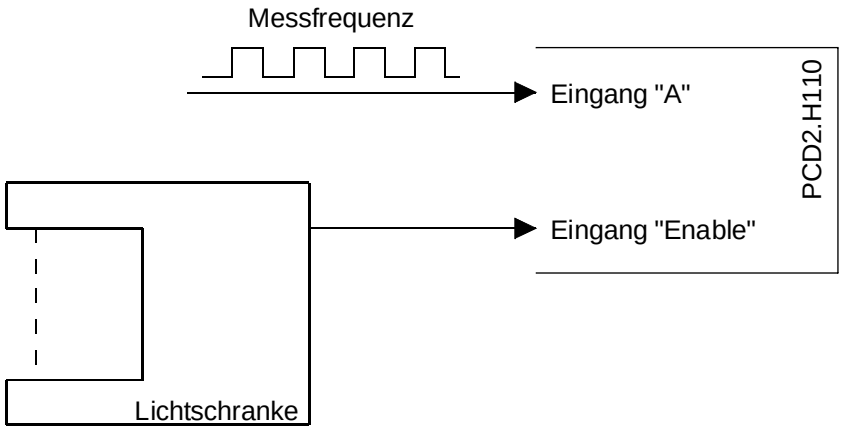
Im Abschnitt 9.1.5 wird gezeigt, wie parallel zu der eben gezeigten Positionierung auch noch die Frequenz (Drehzahl) gemessen werden kann.

8.8 Anwendungsbeispiel Nr. 3: Messen mittels Zählung

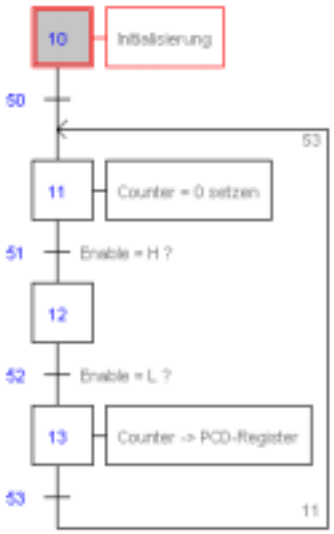
Während dem eine Lichtschranke durch Stücke, welche auf einem Förderband transportiert werden, abgedeckt wird, sollen Impulse, welche proportional zur Geschwindigkeit des Förderbandes ausgegeben werden, gezählt und so die Grösse der Stücke, zwecks Aussortierung gemessen werden. Diese Methode wird z.B. in Südfrankreich zur Sortierung von Melonen und Aprikosen seit Jahren mit grossem Erfolg angewandt.

Es geht also darum, Signale während einer bestimmten Situation, z.B. Lichtschranke abgedeckt, zu zählen, wobei die Zählung direkt durch die Lichtschranke über einen Eingang des Zählmoduls und **nicht** via einen digitalen PCD-Eingang gesteuert wird.

Es werden zu diesem Zweck die Messimpulse an den Eingang "A" und die Lichtschranke an den Eingang "Enable" des Zählmoduls geführt, womit die Aufgabe bereits weitgehend gelöst ist.



Die GRAFTEC-Struktur sieht etwa folgendermassen aus:



Der Code des Programms "mess-01.sfc"

(Um diese Darstellung zu erhalten, ist die Datei "mess-01.sfc" in "mess-01.src" umzubenennen).

```
SB        0

;------------------------------
IST       10       ; Initialisierung
    O 50

$include d2h110_b.equ
$group h110

ld      r 999      ; Startwert für Counter
          0
ld      r 998      ; Wert für Vergleichs-Register
          0
ld      r 0        ; leeres Hilfsregister
          0

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1  Modul-Nummer
          0        ;  Par 2  Zähl-Modus 'x1'
        r 999      ;  Par 3  Startwert für Counter
        r 998      ;  Par 4  Wert für Register (nicht verw.)
          1        ;  Par 5  Enable "statisch-invers"
          0        ;  Par 6  CCO (nicht verwendet)
          0        ;  Par 7  Input A "normal"
          1        ;  Par 8  Input B "invers"
          0        ;  Par 9  nicht verwendet
          0        ;  Par 10 nicht verwendet
          0        ;  Par 11 nicht verwendet
          0        ;  Par 12 nicht verwendet

 EST               ; 10

;------------------------------
ST        11       ; Counter = 0 setzen
    I 50
    I 53
    O 51           ; Enable = H ?

cfb       exec     ; Laden des Counters (mit 0)
        k 1        ;  Modul-Nummer
          LdCtPres ;  Befehl: Laden des Counters
        r 998      ;  PCD-Register mit Ladewert

cfb       exec     ; Start des Counters
        k 1        ;  Modul-Nummer
          StartCt  ;  Befehl: Start Counter
        r 0        ;  leeres Register

EST                ; 11

;------------------------------
ST        12
    I 51           ; Enable = H ?
    O 52           ; Enable = L ?
EST                ; 12
```

```
;-------------------------------
ST        13       ; Counter -> PCD-Register
    I 52           ; Enable = L ?
    O 53

cfb       exec     ; Read Counter (für Resultat)
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert in R 777

putx    r 777
        r 2000
ini     k 1000

EST                ; 13

;-------------------------------
TR        50
    I 10           ; Initialisierung
    O 11           ; Counter = 0 setzen
ETR                ; 50

;-------------------------------
TR        51       ; Enable = H ?
    I 11           ; Counter = 0 setzen
    O 12

cfb       exec     ; Read Counter (für Display)
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert in R 777

dsp     r 777      ; Anzeige an Displaymodul

sth       Cstart_1 ; Abfrage 'EnableC'

ETR                ; 51

;-------------------------------
TR        52       ; Enable = L ?
    I 12
    O 13           ; Counter -> PCD-Register

cfb       exec     ; Read Counter (für Display)
        k 1        ;  Modul-Nummer
          RdCt     ;  Befehl: Read Counter
        r 777      ;  gelesener Wert in R 777

dsp     r 777      ; Anzeige an Displaymodul

stl       Cstart_1 ; Abfrage 'EnableC'

ETR                ; 52

;-------------------------------
TR        53
    I 13           ; Counter -> PCD-Register
    O 11           ; Counter = 0 setzen

$endgroup

ETR                ; 53

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.

Zur Definition des 'Enable' und des CCO: Die in diesem Beispiel eingesetzte Lichtschranke liefert an den Enable-Eingang ein H-Signal, wenn diese nicht abgedeckt ist. Das Signal ist als "statisch-invers" zu definieren. Der CCO wird nicht verwendet und wird mit irgend einem gültigen Code (0 - 3) definiert.

Der Zählmodus ist 'up/down' was dem Modus 'x1' entspricht. Der Code für Eingang 'A' ist 0 (normal). Wird der Eingang 'B' mit Code '0' definiert, muss +24V angelegt werden, um "up" zu zählen. Wird Code '1' definiert (invertiert) braucht die Klemme nicht verdrahtet zu werden um "up" zu zählen.

Die Steuerung des Programms erfolgt durch die softwaremässige Abfrage des Eingangs 'EnableC' mit "STH oder STL Cstart_1". Die Steuerung des Counters erfolgt direkt über den Eingang 'Enable'. (Für die statischen Modi des 'EnableC' ist das Element 'Cstart_x' das Abbild des Eingangs 'EnableC').

In den TR 51 und 52 wird der Counter nur zu Displayzwecken ausgelesen. Im ST 13, nach beendeter Messung, wird das Messresultat aus dem Counter gelesen und weiter verarbeitet.

Im ST 13 ist noch eine Auswertung des Resultates angedeutet. Es wird jeweils der Wert im Counter in fortlaufende PCD-Register abgelegt (ab R 2000). Diese Resultate können im Debugger eingesehen werden.

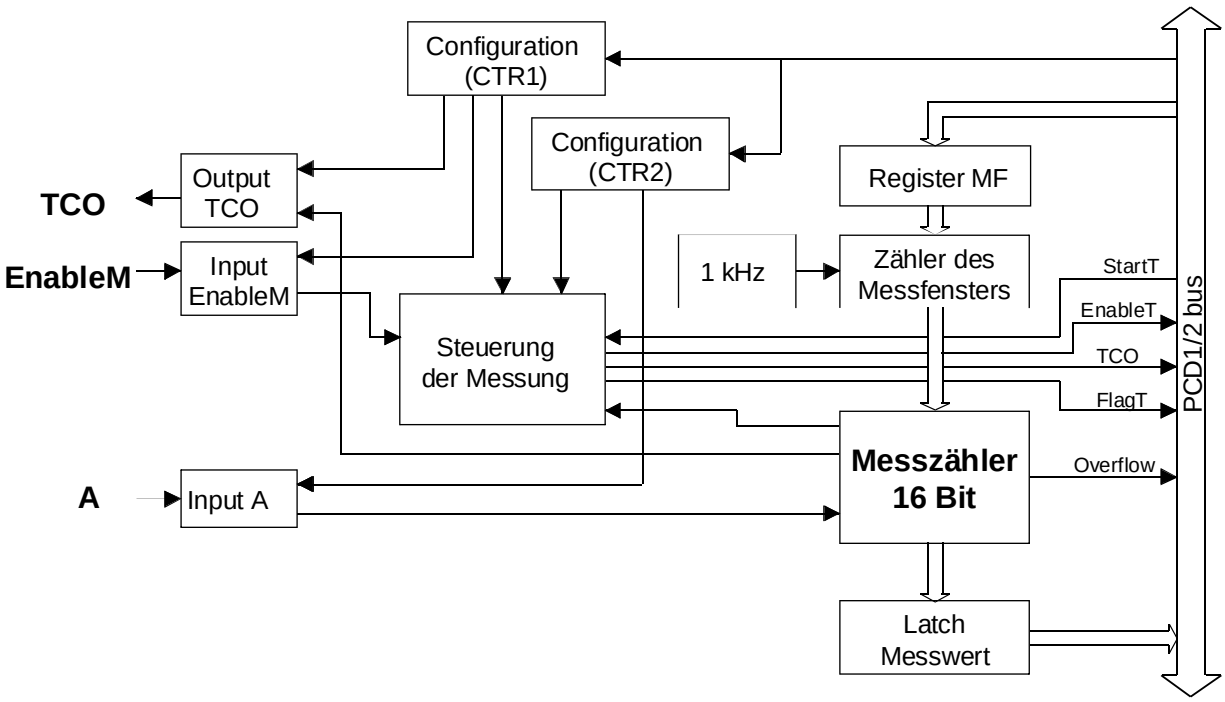
Anstelle einer Lichtschranke kann auch ein prellfreier Schalter eingesetzt werden. Als Ersatz für die von einem Förderband abgegebenen Mess-Impulse wird ein Frequenzgenerator verwendet.

Notizen

9. Das PCD2.H110 für Messaufgaben

9.1 Frequenzmessung

9.1.1 Blockschaltbild



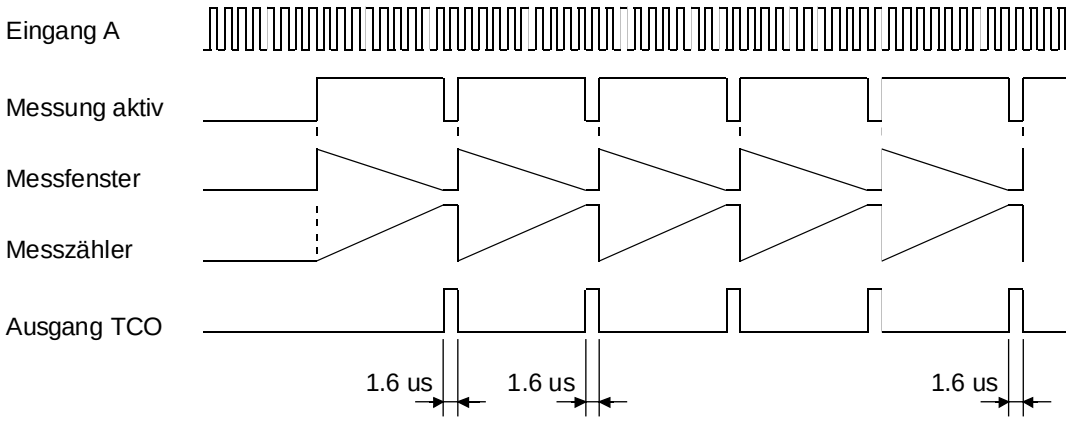
9.1.2 Beschreibung der Frequenzmessung

Der Bereich der Frequenzmessung reicht von 500 Hz bis 100 kHz.

Die Frequenzmessung kann parallel zum Counter verwendet werden. Die Frequenzmessung wird mit 2 Zählern zu je 16 Bit realisiert:

- Der eine dieser beiden Zähler, der Zähler des Messfensters, hat einen festen Clock von 1 kHz. Dies ist die Zeitbasis von 1 ms für das programmierbare Messfenster.
- Der andere Zähler, der Mess-Zähler, zählt die Signale, welche am Eingang 'A' während der Dauer des geöffneten Messfensters ankommen. Wurde das Messfenster für 1s (1000 ms) definiert, erscheint das Resultat im Messzähler direkt in Hz bzw. in Signalen pro Sekunde.

Die Frequenzmessung läuft automatisch, d.h. es wird während dem geöffneten Zeitfensters gemessen, danach das Signal 'TCO' aktiviert, der gemessene Wert gespeichert (Latch) der Messzähler wieder rückgestellt und die neue Messung gestartet. Die Dauer des Signals 'TCO' sowie das Speichern und die Rückstellung dauert jeweils 1.6 µs.



9.1.3 Konfigurierung der Frequenzmessung

Die Konfigurierung erfolgt im FB 'Init'. Für eine kontinuierliche Frequenzmessung ist der Parameter 9 auf '5' einzustellen (frequency-automatic).

Eine "single shot" Frequenzmessung ist denkbar, jedoch kaum üblich. Es könnte dafür der Parameter 9 auf '4' eingestellt werden. In diesem Fall würde nur eine einzige Messung durchgeführt. Danach müsste die ganze Prozedur für eine nächste Messung erneut durchgeführt werden.

Messbereich und Zeitfenster

Der Messbereich ist zwischen 0 und 65'535 (16 Bit).

Um eine Auflösung von 1‰ zu erhalten sind pro Messung mindestens 1000 Signale zu erfassen. Die Öffnungszeit des Messfensters ist von der zu messenden Frequenz abhängig.

Der Wert für die Öffnungszeit des Messfensters wird als ganzzahliger Wert in Millisekunden (ms) direkt als Parameter 10 eingegeben. Zum Messen von 100 kHz ist eine minimale Messzeit von 10 ms, zum Messen von 500 Hz eine Messzeit von mind. 2 s vorzusehen.

Ist die zu messende Frequenz noch kleiner, wird die Messzeit immer länger, was nicht in jeder Anwendung hingenommen werden kann. Für Frequenzen keiner etwa 100 Hz ist für eine ausreichende Genauigkeit innerhalb einer annehmbaren Messzeit die nachfolgend beschriebene Periodendauermessung der Frequenzmessung möglicherweise vorzuziehen.

Als Parameter 11 ist das Verhalten des Eingangs 'EnableM' und als Parameter 12 das Verhalten des Ausgangs 'TCO' einzugeben.

= 9	Messung-Mode Konfigurierung		Integer	0 - 5	4 = frequency-manual 5 = frequency-autom.
= 10	Lade-Wert für Messung		Integer	0 - 65535	Messfenster bei Frequenzmessung
= 11	EnableM (Aktivierung) der Konfigurierung für Messung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers
= 12	TCO-Konfigurierung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers

Die komplette Tabelle ist im Anhang, Seite A-2, dargestellt.

9.1.4 Prinzip der Programmierung

Aufgabe: An den Eingang 'A' wird eine Impulsfolge angelegt.
Es ist eine kontinuierliche (automatische) Frequenzmessung mit einem Messfenster von 1s (1000 ms) durchzuführen. Das Messresultat soll in Zähl-Einheiten (Hz) am Displaymodul angezeigt werden.

```
; ********************************************
; Basis-Anwenderprogramm für das PCD2.H110-Modul
; zur Frequenzmessung: frequ-01.src
; ********************************************
;
$include d2h110_b.equ
$group h110

xob       16

ld      r 0         ; Leeres Hilfsregister
          0

cfb       init      ; Initialisierung H110
        k 1         ;  Par 1  Modul-Nummer
          0         ;  Par 2  nicht verwendet
        r 0         ;  Par 3  nicht verwendet
        r 0         ;  Par 4  nicht verwendet
          0         ;  Par 5  nicht verwendet
          0         ;  Par 6  nicht verwendet
          0         ;  Par 7  Input A "normal"
          0         ;  Par 8  nicht verwendet
          5         ;  Par 9  Messmodus: "frequency auto."
          1000      ;  Par 10 Länge Messfenster in ms
          0         ;  Par 11 EnableM: "statisch-normal"
          2         ;  Par 12 TCO: "dynamisch-normal"

cfb       exec      ; Start der Messung
        k 1         ;  Modul-Nummer
          StartMs   ;  Befehl: Start Messung
        r 0         ;  leeres Register

exob
; -------------------------------------------------------

cob       0
          0

  wait: sth       EndMes_1 ; Messung beendet ?
        jr      l wait     ; wenn nicht, warten

        cfb       exec     ; Lesen des Messresultates
                k 1        ;  Modul-Nummer
                  RdMsImp  ;  Befehl: Read Measure in impulsion
                r 777      ;  gelesener Wert in R 777
        dsp     r 777      ; Anzeige an Displaymodul
;(wait:)
        ecob
        ; -------------------------------------------------------

        xob       20       ; Interrupt "INB1"
        com     o 101      ; invertiert nach jeder Messung
        exob

        $endgroup
```

Beschreibung des Programms

Hardwaremässig ist der Signalgeber, welcher die zu messenden Signale liefert, z.B. ein Impulsgenerator, an den Eingang 'A' zu verdrahten. Der Eingang 'B' bleibt offen. Der Eingang 'EnableC' wird bei einer Messung überhaupt nicht verwendet und bleibt offen. Der Eingang 'EnableM' wird an +24V gelegt, um die Frequenzmessung aktiv zu halten. *)

Der Ausgang 'TCO' wird an den Interrupt-Eingang 'INB1' der PCD verdrahtet. Es wird am Ende jeder Messung der XOB 20 aufgerufen und darin, in diesem Beispiel, der PCD-Ausgang A 101 invertiert. Dies, um die Funktion des Programms besser verfolgen zu können. Im vorliegenden Beispiel wird der Ausgang alle 1s invertiert.

Die Zeitbasis wird für dieses Beispiel mit 1000, für ein Zeitfenster von 1000 ms = 1s gewählt. Siehe auch "Messbereich und Zeitfenster"

Im XOB 16 (Kaltstart-Routine) wird die Initialisierung des H110 durchgeführt und die Messung gestartet. Ab diesem Moment läuft die Frequenzmessung selbständig ab (+24V am Eingang 'EnableM' vorausgesetzt).

Stellvertretend für ein Anwenderprogramm wird im COB das jeweilige Ende jeder Messung abgewartet und danach das Resultat aus dem Messzähler ausgelesen und im Debugger (Basis-Register + 8) oder an einem Displaymodul im Ganzzahlformat angezeigt. Die Messeinheit richtet sich nach der Definition der Zeitbasis. Im vorliegenden Beispiel sind dies Hertz. Wird anstelle des Befehls 'RdMsImp' der Befehl 'RdMsUnit' eingesetzt, erscheint das Resultat im Fliesspunktformat. Dieses Format kann allerdings am Displaymodul nicht angezeigt werden. Es ist der Debugger zu Hilfe zu nehmen.

Mit dem Befehl 'StopMs' kann eine angefangene Messung gestoppt werden. Das Resultat der unterbrochenen Messung ist ungültig. Eine neue Messung muss mit 'StartMs' wieder neu gestartet werden.

*) Nach einer Deaktivierung des 'EnableM' ist der Messwert ungültig. Eine neue Messung kann nur mit 'StartMs' wieder gestartet werden.

9.1.5 Kombination von Zähler und Frequenzmessung

Wie bereits erwähnt, kann die Frequenzmessung parallel zum Counter verwendet werden. Dies kann sehr schön an der Kombination der beiden Beispiele "Positionierung mit Inkrementaldrehgeber" (Anwendungsbeispiel Nr. 2, Abschnitt 8.7) und dem vorliegenden Prinzipbeispiel zur Frequenzmessung gezeigt werden. Es sind einzig am Positionierungsbeispiel die Befehle für die Positionsanzeige (Counter lesen) zu entfernen, da nur ein einziges Displaymodul vorhanden ist und dieses für die Anzeige der Frequenz verwendet wird.

Da beide Funktionen mit ein- und demselben H110 ausgeführt werden, ist eine gemeinsame, für das ganze Modul gültige Initialisierung durchzuführen. Die Initialisierung wird hier im IST des Positionierprogramms gemacht. Der Start der Messung wird auch hier, und zwar unmittelbar nach der Initialisierung, ausgeführt.

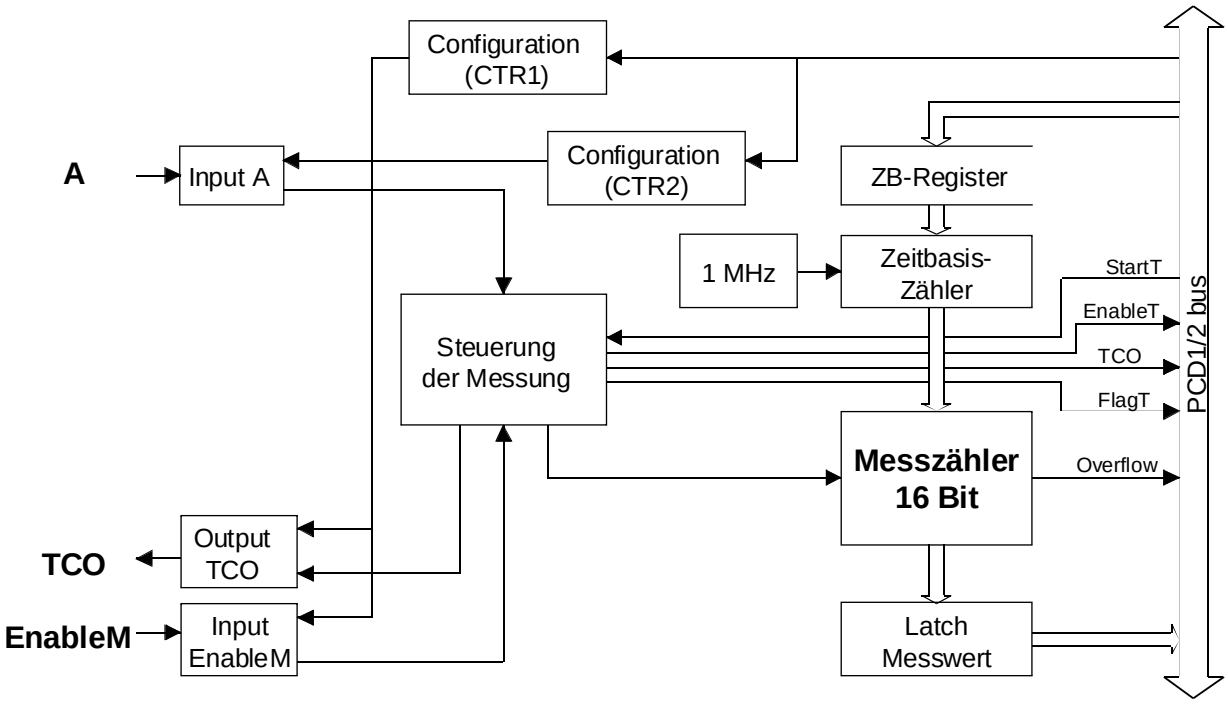
```
IST       10

$include d2h110_b.equ
$group h110

ld      r 0         ; Leeres Register
          0
ld      r 999       ; Startwert für Counter
          0
ld      r 998       ; Wert für Vergleichs-Register
          0

cfb       init      ; Initialisierung H110
        k 1         ;  Par 1  Modul-Nummer
          2         ;  Par 2  Zähl-Modus "x2"
        r 999       ;  Par 3  Startwert für Counter
        r 998       ;  Par 4  Wert für Vergleichs-Register
          1         ;  Par 5  Enable "statisch-invers"
          1         ;  Par 6  CCO "statisch-invers"
          0         ;  Par 7  Input A "normal"
          0         ;  Par 8  Input B "normal"
          5         ;  Par 9  Messmodus: "frequency-auto."
          100       ;  Par 10 Länge Messfenster in ms
          0         ;  Par 11 EnableM: "statisch-normal"
          2         ;  Par 12 TCO: "dynamisch-normal"

cfb     h exec      ; Start der Messung
        k 1         ;  Modul-Nummer
          StartMs   ;  Befehl: Start Messung
        r 0         ;  leeres Register

EST               ; 10
```

Das Projekt besteht also aus den beiden Programmen 'move-02.sfc' mit der Initialisierung des H110 und 'frequ-02' ohne den XOB 16.

Der Ablauf der Positionierung kann online im GRAFTEC und die dazugehörige Drehzahl bzw. Frequenz am Displaymodul verfolgt werden.

9.2 Periodendauermessung

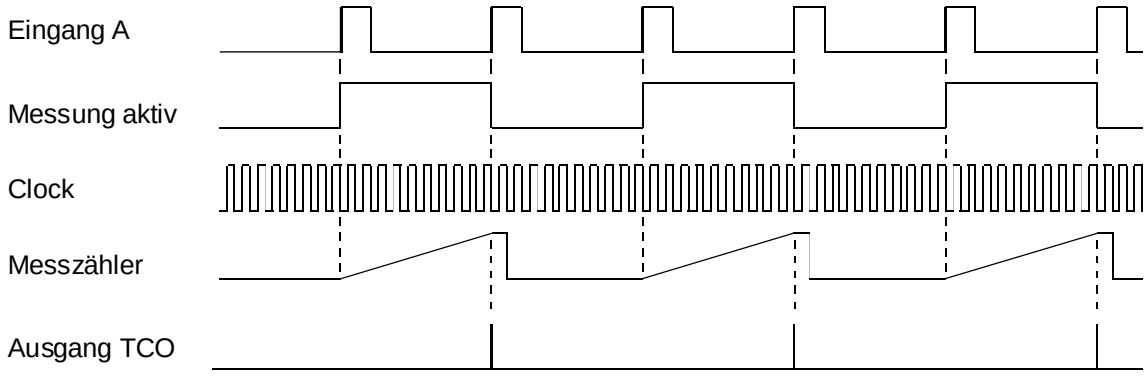
9.2.1 Blockschaltbild



9.2.2 Beschreibung der Periodendauermessung

Die Periodendauermessung wird mit 2 Zählern zu je 16 Bit realisiert:

- Der eine dieser beiden Zähler, der Zeitbasis-Zähler, hat einen festen Clock von 1 MHz, was eine Grundzeitbasis von 1 µs ergibt. Es wird hier die vom Anwender programmierte Zeitbasis erzeugt.
- Der andere Zähler, der Mess-Zähler, zählt die Zeitbasisimpulse zwischen 2 Flanken am Eingang 'A'. Es wird also bei aufeinaderfolgenden Flanken am Eingang 'A' immer zwischen 2 Flanken gemessen, danach erfolgt eine Messpause für die Wiederherstellung der Bereitschaft für die nächste Messung.



9.2.3 Konfigurierung der Periodendauermessung

Die Konfigurierung erfolgt im FB 'Init'. Für eine kontinuierliche Periodendauermessung ist der Parameter 9 auf '3' einzustellen (periode-automatic).

Eine manuelle Periodendauermessung ist möglich. Es müsste dafür der Parameter 9 auf '2' eingestellt werden. In diesem Fall würde nur eine einzige Periode gemessen. Danach müsste die Prozedur für die nächste Messung erneut durchgeführt werden.

Mit Parameter 7 kann das Signal am Eingang 'A' invertiert werden.

Messbereich und Zeitbasis

Der Messbereich ist zwischen 0 und 65'535 (16 Bit).

Mit der nachfolgend gezeigten Formel lässt sich der einzugebende Wert für die Zeitbasis errechnen:

$$n = \frac{T * 10^6}{clk} - 1$$

worin: T = Periodendauer in Sekunden
clk = Anzahl Clocksignale
n = zu ladender Wert

Beispiel: Die Periodendauer betrage 10s und die Anzahl Clocksignale sei 10'000

$$n = \frac{10 * 10^6}{10'000} - 1 = 999$$

Als Parameter 11 ist das Verhalten des Eingangs 'Enable' und als Parameter 12 das Verhalten des Ausgangs 'TCO' einzugeben.

= 7	Input A Konfigurierung		Integer	0 - 3	0: C=norm M=normal 1: C=invers M=normal 2: C=norm. M=invers 3: C=invers M=invers
= 9	Messung-Mode Konfigurierung		Integer	0 - 5	2 = periode-manual 3 = periode-automatic
= 10	Lade-Wert für Messung		Integer	0 - 65535	Zeitbasis bei Periodendauermessung
= 11	EnableM (Aktivierung) der Konfigurierung für Messung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers
= 12	TCO-Konfigurierung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers

Die komplette Tabelle ist im Anhang, Seite A-2, dargestellt.

9.2.4 Prinzip der Programmierung

Aufgabe: An den Eingang 'A' wird eine Lichtschranke (oder ein prellfreier Schalter) angelegt. Es soll jeweils die Dauer zwischen zwei 'H'-Signalen gemessen und am Diaplaymodul angezeigt werden. Die Konfigurierung soll so ausgelegt werden, dass bei einer Messzeit von 1 sek. 10000 Clockimpulse gezählt werden. Die Zeitbasis ist 99 (gemäss Formel)

```
; ********************************************
; Basis-Anwenderprogramm für das PCD2.H110-Modul
; zur Periodendauermessung: peri-01.src
; ********************************************
;
$include d2h110_b.equ
$group h110

xob       16

ld      r 0        ; Leeres Hilfsregister
          0

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1  Modul-Nummer
          0        ;  Par 2  nicht verwendet
        r 0        ;  Par 3  nicht verwendet
        r 0        ;  Par 4  nicht verwendet
          0        ;  Par 5  nicht verwendet
          0        ;  Par 6  nicht verwendet
          3        ;  Par 7  Input A "invers"
          0        ;  Par 8  nicht verwendet
          3        ;  Par 9  Messmodus: "periode-auto."
          99       ;  Par 10 Zeitbasis in µs
          0        ;  Par 11 EnableM: "statisch-normal"
          2        ;  Par 12 TCO: "dynamisch-normal"

cfb       exec     ; Start der Messung
        k 1        ;  Modul-Nummer
          StartMs  ;  Befehl: Start Messung
        r 0        ;  leeres Register

exob
; ---------------------------------------------------------

cob       0
          0

sth       EndMes_1 ; Messung beendet ?
wait:   jr      l wait     ; wenn nicht, warten

cfb       exec     ; Lesen des Messresultates
        k 1        ;  Modul-Nummer
          RdMsImp  ;  Befehl: Read Measure in impulsion
        r 777      ;  gelesener Wert in R 777
dsp     r 777      ; Anzeige an Displaymodul

ecob
; ---------------------------------------------------------

xob       20       ; Intrrupt "INB1"
com     o 101      ; invertiert nach jeder Messung
exob

$endgroup
```

Beschreibung des Programms

Hardwaremässig ist der Signalgeber, welcher die zu messenden Signale liefert, z.B. eine Lichtschranke, an den Eingang 'A' zu verdrahten. Der Eingang 'B' bleibt offen. Der Eingang 'EnableC' wird bei einer Messung überhaupt nicht verwendet und bleibt offen. Der Eingang 'EnableM' wird an +24V gelegt, um die Periodendauermessung aktiv zu halten. *)

Der Ausgang 'TCO' wird an den Interrupt-Eingang 'INB1' der PCD verdrahtet. Es wird am Ende jeder Messung der XOB 20 aufgerufen und darin, in diesem Beispiel, der PCD-Ausgang A 101 invertiert. Dies, um die Funktion des Programms besser verfolgen zu können. (Der Modus 'TCO statisch' kann nur im Zusammenspiel mit dem Modus 'periode-manual' verwendet werden).

Die Zeitbasis wird für dieses Beispiel mit 99, für ein Resultat von 10000 ms für 1s Messzeit, angegeben. Siehe auch "Messbereich und Zeitbasis".

Der Eingang 'A' wird mit 'Input A invers' (Parameter 7 = '3') parametriert, da die Lichtschranke ein "verkehrtes" Signal liefert.

Im COB wird das jeweilige Ende jeder Messung abgewartet und danach das Resultat aus dem Messzähler ausgelesen und im Debugger oder an einem Displaymodul im Ganzzahlformat angezeigt. Die Messeinheit richtet sich nach der Definition der Zeitbasis. Im vorliegenden Beispiel sind dies 10/1000stel-Sekunden. Wird anstelle des Befehls 'RdMsImp' der Befehl 'RdMsUnit' eingesetzt, erscheint das Resultat im Fliesspunktformat. Dieses Format kann allerdings am Displaymodul nicht angezeigt werden. Es ist der Debugger zu Hilfe zu nehmen.

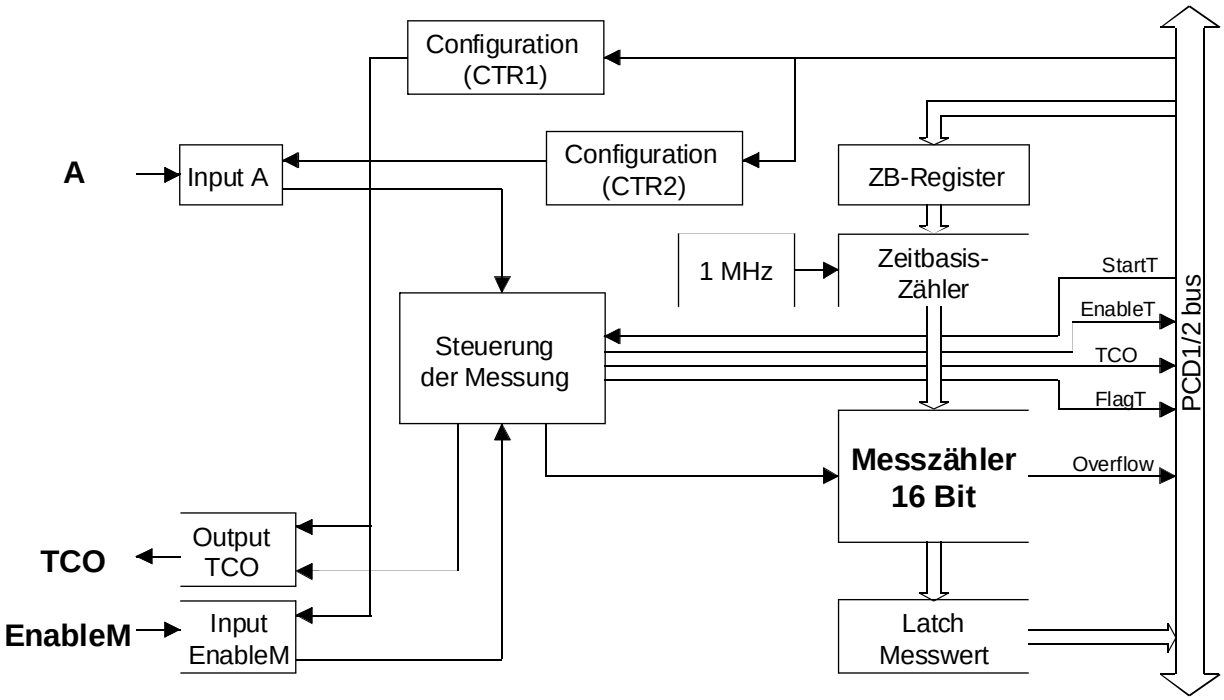
Mit dem Befehl 'StopMs' kann eine angefangene Messung gestoppt werden. Das Resultat der unterbrochenen Messung ist ungültig. Eine neue Messung muss mit 'StartMs' wieder neu gestartet werden.

*) Nach einer Deaktivierung des 'EnableM' ist der Messwert ungültig. Eine neue Messung kann nur mit 'StartMs' wieder gestartet werden.

Notizen

9.3 Impulsdauermessungen

9.3.1 Blockschaltbild

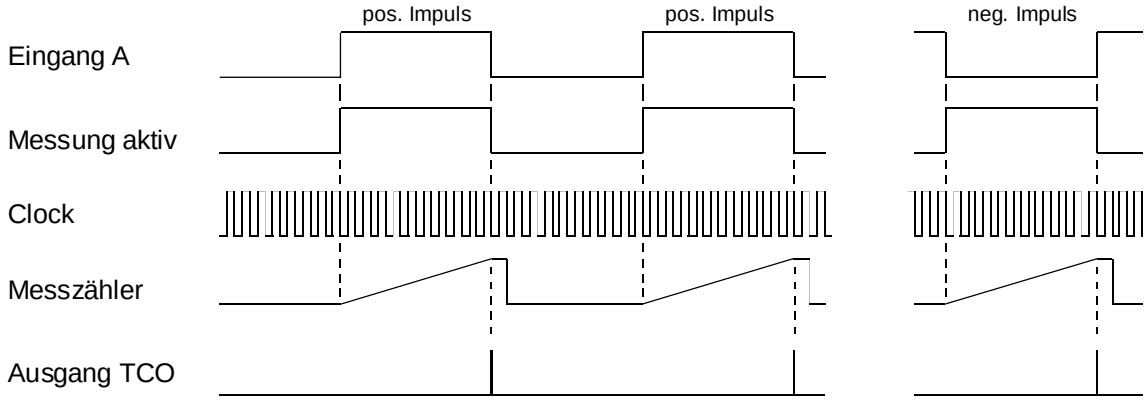


9.3.2 Beschreibung der Impulsdauermessung

Die Impulsdauermessung wird mit 2 Zählern zu je 16 Bit realisiert:

- Der eine dieser beiden Zähler, der Zeitbasis-Zähler, hat einen festen Clock von 1 MHz, was eine Grundzeitbasis von 1 µs ergibt. Es wird hier die vom Anwender programmierte Zeitbasis erzeugt.
- Der andere Zähler, der Mess-Zähler, zählt die Zeitbasisimpulse während der Eingang 'A' = H (positive oder normale Impulsdauermessung) oder während der Eingang 'A' = L ist (negative oder invertierte Impulsdauermessung).

Die negative oder invertierte Impulsdauermessung wird mittels der Intialisierung des Paramters 7 = '3' realisiert.



9.3.3 Konfigurierung der Impulsdauermessung

Die Konfigurierung erfolgt im FB 'Init'. Für eine kontinuierliche Periodendauermessung ist der Parameter 9 auf '1' einzustellen (impulse-automatic).

Eine manuelle Periodendauermessung ist möglich. Es müsste dafür der Parameter 9 auf '0' eingestellt werden. In diesem Fall würde nur ein einziger Impuls gemessen. Danach müsste die Prozedur für die nächste Messung erneut durchgeführt werden.

Messbereich und Zeitbasis

Der Messbereich ist zwischen 0 und 65'535 (16 Bit).

Mit der nachfolgend gezeigten Formel lässt sich der einzugebende Wert für die Zeitbasis errechnen:

$$n = \frac{T * 10^6}{clk} - 1$$

worin: T = Periodendauer in Sekunden
clk = Anzahl Clocksignale
n = zu ladender Wert

Beispiel: Die Impulsdauer betrage 10s und die Anzahl Clocksignale sei 10'000

$$n = \frac{10 * 10^6}{10'000} - 1 = 999$$

Als Parameter 11 ist das Verhalten des Eingangs 'Enable' und als Parameter 12 das Verhalten des Ausgangs 'TCO' einzugeben.

= 7	Input A Konfigurierung		Integer	0 - 3	0: C=norm M=normal 1: C=invers M=normal 2: C=norm. M=invers 3: C=invers M=invers
= 9	Messung-Mode Konfigurierung		Integer	0 - 5	0 = impulse-manual 1 = impulse-automatic
= 10	Lade-Wert für Messung		Integer	0 - 65535	Zeitbasis bei Impulsdauermessung
= 11	EnableM (Aktivierung) der Konfigurierung für Messung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers
= 12	TCO-Konfigurierung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers

Die komplette Tabelle ist im Anhang, Seite A-2, dargestellt.

9.3.4 Prinzip der Programmierung

Aufgabe: An den Eingang 'A' wird eine Lichtschranke (oder ein prellfreier Schalter) angelegt. Es soll jeweils die Dauer eines Impulses (Eingang A = H bzw. Eingang A = L) gemessen und am Displaymodul angezeigt werden. Die Konfigurierung soll so ausgelegt werden, dass bei einer Messzeit von 1 sek. 1000 Clockimpulse gezählt werden. Die Zeitbasis ist 999 (gemäss Formel)

```
; ********************************************
; Basis-Anwenderprogramm für das PCD2.H110-Modul
; zur Impulsdauermessung: imp-01.src
; ********************************************
;
$include d2h110_b.equ
$group h110

xob       16

ld      r 0        ; Leeres Hilfsregister
          0

cfb       init     ; Initialisierung H110
        k 1        ;  Par 1  Modul-Nummer
          0        ;  Par 2  nicht verwendet
        r 0        ;  Par 3  nicht verwendet
        r 0        ;  Par 4  nicht verwendet
          0        ;  Par 5  nicht verwendet
          0        ;  Par 6  nicht verwendet
          3        ;  Par 7  Input A "invers"
          0        ;  Par 8  nicht verwendet
          1        ;  Par 9  Messmodus: "impulse-auto."
          999      ;  Par 10 Zeitbasis in µs
          0        ;  Par 11 EnableM: "statisch-normal"
          2        ;  Par 12 TCO: "dynamisch-normal"

cfb       exec     ; Start der Messung
        k 1        ;  Modul-Nummer
          StartMs  ;  Befehl: Start Messung
        r 0        ;  leeres Register

exob
; -------------------------------------------------------

cob       0
          0

sth       EndMes_1 ; Messung beendet ?
wait:   jr      l wait     ; wenn nicht, warten

cfb       exec     ; Lesen des Messresultates
        k 1        ;  Modul-Nummer
          RdMsImp  ;  Befehl: Read Measure in impulsion
        r 777      ;  gelesener Wert in R 777
dsp     r 777      ; Anzeige an Displaymodul

ecob
; -------------------------------------------------------

xob       20       ; Intrrupt "INB1"
com     o 101      ; invertiert nach jeder Messung
exob

$endgroup
```

Beschreibung des Programms

Hardwaremässig ist der Signalgeber, welcher die zu messenden Signale liefert, z.B. eine Lichtschranke, an den Eingang 'A' zu verdrahten. Der Eingang 'B' bleibt offen. Der Eingang 'EnableC' wird bei einer Messung überhaupt nicht verwendet und bleibt offen. Der Eingang 'EnableM' wird an +24V gelegt, um die Periodendauermessung aktiv zu halten. *)

Der Ausgang 'TCO' wird an den Interrupt-Eingang 'INB1' der PCD verdrahtet. Es wird am Ende jeder Messung der XOB 20 aufgerufen und darin, in diesem Beispiel, der PCD-Ausgang A 101 invertiert. Dies, um die Funktion des Programms besser verfolgen zu können.

Die Zeitbasis wird für dieses Beispiel mit 999, für ein Resultat in Millisekunden, angegeben. Siehe auch "Messbereich und Zeitbasis"

Der Eingang 'A' wird mit 'Input A invers' (Parameter 7 = '3') parametriert, da die Lichtschranke ein "verkehrtes" Signal liefert.

Im COB wird das jeweilige Ende jeder Messung abgewartet und danach das Resultat aus dem Messzähler ausgelesen und im Debugger (Basis-Register + 8) oder an einem Displaymodul im Ganzzahlformat angezeigt. Die Messeinheit richtet sich nach der Definition der Zeitbasis. Im vorliegenden Beispiel sind dies Millisekunden. Wird anstelle des Befehls 'RdMsImp' der Befehl 'RdMsUnit' eingesetzt, erscheint das Resultat im Fliesspunktformat. Dieses Format kann allerdings am Displaymodul nicht angezeigt werden. Es ist der Debugger zu Hilfe zu nehmen.

Mit dem Befehl 'StopMs' kann eine angefangene Messung gestoppt werden. Das Resultat der unterbrochenen Messung ist ungültig. Eine neue Messung muss mit 'StartMs' wieder neu gestartet werden.

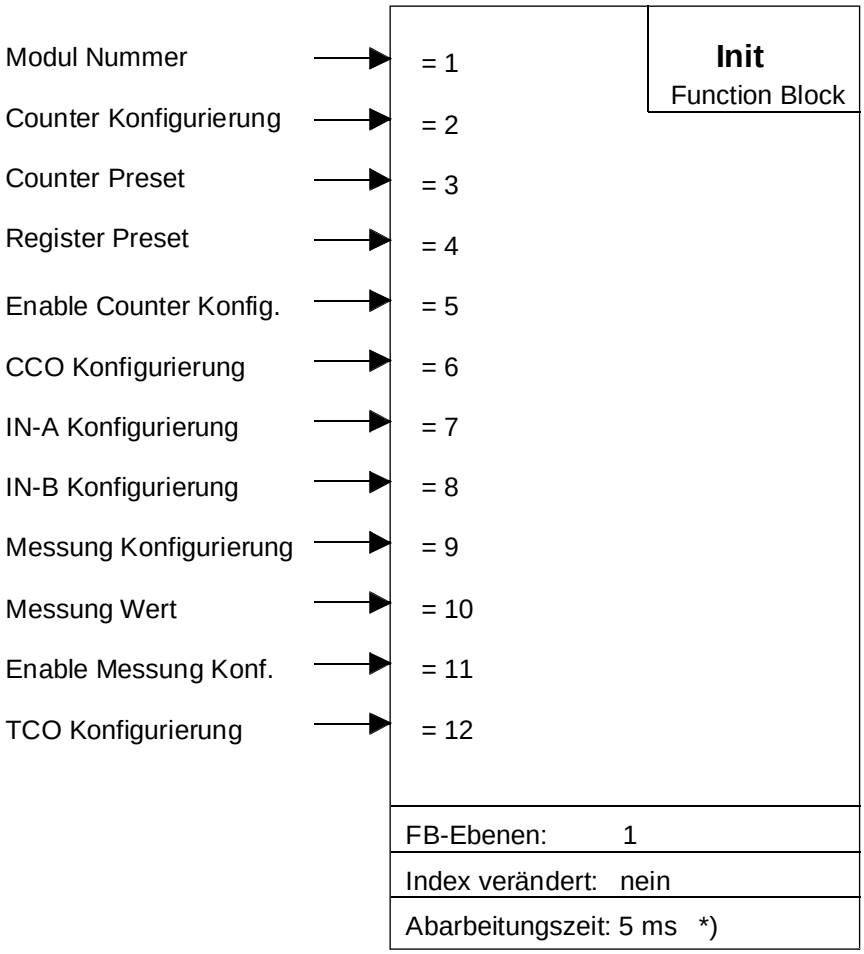
*) Nach einer Deaktivierung des 'EnableM' ist der Messwert ungültig. Eine neue Messung kann nur mit 'StartMs' wieder gestartet werden.

Notizen

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

Der Funktionsblock 'Init'

Init **FB:** Initialisierung eines H110 Moduls



*) gemessen an einer PCD2.M120

Funktionsbeschreibung:

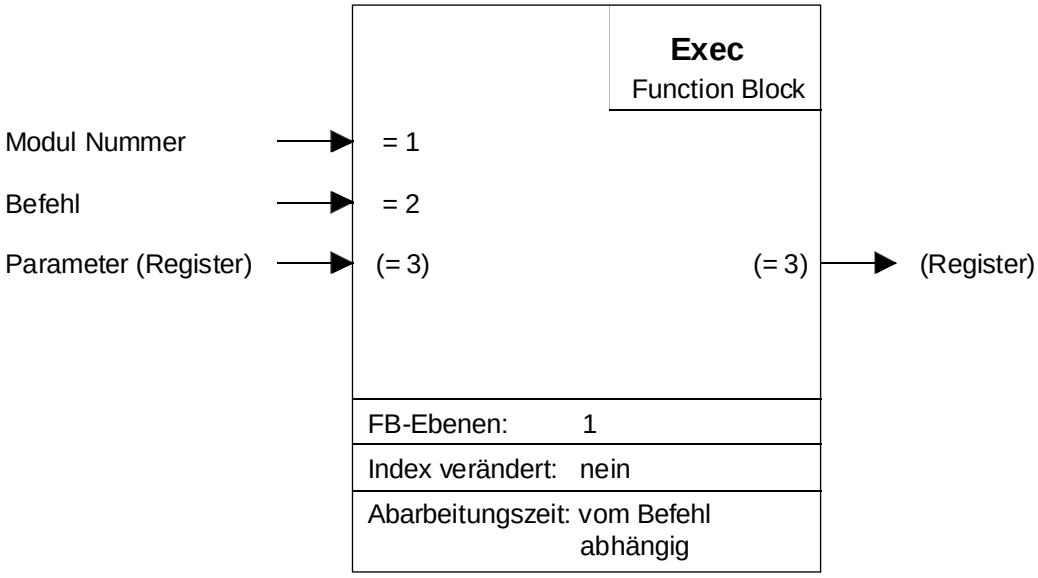
Dieser FB definiert die Einstellungen eines PCD2.H110 Moduls und liest die Basisadresse des Moduls aus der D2H110_B.MBA-Datei.

Parameter '1' ist als Konstante 'K', Parameter '3' und '4' sind als PCD-Registeradressen (absolut oder symbolisch), alle andern Parameter als ganzzahlige (Integer-) Werte anzugegeben.

Para-meter	Bedeutung	Typ	Format	Wert	Kommentar
= 1	Modul-Nummer	K	K n	K 1 – K 16	
= 2	Counter-Mode Konfigurierung		Integer	0 - 4	0 / 1 = mode x1 2 = mode x2 3 = mode x3 4 = mode x4
= 3	Load Counter Pre-set-Wert	R	Integer	0 - 16777215	Start-Wert für den Counter
= 4	Load Register Pre-set-Wert (Vergleichswert)	R	Integer	0 - 16777215	CCO Set-Wert
= 5	EnableC (Aktivie-rung) der Counter Konfigurierung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers
= 6	CCO Konfigurie-rung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers
= 7	Input A Konfigurie-rung		Integer	0 – 3	0: C=norm M=normal 1: C=invers M=normal 2: C=norm. M=invers 3: C=invers M=invers
= 8	Input B Konfigurie-rung		Integer	0 - 1	0 = normal 1 = invers
= 9	Messung-Mode Konfigurierung		Integer	0 - 5	0 = impulse-manual 1 = impulse-autom. 2 = periode-manual 3 = periode-autom. 4 = frequency-manual 5 = frequency-autom.
= 10	Lade-Wert für Messung		Integer	0 - 65535	Messfenster bei Fre-quenzmessung, Zeitba-sis bei Periodendauer-messung
= 11	EnableM (Aktivie-rung) der Konfigu-rierung für Mes-sung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers
= 12	TCO Konfigurie-rung		Integer	0 - 3	0 = statisch-normal 1 = statisch-invers 2 = dynamisch-normal 3 = dynamisch-invers

Der Funktionsblock 'Exec'

Exec: **FB:** Ausführung eines Befehls für das H110-Modul



Mit diesem FB werden die Ausführungsbefehle zum PCD2.H110-Modul geschickt.

Die Modul-Nr. (Parameter 1) muss eine Konstante sein (k 1…k 16). Die Basisadresse wird in der Datei 'D2H110_B.MBA' definiert. Die FBs unterstützen max. 16 PCD2.H110-Module pro PCD-System.

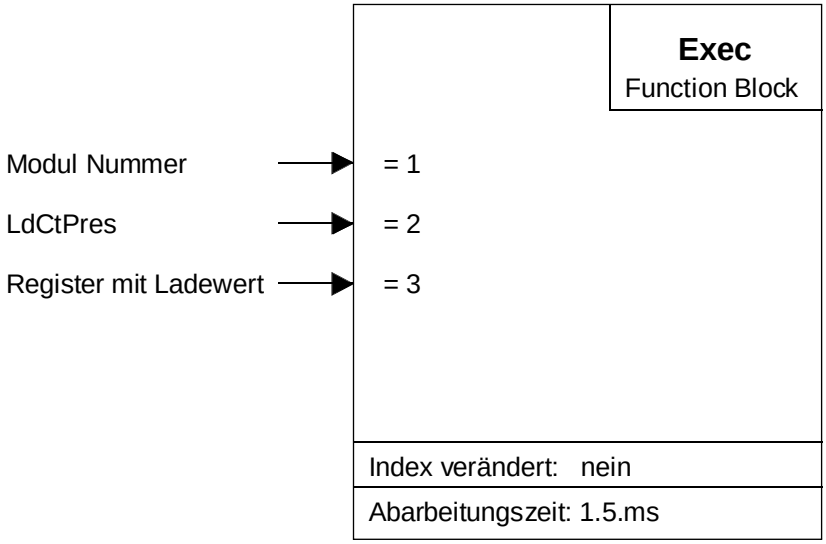
Die einzelnen Befehle (Parameter 2) werden auf den folgenden Seiten behandelt.

Der Parameter eines Befehls (z.B. der Beschleunigungswert beim Befehl LdCtPres) 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.H110 (FB-Parameter)

LdCtPres

Befehl: Load counter preset
(Laden des Counter-Vorwahlwertes)



Funktionsbeschreibung:

Mit diesem Befehl wird der Vorwahlwert in den Counter geladen.

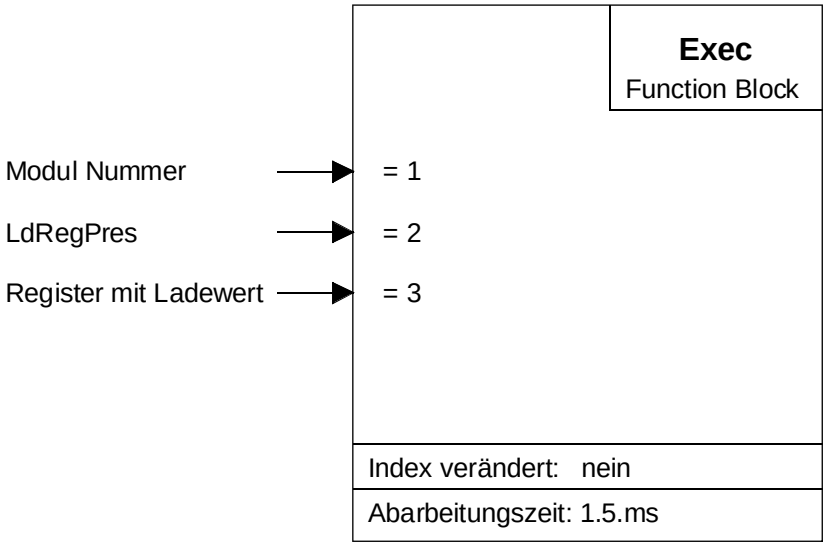
Mit diesem Befehl wird der Counter nach einem Start-Befehl auf- oder abwärts zu zählen beginnen. Mit andern Worten: Dieser Befehl blockiert eine laufende Zählung und es ist zum Weiterzählen ein neuer Start-Befehl 'StartCt' durchzuführen.

Beschreibung der beteiligten Ein- und Ausgabeelemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdCtPres				24 bit Counter
= 3	Register mit Ladewert	R	Integer	0 - 16777215	

LdRegPres

Befehl: Load register preset
(Laden des Vergleichs-Registers)



Funktionsbeschreibung:

Mit diesem Befehl wird ein Wert in das Vergleichs-Register geladen.

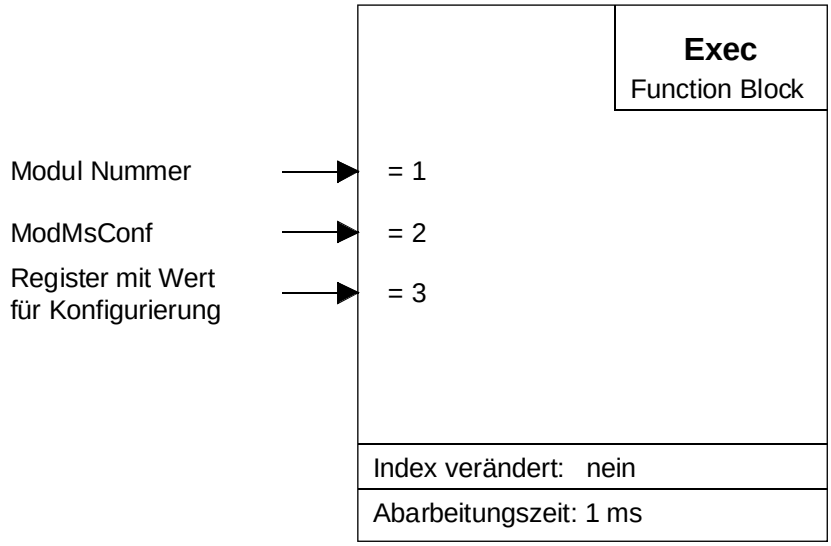
Dieser Wert wird zum Vergleich mit dem Zählwert des Counters verwendet. Sind beide Werte gleich, wird der CCO gemäss der Konfiguration beeinflusst.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdRegPres				24 Bit Counter
= 3	PCD-Reg. mit Ladewert	R	Integer	0 - 16777215	

ModMsConf

Befehl: Measure mode configuration
(Konfigurierung des Mess-Modus)



Funktionsbeschreibung:

Mit diesem Befehl wird die Mess-Modus Konfiguration ins dafür vorgesehene Register geladen.

Es stehen die Modi 'manual' und 'automatic' zur Verfügung.

Manual-Modus bedeutet, dass nach jeder Messung ein neuer 'Start Counter'-Befehl durchzuführen ist. Im Automatik-Modus wird der Start jeweils im Anwenderprogramm ausgelöst und es wird mit einer Bedingung, wie z.B. dem CCO-Status, die Messung beendet.

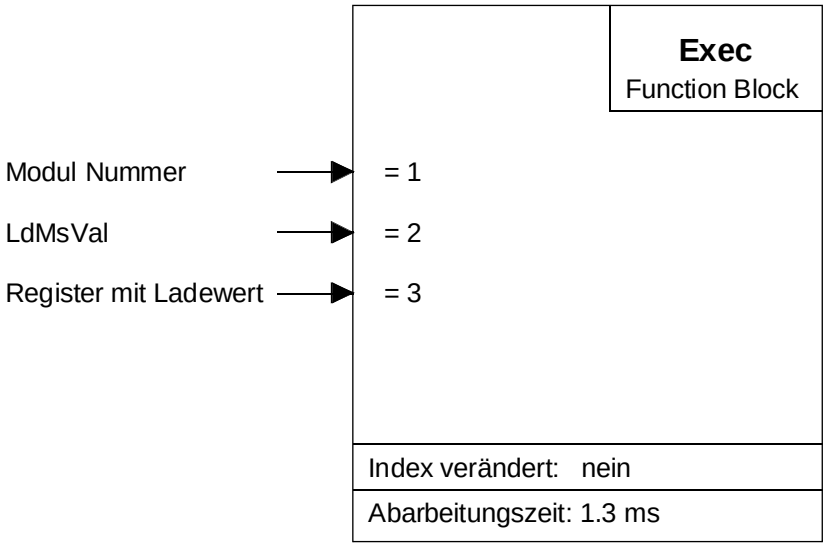
Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: Mod MsConf				
= 3	PCD-Register mit Wert für die Konfigurierung	R	Integer	0 - 5	Default Wert = 0

Dieser Befehl kommt nur dann zur Anwendung, wenn die Konfigurierung des Mess-Modus während dem Programmablauf geändert werden soll. Im Normalfall wird die Konfigurierung am Programmanfang im FB 'Init' durchgeführt (Parameter 9).

LdMsVal

Befehl: Load measure value
(Lade Messwert)



Funktionsbeschreibung:

Mit diesem Befehl wird die Länge des Messfensters für die Frequenzmessung oder der Teilerwert für die Periodendauermessung definiert.

Für die Frequenzmessung gilt:

Um eine Auflösung von 0.1% zu erhalten, müssen pro Messung mindestens 1000 Impulse am Eingang 'A' gezählt werden. Die Länge des Messfenster ist also von der zu messenden Frequenz abhängig, z.B.:

für 100 kHz	→	Länge des Messfensters:	10 ms
für 500 Hz	→	Länge des Messfensters:	2000 ms

Formel:

f = Frequenz [Hz]
t = Länge des Messfensters [ms]
R = Auflösung

$$t = \frac{1000 * R}{f}$$

Für die Periodendauermessung, siehe nächste Seite

Für die Periodendauermessung gilt:

Die Messzeit entspricht immer einer Periode des Messsignals oder der Länge eines Impulses.

Für die Messung einer einzigen Periode eines Impuszuges:

für 500 Hz ist die Messzeit 2 ms
für 270 µHz ist die Messzeit 1 Stunde

Mit der nachfolgend gezeigten Formel kann der Wert für die Zeitbasis errechnet werden. Um eine Auflösung von 0.1% zu erhalten müssen mindestens 1000 Messimpulse erfasst werden.

t = Periodendauer oder Impulslänge
n = zu ladender Wert
clk = Anzahl Messimpulse

$$n = \frac{10^6 * t}{clk} - 1$$

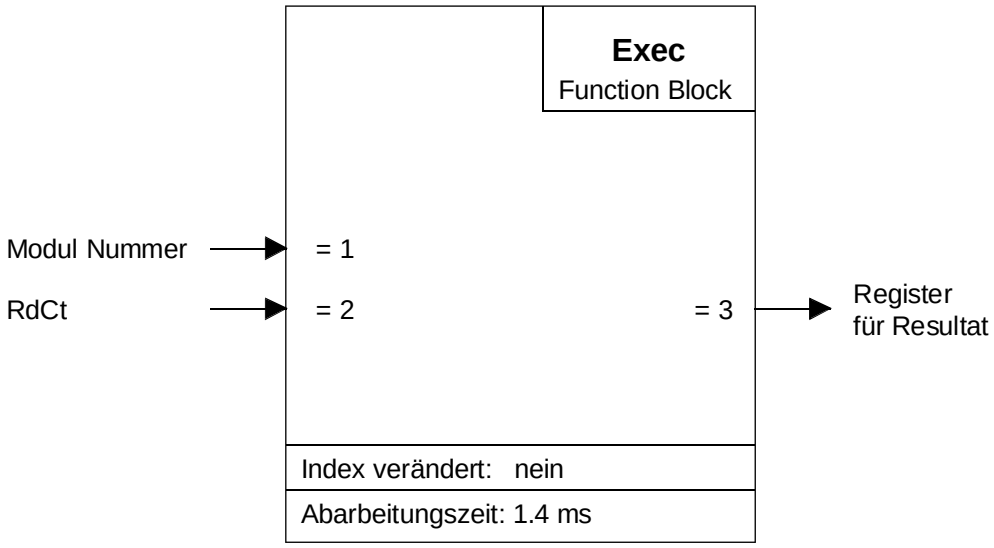
Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bem.
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: LdMsVal				
= 3	PCD-Reg. mit Ladewert	R	Integer	0 - 65535	16 Bit

Dieser Befehl kommt nur dann zur Anwendung, wenn der Parameter während dem Programmablauf geändert werden soll. Im Normalfall wird dieser Wert am Programmanfang im FB 'Init' angegeben (Parameter 10).

RdCt

Befehl: Read counter
(Counter lesen)



Funktionsbeschreibung:

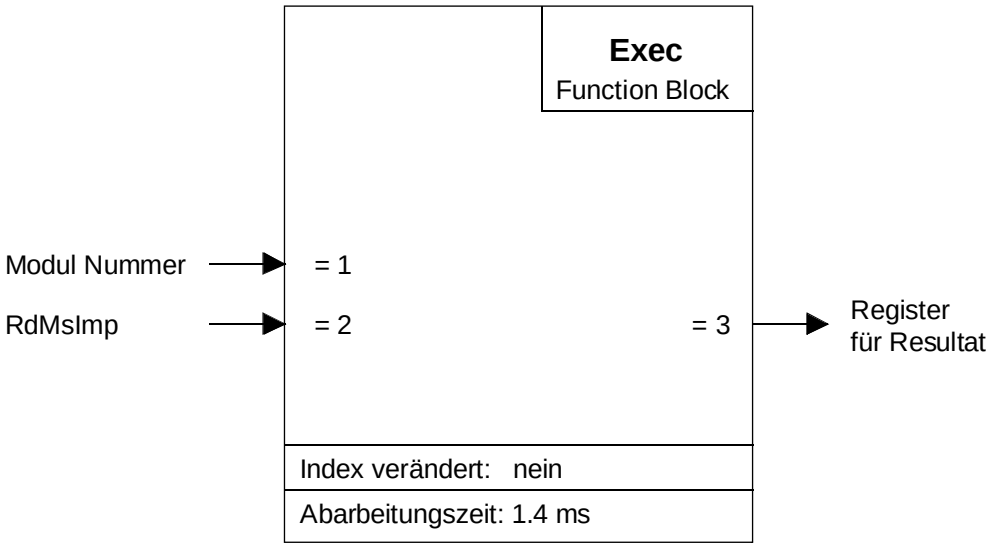
Mit diesem Befehl wird der aktuelle Zählerstand ausgelesen.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bem.
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdCt				
= 3	PCD-Reg. für Resultat	R	Integer	0 - 16777215	24 Bit

RdMsImp

Befehl: Read measure in impulsion
(Lesen des Messreultats als Zählwert)



Funktionsbeschreibung:

Mit diesem Befehl wird das Messresultat einer Messung gelesen.

Bei einer Frequenzmessung ist das Messresultat die Anzahl Perioden der zu messenden Frequenz, welche während der Zeit, in welcher das Messfenster offen war, gezählt wurden.

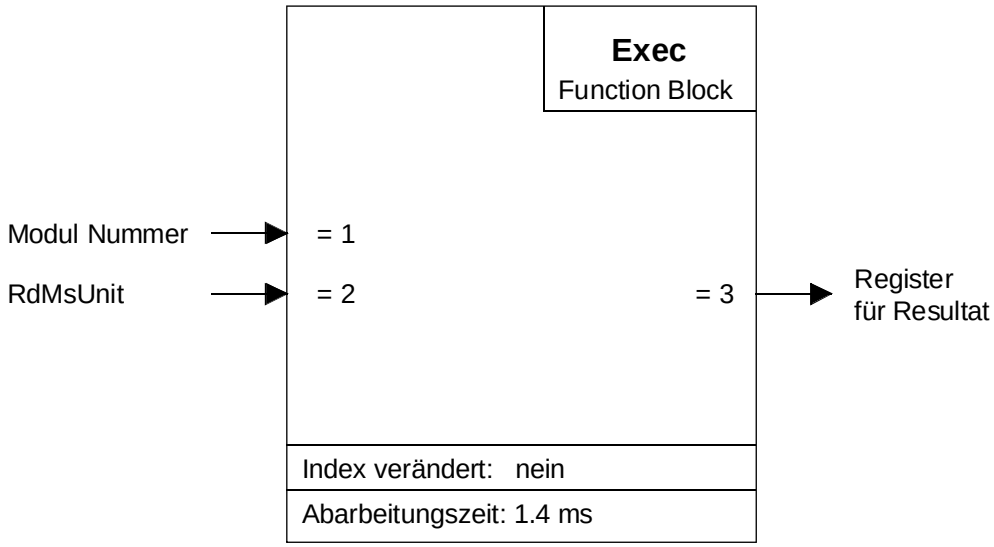
Bei einer Impuls- oder Periodendauermessung ist das Resultat die Anzahl Impulse der modulintern erzeugten Messfrequenz, welche zwischen zwei gleichen Flanken bei der Periodendauermessung oder zwischen zwei unterschiedlichen Flanken bei der Impulsdauermessung gezählt wurden.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bem.
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdMsImp				
= 3	PCD-Reg. für Resultat	R	Integer	0 - 65535	16 Bit

RdMsUnit

Befehl: Read measure in Unit
(Lesen des Messresultats in Einheiten)



Funktionsbeschreibung:

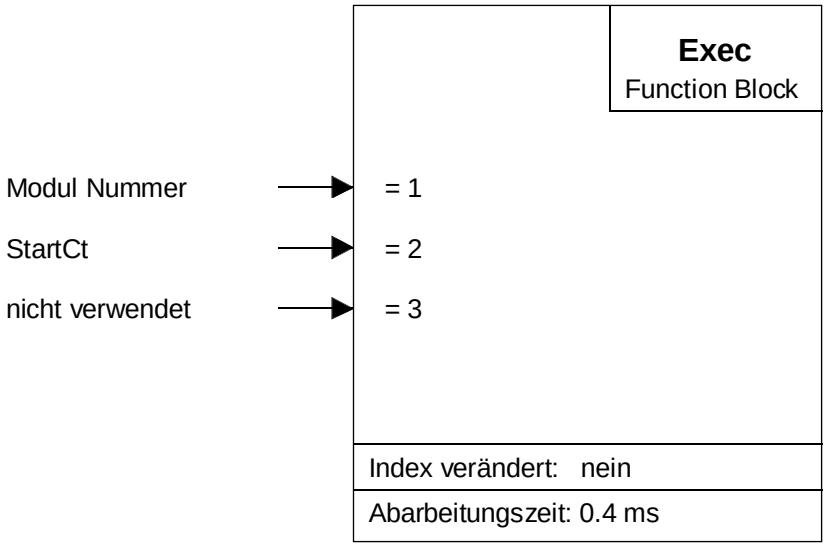
Mit diesem Befehl wird das Messresultat in spezifizierten Einheiten gelesen.

Bei einer Frequenzmessung wird das Resultat in Hertz (Hz), bei einer Impuls- oder Periodendauermessung in Sekunden (s) ausgegeben. Es kommt in beiden Fällen das Fliesspunktformat zur Anwendung.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Par.	Bezeichnung/Funktion	Typ	Format	Wert	Bem.
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdMsUnit				
= 3	PCD-Reg. für Resultat	R	FP		

StartCt **Befehl:** Start counter



Funktionsbeschreibung:

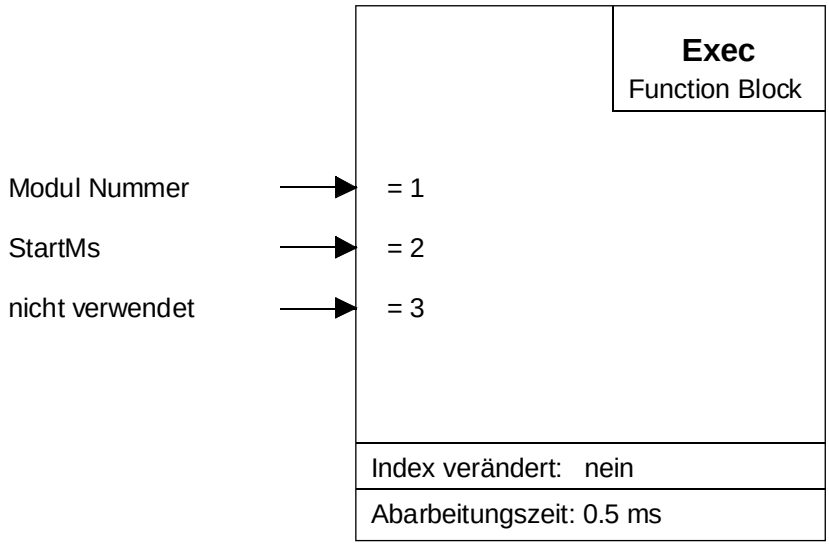
Mit diesem Befehl wird der Counter gestartet, wenn der 'EnableC' aktiv ist. Falls der 'EnableC' desaktiviert ist, wird der Start des Counters gespeichert, bis der 'EnableC' aktiv wird.

In einem 'Manual-Modus' ist dieser Befehl nach jeder ausgeführten Messung zu aktivieren. Bei den automatischen Modi erfolgt der Start nur einmal im Anwenderprogramm.

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

StartMs

Befehl: Start measure
(Start einer Messung)



Funktionsbeschreibung:

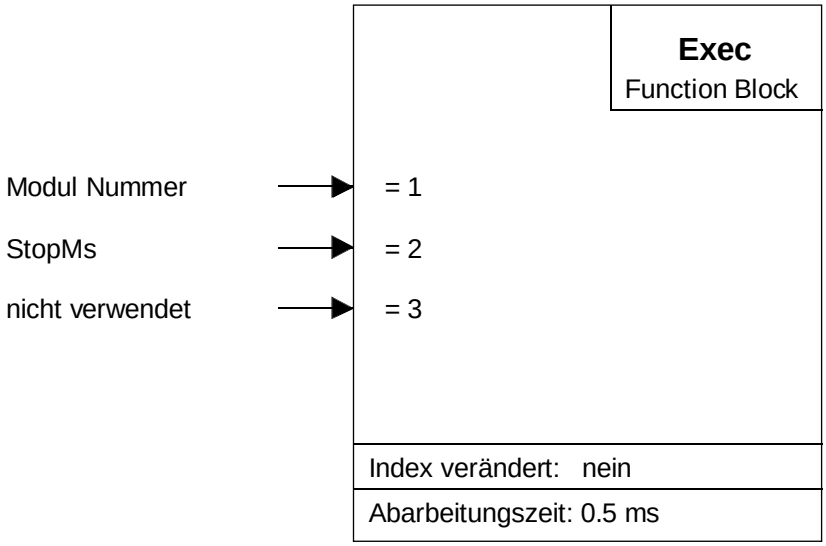
Mit diesem Befehl wird eine Messung gestartet, wobei der 'EnableM' aktiv sein muss. Falls in einer Anwendung der 'EnableM' desaktiviert und wieder aktiviert wird, ist ein neuer Befehl 'StartMs' durchzuführen.

In einem 'Manual-Modus' ist dieser Befehl nach jeder ausgeführten Messung zu aktivieren. Bei den automatischen Modi erfolgt der Start nur einmal im Anwenderprogramm.

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

StopMs

Befehl: Stop measure
(Stoppen einer Messung)



Funktionsbeschreibung:

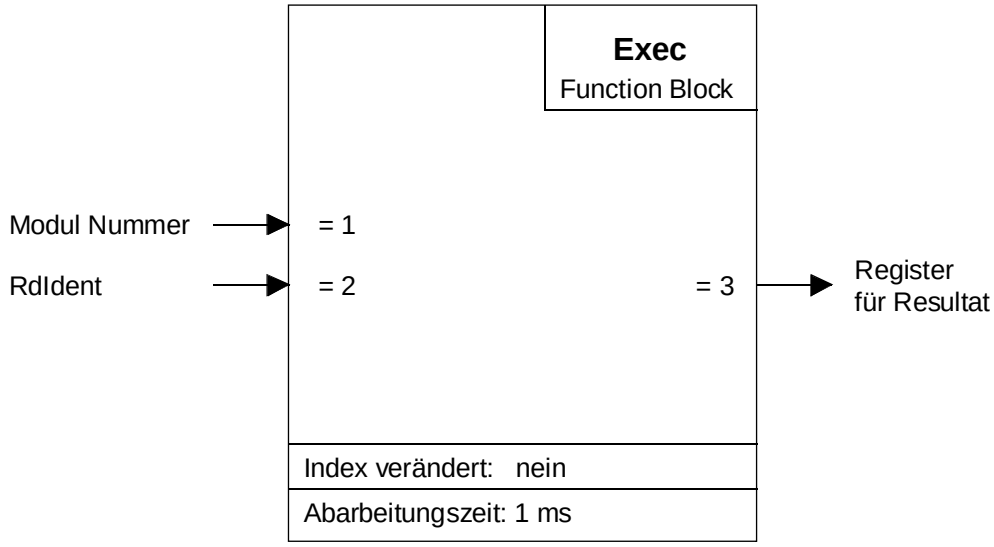
Mit diesem Befehl wird eine Messung abgebrochen. Das Resultat im 16-Bit Zähler ist ungültig.

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

Für einen Neustart ist der Befehl 'StartMs' wieder auszuführen.

RdIdent

Befehl: Read module identification
(Lesen der Modul-Identifikation)



Funktionsbeschreibung:

Dieser Befehl dient der Überprüfung der korrekten Funktion des PCD2.H110-Moduls und liest die Version des FPGA. Läuft das Modul korrekt, wird der Wert 17xx ausgegeben, siehe untenstehende Tabelle. Sollte das Modul eine Fehler haben oder ist dieses falsch adressiert, wird der Wert Null gelesen.

Beschreibung der beteiligten Ein- und Ausgangselemente:

Param.	Bezeichnung/Funktion	Typ	Format	Wert	Bemerkung
= 1	Modul-Nummer	K		1 - 16	
= 2	Befehl: RdIdent				
= 3	PCD-Reg. für Resultat	R	Integer	12 Bit	0 → Fehler

Tabelle der gültigen Modul-Identifikatoren:

Value	FPGA-Version
2759	Version HC0
1761	Version HC1
1762	Version HC2
1763	Version HC3
...	...
3215	Version HDF

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.H110

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: