

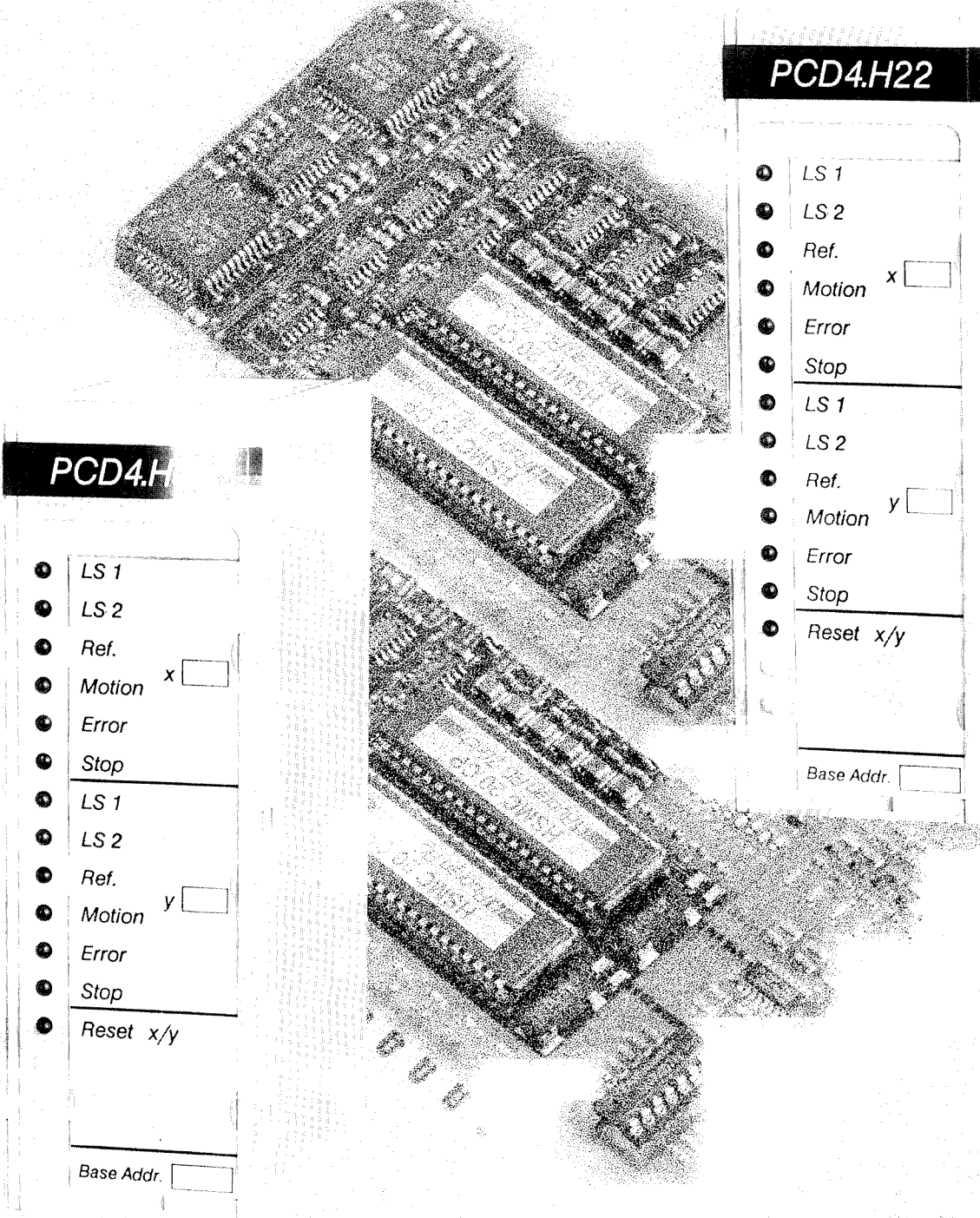
19.11.90



SAIA® PLC

Speicherprogrammierbare Steuerungen

Handbuch Positioniermodule für Schrittmotoren PCD4.H2..



PCD4.H21

- LS 1
 - LS 2
 - Ref.
 - Motion x
 - Error
 - Stop
-
- LS 1
 - LS 2
 - Ref.
 - Motion y
 - Error
 - Stop
 - Reset x/y
- Base Addr.

PCD4.H22

- LS 1
 - LS 2
 - Ref.
 - Motion x
 - Error
 - Stop
-
- LS 1
 - LS 2
 - Ref.
 - Motion y
 - Error
 - Stop
 - Reset x/y
- Base Addr.



SAIA® Speicherprogrammierbare Steuerungen

Handbuch

Positioniermodul für Schrittmotoren

PCD4.H2..

SAIA AG 1990 Alle Rechte vorbehalten
Ausgabe 09/90 D2

Technische Änderungen vorbehalten

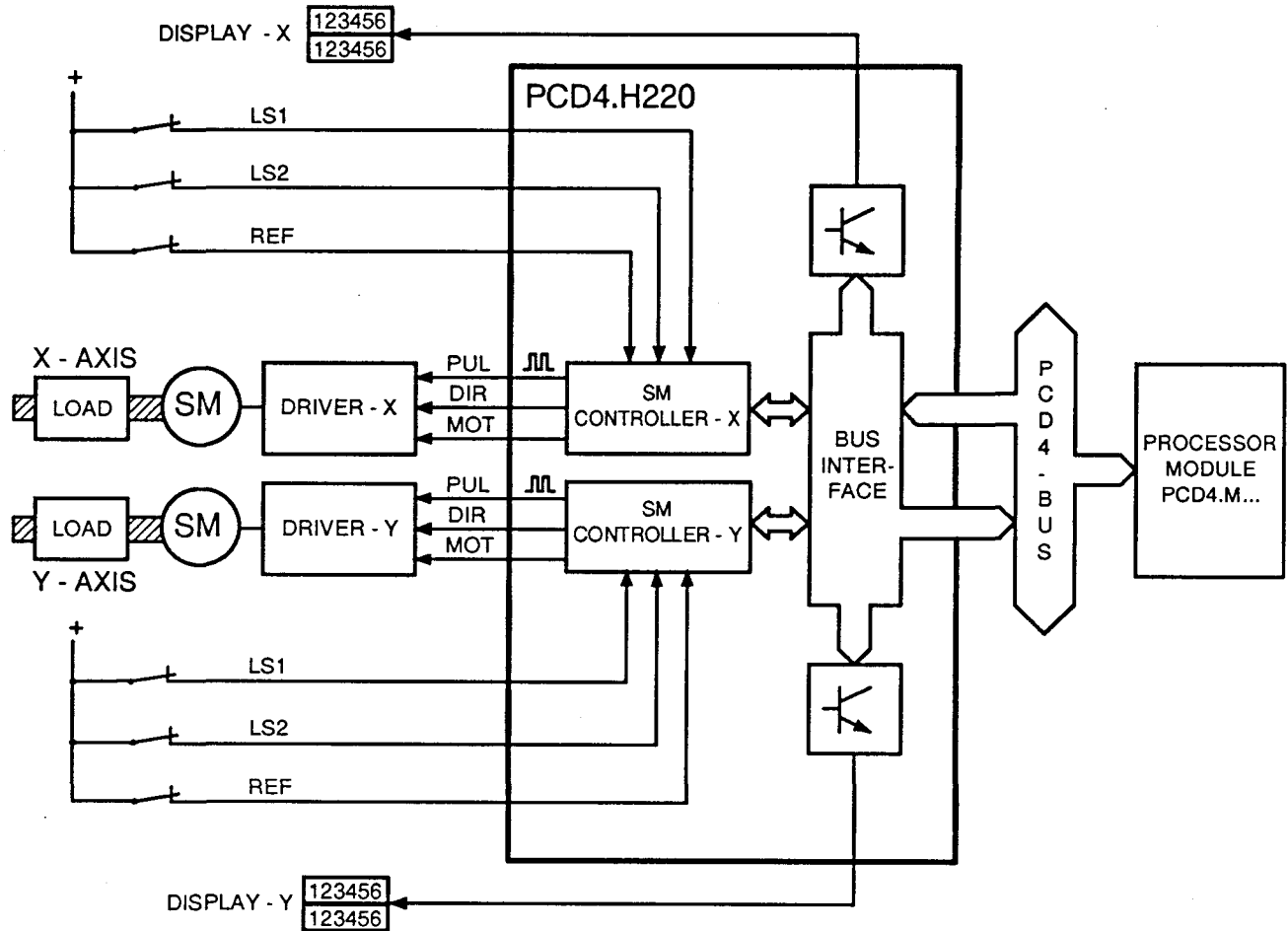
Verkaufspreis sFr. 80.-

Inhaltsverzeichnis

| | | |
|------------|------------------------------------------------------------|------------|
| 1. | Einleitung | |
| 2. | Technische Daten | |
| 3. | Präsentation | |
| 3.1 | Leiterplatte | Seite 3-1 |
| 3.2 | Frontplatte | Seite 3-2 |
| 4. | Blockschaltbild | |
| 5. | Anschlüsse und Adressierung | |
| 5.1 | Anschlussklemmen | Seite 5-1 |
| 5.2 | Stopschalter, Endschalter und Referenzschalter | Seite 5-2 |
| 5.3 | Installation der H2...-Module | Seite 5-3 |
| 5.4 | Adressierung | Seite 5-5 |
| 6. | Programmerstellung für das H2-Modul | |
| 6.1 | Installation der Software | Seite 6-1 |
| 6.2 | Anweisungen für den Einsatz der Funktionsbausteine | Seite 6-2 |
| 6.3 | Definition der Symbole in H2DEF.SRC | Seite 6-4 |
| 6.4 | Organisation des Datenverkehrs zwischen CPU und H2-Modulen | Seite 6-7 |
| 6.5 | Betriebsarten | Seite 6-10 |
| 7. | Die einzelnen Funktionsbausteine | |
| 7.1 | Übersicht der Funktionsbausteine | Seite 7-1 |
| 7.2 | Beschreibung der Funktionsbausteine (Befehle) | Seite 7-3 |
| 8. | Fehlererkennung und -Behandlung | |
| 9. | Anwendungsbeispiele | |
| 10. | Alphabetische Symbol-Übersicht | |

1. Einleitung

Blockschema eines Schrittmotorantriebes:



Funktion und Anwendung

Das Schrittmotormodul PCD4.H2.. dient der Ansteuerung von einem oder zwei Schrittmotorantrieben. Diese Antriebe bestehen immer aus einem Leistungsteil (Driver) mit zusätzlicher Speisung und einem Schrittmotor (SM).

Die im H2..-Modul enthaltenen Prozessoren sind digitale Hochleistungsbausteine, welche die Leistungsstufen mit den Signalen für Schritimpulse (PUL), Drehrichtungs- (DIR) und Bewegungs-Signale (MOT) versorgen. Die für den Ablauf der Schrittmotor-Bewegungen nötigen Befehle werden vom PCD4-Anwenderprogramm über den PCD4-Bus den Schrittmotorprozessoren mitgeteilt. Nachdem alle Parameter (Geschwindigkeit, Beschleunigung und Zielposition) eingegeben sind, rechnet der Prozessor ein mathematisch genaues Geschwindigkeitsprofil aus, das erlaubt, im Hochlaufbetrieb mittels Rampen die anzufahrende Position ohne Schritverlust zu erreichen.

Während des Bewegungsablaufes arbeitet der Schrittmotor-Prozessor vollkommen selbstständig, insbesondere werden auch die Endschalter-, Referenzschalter und Stop-Signale wahrgenommen und autonom verarbeitet.

Jeder Schrittmotor kann unabhängig oder koordiniert zu anderen Achsen arbeiten. In der gleichen PCD4-Steuerung können theoretisch bis max. 16 H220-Module (32 Achsen) betrieben werden. Der Frequenzbereich beträgt 33 bis 20'000 Hz.

Das Schrittmotormodul erlaubt ausserdem die Anschaltung eines oder zwei Displaymodule vom Typ PCA2.D14.

Typische Einsatzgebiete

- Positioniersteuerung hoher Genauigkeit für Fräs- und Schleif-Maschinen
- Preisgünstige Palletierungs- und Montage-Antriebe
- Antriebe, die ein hohes Stillstand-Drehmoment erfordern
- Antriebe für Teilapparate
- Antriebe mit hoher Systemzuverlässigkeit

Wichtigste Eigenschaften

- Das Modul erlaubt eine vollkommen autonome Steuerung und Überwachung des Bewegungsablaufes.
- Automatik und Handbetrieb sind im Modul integriert.
- Die Signaleingänge für Endschalter und Referenzschalter entsprechen den gebräuchlichen Regeln der Antriebstechnik.
- Die Signalausgänge erlauben die Anschaltung von Leistungsstufen und Schrittmotoren aller Art.
- Geschwindigkeit und Beschleunigung werden quartzgenau gesteuert.
- Die Bewegungsabläufe können mit höchster Genauigkeit repetiert werden.
- Das H2.-Modul bietet zudem digitale Ausgänge zum Anschluss von Anzeige-Modulen.

Programmierung

Für die Eingabe der verschiedenen Bewegungsparameter und der Fahrbefehle wird eine Bibliothek mit Software-Funktionsbausteinen zur Verfügung gestellt. Der Positionierungsvorgang wird durch Aufruf einzelner Funktionsblöcke ausgelöst. Diese sind so gestaltet, dass sie sowohl von einem Bloctec- als auch zweckmässiger von einem Graftec-Programm aufgerufen werden können.

Typenübersicht

| Bezeichnung | Anzahl Funktionseinheiten |
|-------------|-----------------------------|
| PCD4.H210 | 1 Schrittmotoransteuerung |
| PCD4.H220 | 2 Schrittmotoransteuerungen |

2. Technische Daten

Schrittmotorprozessor

| | |
|-----------------------------|-----------------------------------------------------|
| Maximale Positionierdistanz | 16'777'215 Schritte ($2^{24} - 1$) |
| Maximale Rampenschrittzahl | 65'535 Schritte ($2^{16} - 1$) |
| Maximalfrequenz im Hochlauf | 20'000 Hz |
| Start/Stop-Frequenzbereich | 33 - 10'000 Hz |
| Beschleunigung | 1 kHz/sec bis 1000 kHz/sec (Auflösung 1 kHz/sec) |

Digitale Eingänge

Pro Schrittmotoransteuerung:
2 Endschalter und
1 Referenzschalter an 24VDC

| | |
|-----------------------|-----------------------------------|
| Signalpegel | low = 0 .. 4V high = 19 .. 32V |
| Eingangsstrom bei 24V | 10mA |
| Betriebsart | Quellbetrieb |
| Eingangsverzögerung | < 1 msec |

Digitale Ausgänge

| | |
|------------------------------------|-------------------------------------------------------------------------------|
| Schrittpulse (PUL) | aktiv high |
| Pulsbreite für das Schrittsignal | 16.5 usec |
| Auflösung aller Zeitintervalle | 0.5 usec |
| Richtungssignal (DIR) | low = vorwärts (cw) high = rückwärts (ccw) |
| Bewegungssignal (MOT) | high = Motor in Bewegung |
| Signalpegel (für PUL, DIR, MOT) | low = 0 .. 4VDC high = 5 .. 24V (je nach Logikspannung der Leistungsstufe) |
| Belastung pro Ausgang | < 20 mA |
| Betriebsart | Senkbetrieb (der Minus wird geschaltet) |

Data, Clock und Enable für PCA2.D14

| | |
|------------------|--------------------------------------------|
| Spannungsbereich | 19 .. 32V DC geglättet |
| Spannungsabfall | < 0.5V |
| Ausgangsstrom | 1 .. 100mA (nicht kurzschlussfest) |
| Betriebsart | Quellbetrieb (der Plus wird geschaltet) |

Stromversorgung

Extern (Anwender) + 24V DC (19 .. 32V) geglättet
Welligkeit 10%

Stromaufnahme ab externer
24V-Speisung 50mA

Stromaufnahme für Signal-
ausgänge (PUL, DIR, MOT) 40mA

Intern ab PCD4-Bus (+5V)

H210 150mA

H220 200mA

Betriebsbedingungen

Umgebungstemperatur 0°C .. +50°C
ohne Zwangsbelüftung

Störfestigkeit 1kV mit kapazitiver Kopplung
nach IEC 801-4

Mechanische Festigkeit Nach IEC 65A

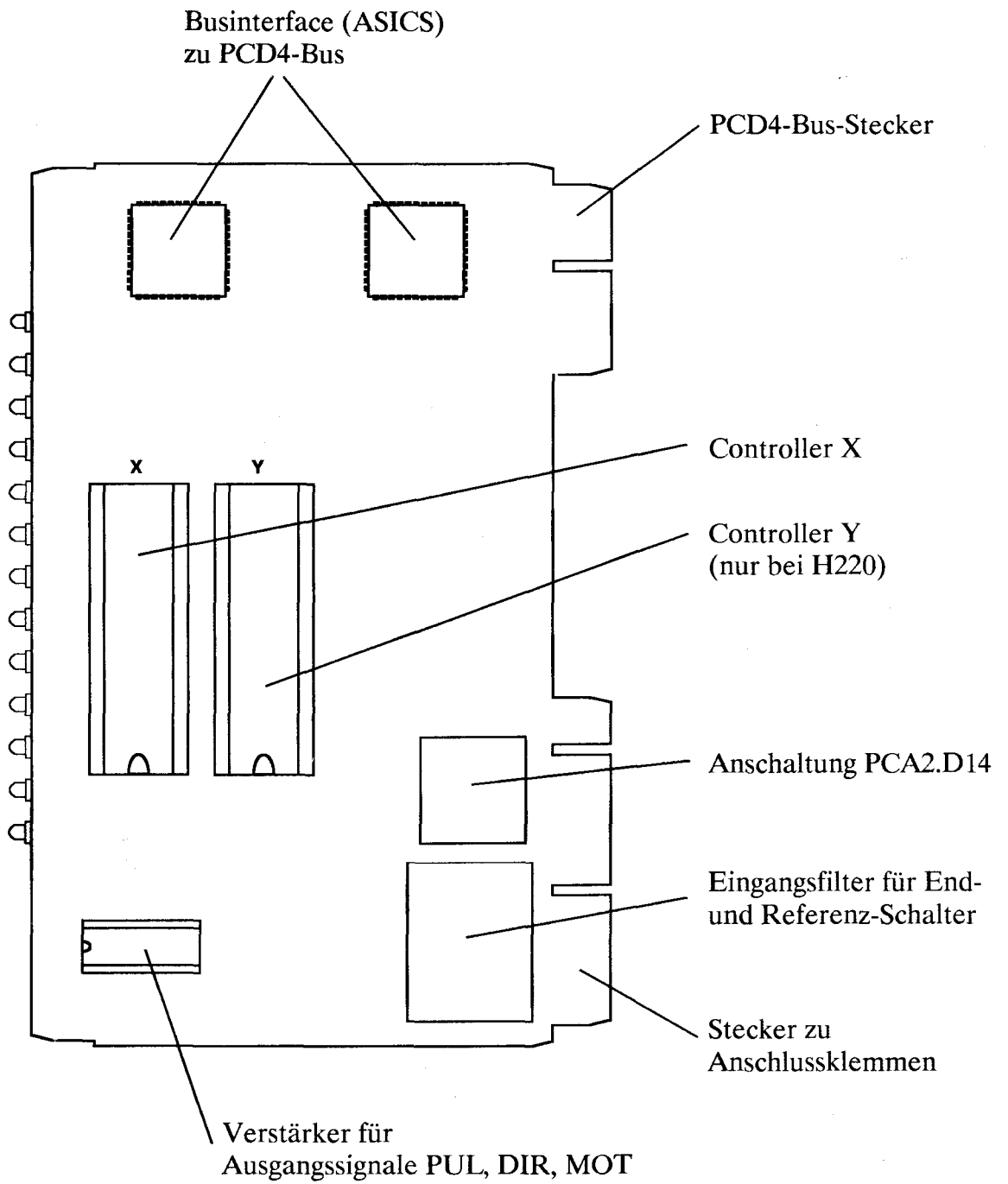
Lagerbedingungen Temperatur: -20°C .. +85°C
Luftfeuchtigkeit: 0 .. 95%

Programmierung

Mit Funktionsbausteinen,
basierend auf
PCD-Anwenderprogramm

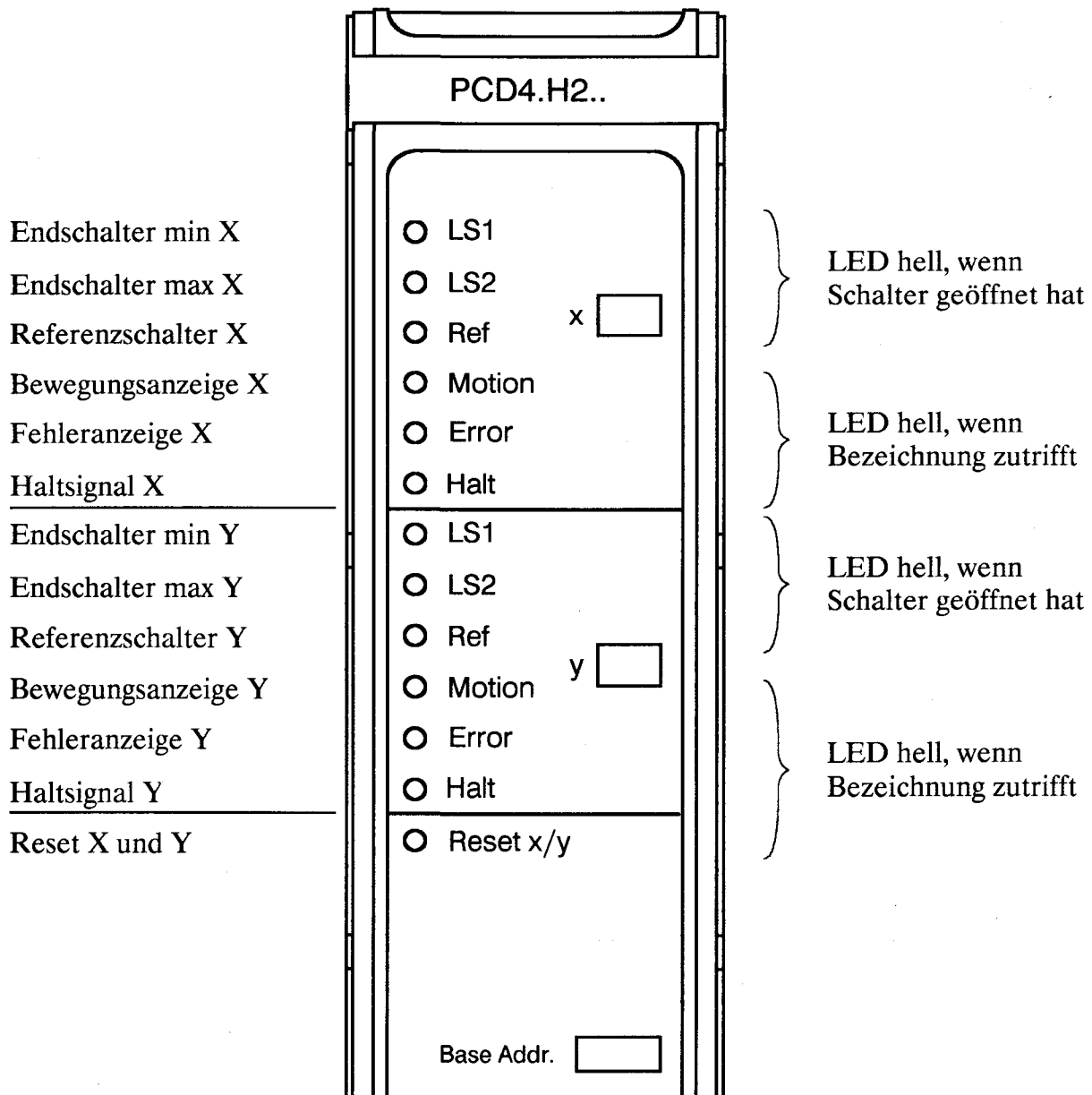
3. Präsentation

3.1 Leiterplatte



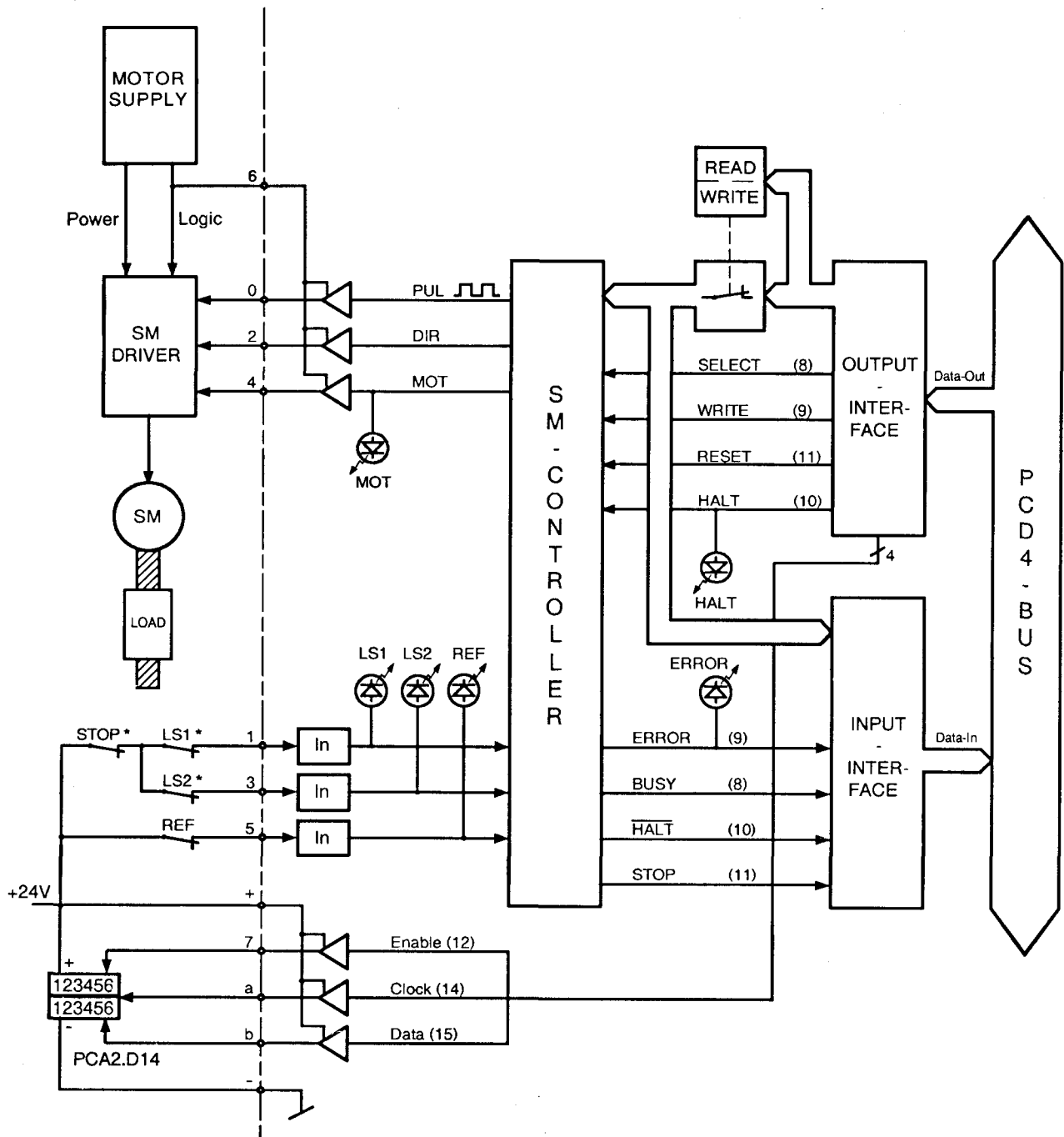
3.2 Frontplatte

(Typ PCD4.H220)



4. Blockschaltbild

Dargestellt ist nur die X-Achse

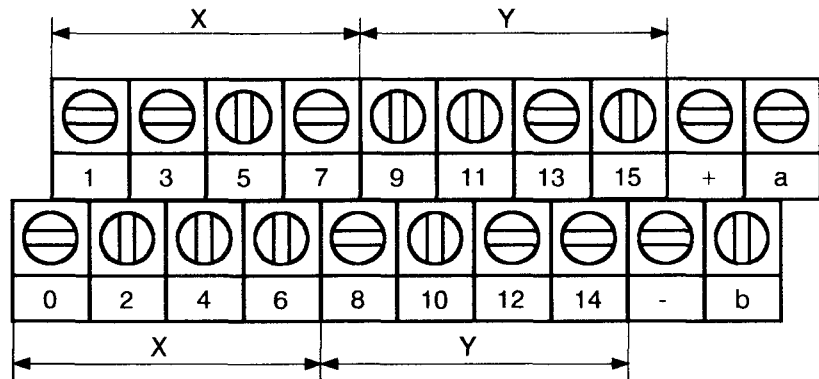


*) Bezüglich Sicherheitsaspekte siehe Abschnitt "Stopschalter, Endschalter und Referenzschalter"

(8 ... 15) Schreib- bzw. lesbare Adressen bezogen auf Basisadresse 0

5. Anschlüsse und Adressierung

5.1 Anschlussklemmen (auf Bus-Modul)



Anschlussklemmen X/Y

| Funktion | | Achse X | Achse Y |
|------------------------------------|-----|---------|---------|
| Endschalter min | LS1 | 1 | 9 |
| Endschalter max | LS2 | 3 | 11 |
| Referenzschalter | REF | 5 | 13 |
| Schrittpulse | PUL | 0 | 8 |
| Drehrichtung | DIR | 2 | 10 |
| Bewegungssignal | MOT | 4 | 12 |
| Speisung der Signalausgänge | | 6 | 14 |
| Enable-Signal für PCA2.D14 | EN | 7 | 15 |
| Gemeinsame Anschlüsse | | | |
| Speisung der Ausgänge für PCA2.D14 | | | + |
| 0V-Anschluss, Masse, Erdung | | | - |
| Clock-Signal für PCA2.D14 | | | a |
| Data-Signal für PCA2.D14 | | | b |

5.2 Stopschalter, Endschalter und Referenzschalter

Der Anschluss für die Endschalter (LS1 und LS2) sowie für den Referenzschalter (REF) erfolgt unter 24V DC im Quellbetrieb. Die Signale dieser Schalter werden autonom vom SM-Controller überwacht, sind also vom Anwenderprogramm unabhängig.

Dennoch dürfen die Endschalter (LS1/2) und ein allenfalls in Serie geschalteter Stop-Schalter keine Abschaltfunktionen im Sinne von **Sicherheitsvorschriften** übernehmen. Dafür sind zusätzliche Sicherheits-Endschalter und Notstop-Schalter vorzusehen, welche direkt auf die Hauptstromkreise der Antriebe wirken.

5.3 Installation der H2.. - Module

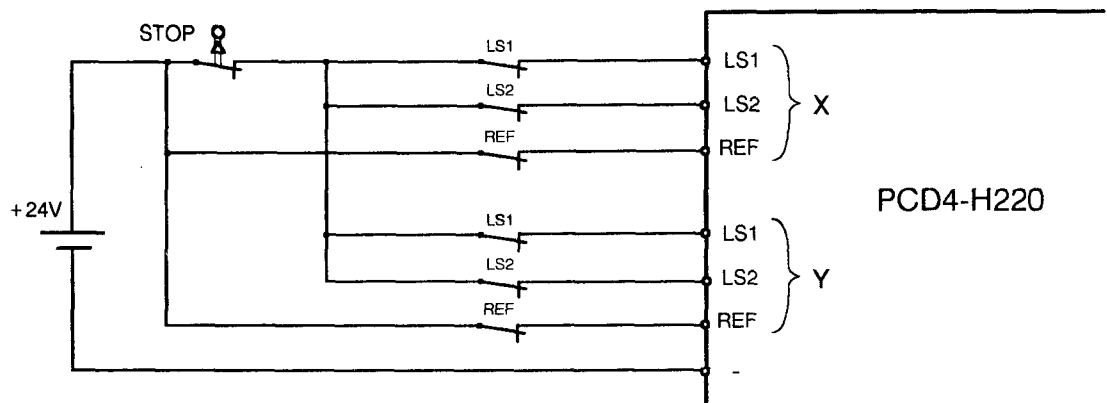
Die Schrittmotormodule PCD4.H2.. können an beliebiger Stelle auf den Busmodulen angeordnet werden. Es erweist sich als Vorteil für den Anwender, die Module auf ihrer Frontseite sofort mit Basis-Adresse und Achsennummer (1,2 usw.) zu bezeichnen.

Eingänge

In den Anwendungen von Schrittmotoren, bei welchen Endschalter und Referenzschalter nicht benötigt werden, **müssen die unbenutzten Eingänge mit +24V geglättet belegt werden**. Dies gilt auch für die Eingänge des Y-Systems wenn dieses nicht gebraucht wird und ebenso für das Modul PCD4.H210.

Die Endschalter LS1 und LS2 sowie der Referenzschalter REF sind in unbetätigtem Zustand (inaktiv) geschlossen und belegen die Eingänge mit +24V. Es ist zu beachten, dass sie mit Gleichspannung versorgt werden die geglättet ist (siehe technische Daten) weil die Eingangsschaltungen so ausgelegt sind, dass sie ein Anhalten unverzögert, d.h. schrittgenau ermöglichen (Zeitkonstante des Eingangsfilters < 1ms).

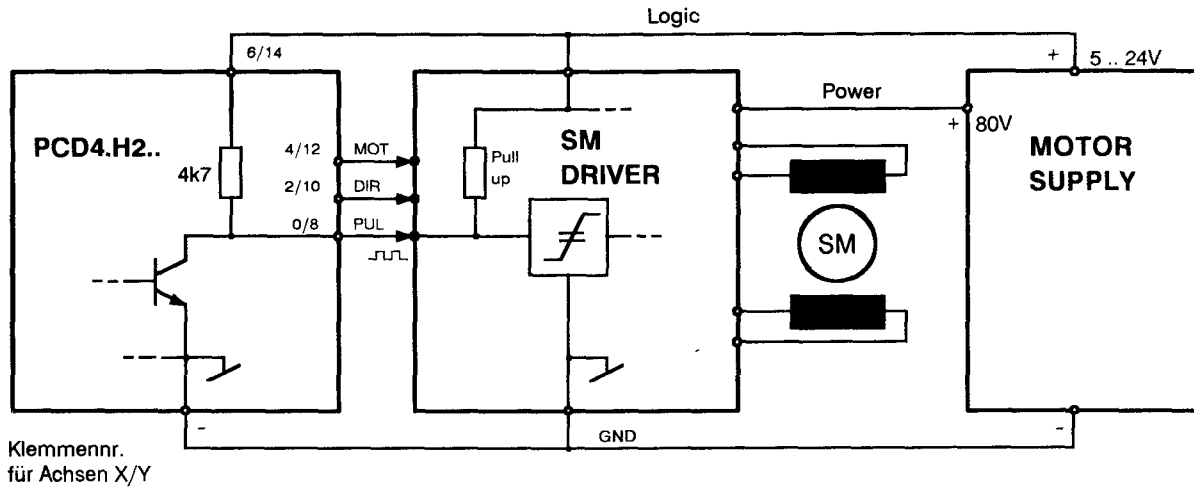
Anschlusschema der Eingänge:



Bei der Betätigung der Taste "Stop" werden LS1 und LS2 gleichzeitig stromlos. Dieser spezielle Zustand bewirkt im SM-Controller ein sofortiges Stoppen des Motors (ohne Verzögerungsrampe). Der Stop führt meistens zu einem Positionsverlust und kann nur durch neue Initialisierung des Systems gelöst werden.

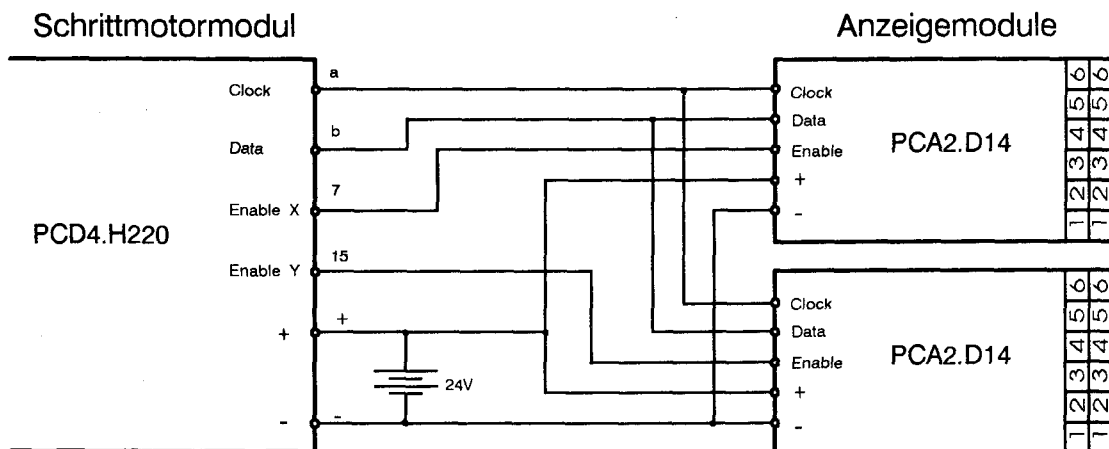
Ausgänge für Motorsignale

Die Signalausgänge PUL, DIR und MOT zur Ansteuerung einer SM-Leistungsstufe sind weitgehend anpassungsfähig. Das folgende Schema beschreibt eine Möglichkeit der Anschaltung an den Eingang einer handelsüblichen Leistungsstufe (SM-DRIVER).





Die Leistungsstufen für Schrittmotoren werden immer mit einer Motorspannung (20 .. 150V DC) und einer Logikspannung (5 .. 24V DC) versorgt. Es ist nun sehr wichtig, dass die Logikspannung der Leistungsstufe auch an die Ausgangsschaltung des H2..-Moduls, d.h. auf die Klemmen 6 und 14 geführt wird. Nebst den Signalen für Takt (PUL) und Drehrichtung (DIR) erzeugt das H2..-Modul auch ein Signal für die Bewegungsanzeige (MOT = Motion). Dieses Signal wird häufig dazu benutzt, um bei Stillstand des Schrittmotors im Leistungsenteil eine Stromabsenkung zu erwirken.

Ausgänge für Anzeigemodul PCA2.D14



5.4 Adressierung

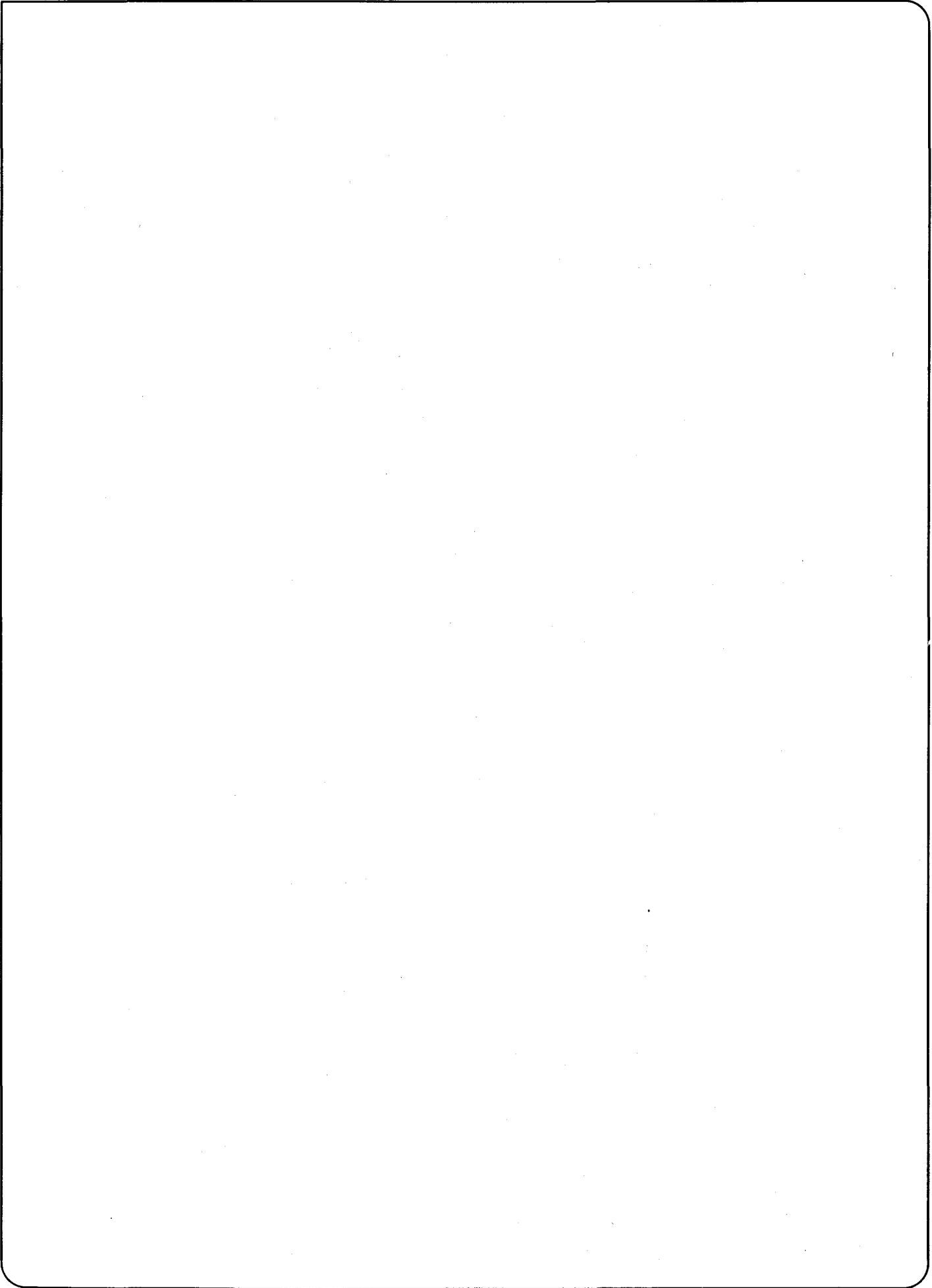
Das Modul belegt 16 Adressen am PCD4-Bus.
 Im Gegensatz zu einem gewöhnlichen Ausgangsmodul haben beim Schrittmotormodul die gelesenen Adressen (Data-In) nicht die gleiche Bedeutung wie die geschriebenen (Data-Out).

| Data-In (lesen) | Data-Out (schreiben) | | |
|----------------------------|---------------------------------------------------------------------------------------------------------|-----------------|-------------------------------|
| | Automatik | Manuell | |
| 0 Data-Bus (LSB) | 0 Data-Bus (LSB) | -- | } versch. manuelle Funktionen |
| 1 " " | 1 " " | MO | |
| 2 " " | 2 " " | M1 | |
| 3 " " | 3 " " | -- | |
| 4 " " | 4 " " | M2 | |
| 5 " " | 5 " " | FOR (vorwärts) | |
| 6 " " | 6 " " | REV (rückwärts) | |
| 7 Data-Bus (MSB) | 7 Data-Bus (MSB) | -- | |
| 8 * BUSY , low = Busy | 8* SELECT XY , low = X , high = Y | | |
| 9 * ERROR X , low = Error | 9 Schreibbefehl ,  | | |
| 10 * HALT X , high = Halt | 10 Haltbefehl,  | | |
| 11 * STOP X , low = Stop | 11 Reset , low = Reset X u. Y | | |
| 12 nicht benützt | 12* Enable D14 x | } aktiv tief | |
| 13 * ERROR Y , low = Error | 13* Enable D14 y | | |
| 14 * HALT Y , high = Halt | 14* Clock D14 | | |
| 15 * STOP Y , low = Stop | 15* DataD14 | | |

Angegebene Adressen sind relativ
 Absolutadresse = Modul-Basisadresse + Relativadresse

Für den Anwender sind die mit * bezeichneten Adressen von Interesse. Die lesbaren Adressen (wie z.B. das BUSY-Signal) können bei der Inbetriebnahme mit dem Debugger oder dem Servicegerät P100 überwacht werden. Alle anderen Adressen werden ausschliesslich von Funktionsblöcken benützt.

Das Enable-Signal für die PCA2.D14 Anzeigen ist aktiv tief. Auf dem Schrittmotormodul befindet sich ein Inverter für diese Ausgänge.



6. Programmerstellung für das H2-Modul

6.1 Installation der Software

Dem Benutzer des H2-Moduls steht das Softwarepaket **PCD9.H2FB.E1** zur Programmierung des H2-Moduls zur Verfügung. Das Paket enthält Funktionsbausteine, welche auf PCD-Anwenderprogramm basieren. Durch Aufrufen dieser Funktionsbausteine im Anwenderprogramm können sämtliche Funktionen des H2-Moduls aktiviert werden. Das Paket enthält folgende zwei Dateien:

H2DEF.SRC Diese Datei enthält die Liste der Ressourcen, die wenn sie in das Anwenderprogramm eingebunden werden, den Zugang zu den Funktionsbausteinen ermöglicht.

H2FB.SRC In dieser Datei sind alle Funktionsbausteine enthalten, welche das Installieren, das Parametrieren sowie das Betreiben des H2-Moduls durch einfaches Aufrufen von FB's (Befehlen) erlaubt.

Das Gesamtpaket hat folgenden Umfang:

- Anzahl Programmzeilen < 600
- Benutzte FB-Ebenen 3

6.2 Anweisungen für den Einsatz der Funktionsbausteine

Für einen problemlosen Einsatz des H2-Softwarepakets ist es von Vorteil, die nachfolgenden Schritte in dieser Reihenfolge durchzuführen :

Definition der Ressourcen

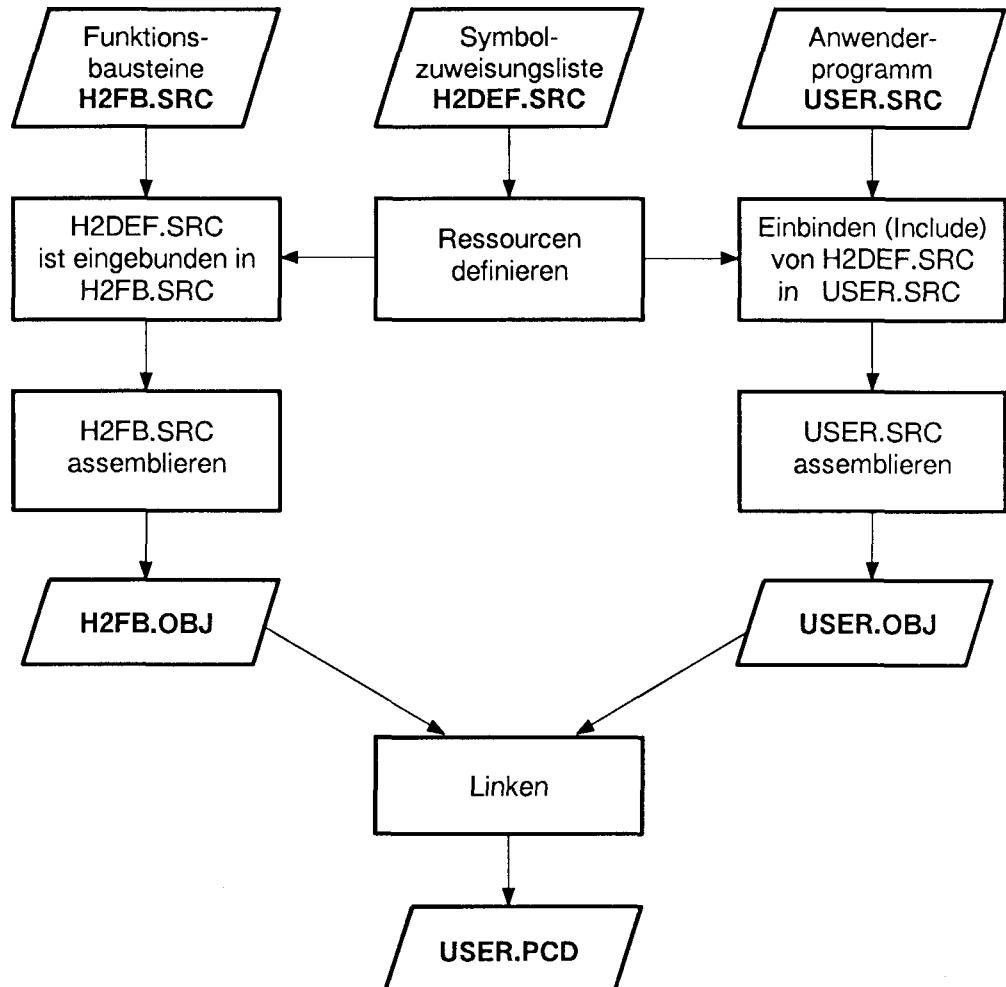
Sämtliche in der Datei H2DEF.SRC enthaltenen Symbole sind zu definieren. Aus Gründen der Syntax, sind einem Teil der Ressourcen schon Adressen zugewiesen worden. Der Anwender kann diese aber neu definieren und damit seinem Programm anpassen.

Einbinden der Ressourcen in das Anwenderprogramm

Die Datei H2DEF.SRC wird in die Quelldatei des Anwenderprogrammes mittels \$Include eingebunden. Das Betreiben der Schrittmotoren erfolgt nun ausschliesslich durch Aufrufen der in H2DEF bezeichneten Funktionsbausteine.

Zusammenfügen der Dateien

Das nachstehende Diagramm zeigt, wie die H2-Dateien und das Anwenderprogramm assembliert und die resultierenden Objekt-Dateien zu einer Gesamtdatei gelinkt werden.



6.3 Definition der Symbole in H2DEF.SRC

Merker

| | | |
|----------|------|------------------------------------------|
| TRAFLAG | F 0 | ; TEMPORARY TRANSFER FLAGS (8 FLAGS) |
| XYFLAG | F 8 | ; X/Y SELECTION FLAG |
| COMERR | F 9 | ; TIMING ERROR WITH BUSY |
| BUSYFLAG | F 10 | ; COMMON FLAG FOR BUSY ("0"->BUSY) |
| DPLM | F 11 | ; DISPLAY MODE ("0"->10, "1"->2*6DIGITS) |
| FH_0 | F 12 | ; FLAG HELP (12 DIGIT = 48 BIT) |

Für TRAFLAG ist ein Merkerfeld von 8 Merkern, für XYFLAG, ERROR, BUSYFLAG und DPLM je 1 Merker und für FH_0 ein Feld von 48 Merkern zu reservieren. Anstelle der oben verwendeten Merker-Adressen können beliebige andere 60 Adressen verwendet werden.

Register

Parameterblock

Im Parameterblock bestehend aus 8 aufeinanderfolgenden Registern sind die Daten enthalten, die für das Betreiben eines Schrittmotors erforderlich sind.

Das Schreiben und Lesen der Daten im Parameterblock erfolgt ausschliesslich indexiert. Das Indexregister enthält die unterste Adresse eines Blockes und wird symbolisch mit dem Namen des Motor-Systems bezeichnet (z.B. M1, M2 ... Mn).

Beispiel für die Definition der Parameterblöcke für 2 Module PCD4.H220 (4 Motoren) :

| | | |
|----|------|--------------------------------------|
| M1 | K 1 | ; PARAMETER MOTOR 1 (BLOCK OF 8 REG) |
| M2 | K 9 | ; PARAMETER MOTOR 2 (BLOCK OF 8 REG) |
| M3 | K 17 | ; PARAMETER MOTOR 3 (BLOCK OF 8 REG) |
| M4 | K 25 | ; PARAMETER MOTOR 4 (BLOCK OF 8 REG) |

Organisation des Parameterblocks

(Indexregister enthält niedrigste Adresse des Registerblocks)

| | | | |
|---|--------|-----------------------------------------|---------------------------------|
| 0 | R0 (i) | +0 : Basis-Adresse des H2-Moduls | |
| 1 | | +1 : Motorwahl X-Motor:"0", Y-Motor:"1" | |
| 2 | | +2 : Start/Stop-Frequenz | in Hz , 33-10'000 |
| 3 | | +3 : Hochlauf-Frequenz (mit Rampen) | in Hz , 33-20'000 |
| 4 | | +4 : Beschleunigung | in kHz/sec , 1-1'000 |
| 5 | | +5 : Positionierdistanz | in Schritten, 0-2 ²⁴ |
| 6 | | +6 : Istposition | in Schritten, 0-2 ²⁴ |
| 7 | | +7 : Status (Diagnose) | |

Beispiel für die Eingabe einer Positionerdistanz für M3:

```

"
"
SEI      M 3      ; Adresse des Parameterblocks
PUTX     R        ; Resultat der errechneten Position
          R0+5    ;
"
"
    
```

Zahlenbereich

Alle Zahlenwerte für die Parameter- und Positioniereingabe beschränken sich auf den Bereich der positiven ganzen Zahlen.

Arbeitsregisterblock

Der Arbeitsregisterblock dient als Datenschnittstelle zwischen den Parameterblöcken und den H2-Modulen. Diese 8 Register werden ausschliesslich von den Funktionsbausteinen gelesen und geschrieben. Einzig die Zuweisung einer freien Adresse für jedes Register ist dem Anwender überlassen.

```

BASE     R 33    ; BASE ADDRESS OF H2-MODUL
XY       R 34    ; SELECT X-CONTROLLER OR Y-CONTROLLER
SSF      R 35    ; START/STOP FREQUENCY
HF       R 36    ; HIGH FREQUENCY (WITH RAMP)
ACCEL    R 37    ; ACCELERATION (RAMP)
POS      R 38    ; POSITION
APOS     R 39    ; ACTUAL POSITION
STATUS   R 40    ; STATUS
    
```

Übrige Register und Zähler

Für folgende Symbole wird je ein freies Register bzw. ein freier Zähler zugewiesen.

```

TRAREG   R 41    ; TRANSFER REGISTER
RINDEX   R 42    ; TEMPORARY INDEX REGISTER
TEMPREG  R 43    ; TEMPORARY HELP REGISTER
RH_0     R 44    ; REGISTER HELP
REG_1    R 45    ; VALUE DISPLAYED FIRST LINE D14
REG_2    R 46    ; VALUE DISPLAYED SECOND LINE D14
CH_0     C 100   ; COUNTER HELP
    
```

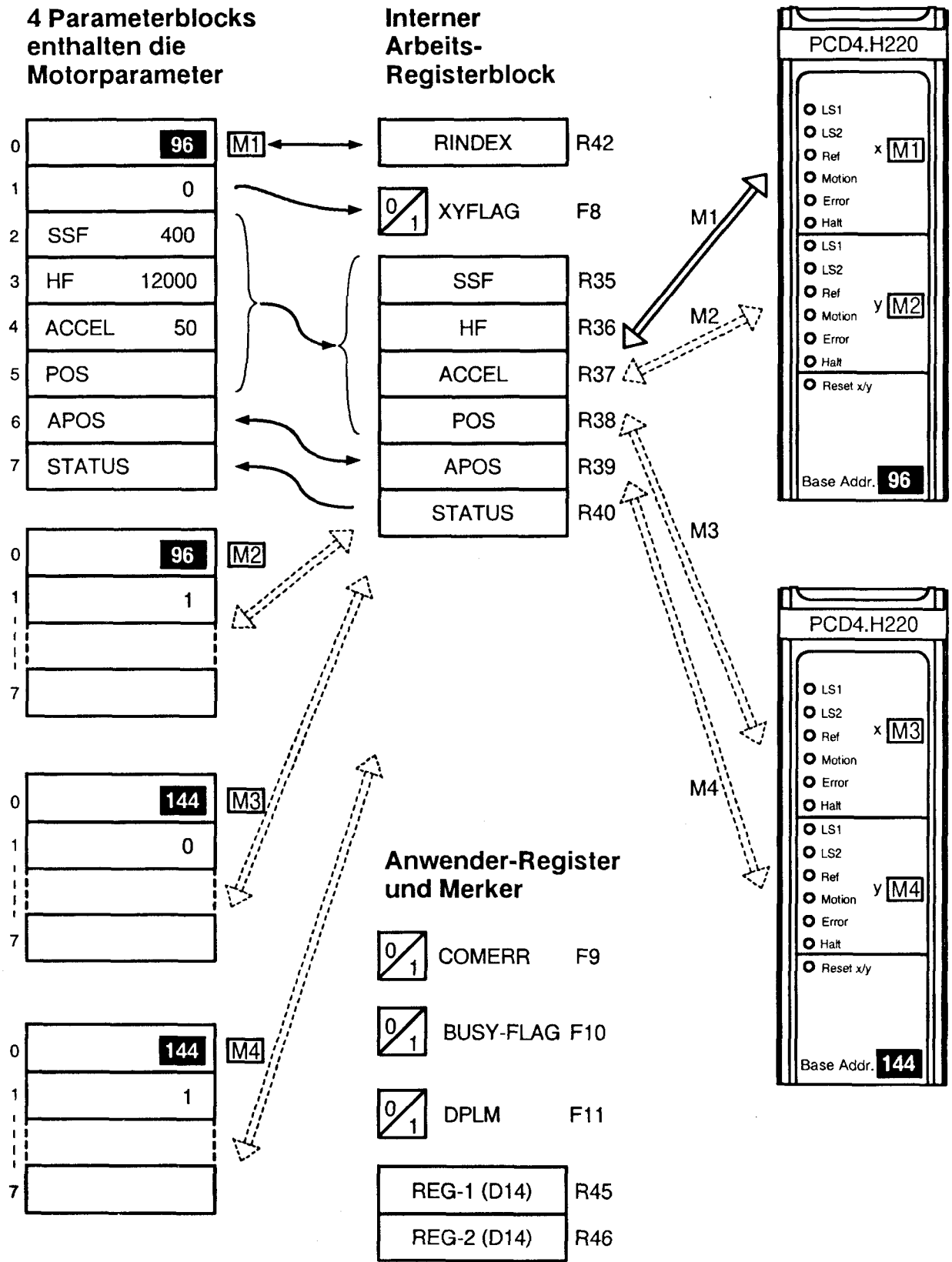
Definition der Funktionsbausteine (Befehle)

Die Funktionsbausteine mit * werden als Unterprogramme verwendet und sind dem Anwender nicht zugänglich.

| | | |
|----------|--------|---------------------------------------|
| RSMC | FB 800 | ;RESET STEPPING MOTOR CONTROLLER |
| LVP | FB 801 | ;LOAD VELOCITY PROFILE |
| CAV | FB 802 | ;COMPUTE ACCURATE VELOCITY |
| LAPOS | FB 803 | ;LOAD ACTUAL POSITION |
| LDEST | FB 804 | ;LOAD DESTINATION ABSOLUTE |
| LDF | FB 805 | ;LOAD DIFFERENCE FORWARD |
| LDB | FB 806 | ;LOAD DIFFERENCE BACKWARDS |
| SISF | FB 807 | ;SINGLE STEP FORWARD |
| SISB | FB 808 | ;SINGLE STEP BACKWARDS |
| PSSF | FB 809 | ;POSITIONING WITH STARTSTOP FREQUENCY |
| PHF | FB 810 | ;POSITIONING WITH HIGH FREQUENCY |
| REF | FB 811 | ;MOVE TO REFERENCE SWITCH |
| REFLS | FB 812 | ;MOVE TO REF. THEN TO LIMIT SWITCH |
| LSREF | FB 813 | ;MOVE TO LIMIT THEN TO REF SWITCH |
| OUTLS | FB 814 | ;MOVE OUT OF LIMIT SWITCH |
| OUTOR | FB 815 | ;MOVE OUT OF OVERRUN |
| RBUSY | FB 816 | ;READ BUSY STATUS |
| RAPOS | FB 817 | ;READ ACTUAL POSITION |
| RSTATUS | FB 818 | ;READ STATUS |
| DAPOS | FB 819 | ;DISPLAY ACTUAL POSITION |
| DPB14 | FB 820 | ;DISPLAY BLANK ON PCA2.D14 |
| DP14 | FB 821 | ;DISPLAY ON PCA2.D14 |
| MANCON | FB 822 | ;SWITCH TO MANUAL CONTROL |
| AUTOCON | FB 823 | ;SWITCH TO AUTOMATIC CONTROL |
| HLTM | FB 824 | ;HALT ANY MOTION |
| CONT | FB 825 | ;RESET HALT CONDITION |
| OSF | FB 826 | ;ONE STEP FORWARD |
| OSB | FB 827 | ;ONE STEP BACKWARDS |
| SSFF | FB 828 | ;MANUAL WITH SSF FORWARD |
| SSFB | FB 829 | ;MANUAL WITH SSF BACKWARDS |
| HFH | FB 830 | ;MANUAL WITH HF FORWARD |
| HFB | FB 831 | ;MANUAL WITH HF BACKWARDS |
| * WBYTE | FB 832 | ;TRANSFER BYTE TO SM CONTROLLER |
| * RBYTE | FB 833 | ;READ ONE BYTE FROM CONTROLLER |
| * RDATA | FB 834 | ;READ DATA FROM CONTROLLER |
| * SWPULS | FB 835 | ;SEND WRITE PULS TO CONTROLLER |
| * WDATA3 | FB 836 | ;TRANSFER DATA TO SM CONT (3 BYTES) |
| * CONTAD | FB 837 | ;GET ADDRESS OF SM CONTROLLER |
| * SETAD | FB 838 | ;SET UP CONTROLLER ADDRESS |
| * INSTR | FB 839 | ;INSTRUCTION TO SM CONTROLLER |
| * RBO | FB 840 | ;REGISTER BIT OUT |
| * WDATA2 | FB 841 | ;TRANSFER DATA TO SM CONT (2 BYTES) |

6.4 Organisation des Datenverkehrs zwischen CPU und H2-Modulen

Beispiel: 2 Module PCD4.H220 auf Basis-Adressen 96 und 144

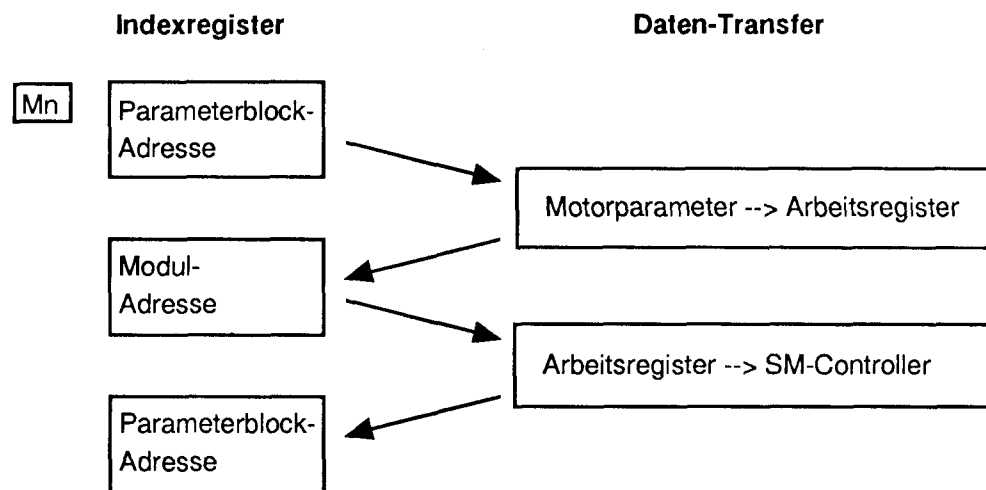


Aufbau der Funktionsblöcke

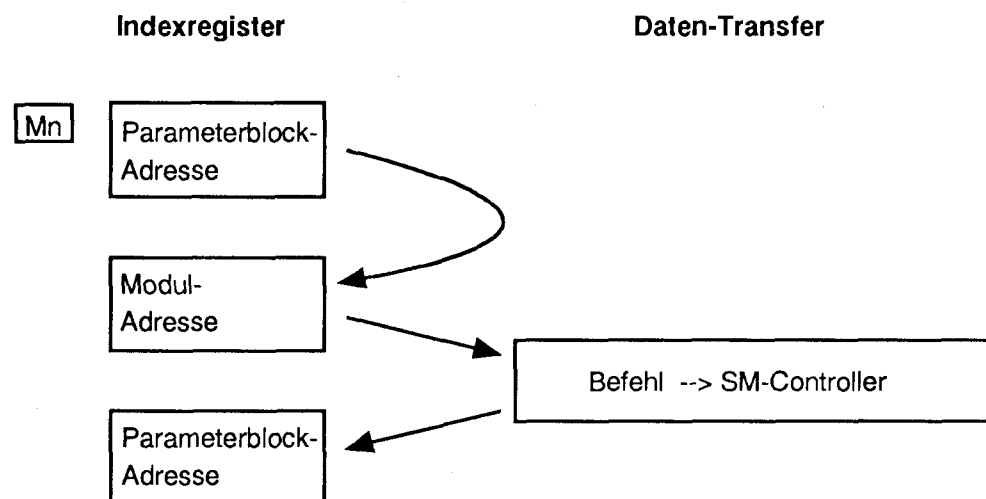
Die Kommunikation zu den H2-Modulen erfolgt ausschliesslich mittels Funktionsblöcken, die mit einem symbolischen Namen aufgerufen werden. Diese Funktionsblöcke sind sich im Aufbau alle sehr ähnlich und erfüllen eine definierte Aufgabe. Somit werden sie als Funktionsbausteine und ihr Aufruf als Befehl bezeichnet.

Grundsätzlicher Programmaufbau in den Funktionsbausteinen

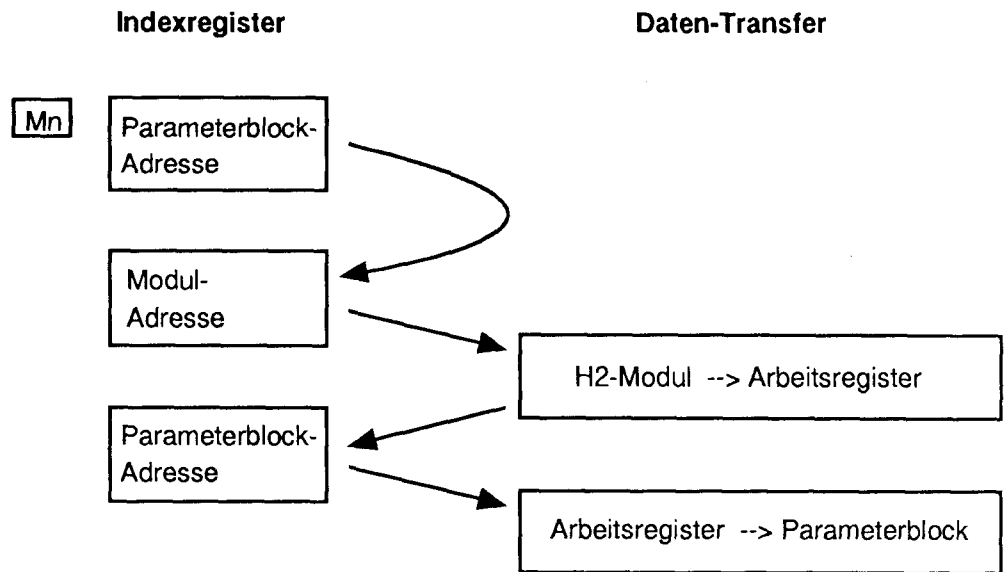
Datentransfer CPU --> H2-Modul



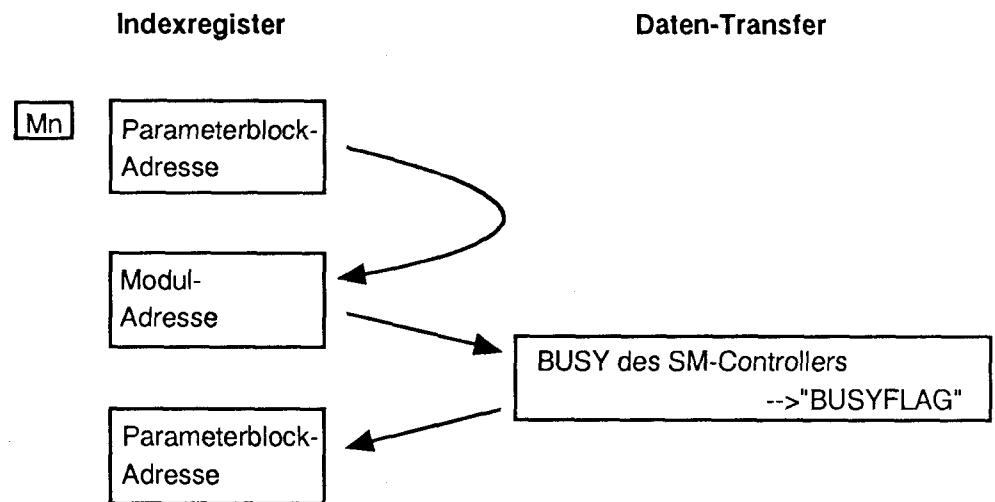
Befehlsausgabe



Datentransfer H2-Modul --> CPU



Lesen des "BUSY" - Zustandes eines Motorsystems



6.5 Betriebsarten

Automatikbetrieb

Im Automatikbetrieb führt der SM-Controller alle ihm zugeteilten Befehle selbstständig aus. Bei der Initialisierung wird das Modul für den Automatikbetrieb vorbereitet, weil unabhängig von der Betriebsart der Motor zuerst mit Parametern versorgt wird, was nur im Automatikbetrieb möglich ist.

Während der Abarbeitung eines Befehls und während des Laufens eines Motors steht das Busy-Signal des entsprechenden SM-Controllers auf "0". Das Ende einer Befehlsausführung kann mit dem Funktionsbaustein "RBUSY" abgefragt werden, welcher den Merker "BUSYFLAG" entsprechend setzt. Die Ausgabe des nächsten Schrittmotorbefehls kann erst dann erfolgen, wenn sichergestellt ist, dass das "BUSYFLAG" = "1" ist.

Handbetrieb

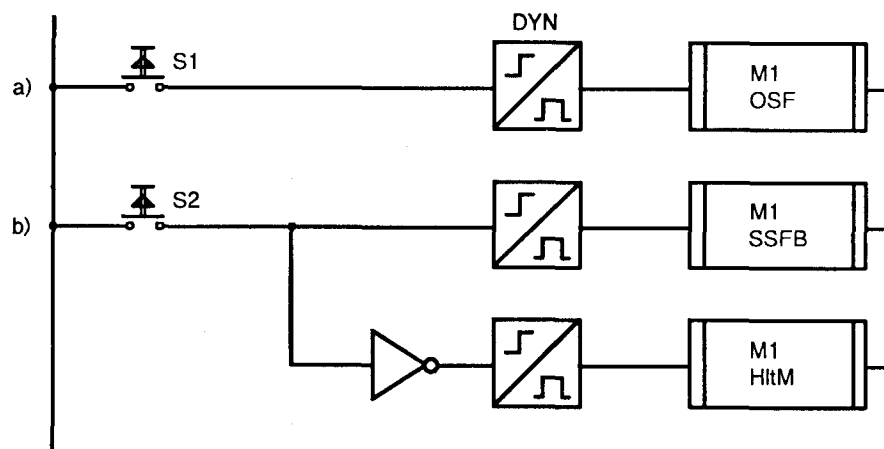
Im Handbetrieb wird das Positionieren von Hand gesteuert. Die Bewegungen werden durch Betätigung von Tasten auf einer Bedienkonsole ausgelöst und dauern solange an, bis sie mit einem Halt-Befehl beendet werden.

Der Halt-Befehl kann durch die Rückstellung der Bewegungstaste ausgegeben werden, wie in nachfolgendem Beispiel gezeigt wird.

Programmbeispiel für Handbetrieb

(Umlaufprogramm)

- Beim Schliessen des Schalters S1 wird bei jeder Schaltbetätigung 1 Schritt vorwärts ausgeführt.
- Beim Schliessen des Schalters S2 wird rückwärts mit Start/Stop-Frequenz gefahren solange der Schalter S2 betätigt bleibt.



c) Herausfahren aus dem Endschalterbereich

Vor allem im Handbetrieb wird häufig auf Endschalter aufgefahren, welche die Bewegung anhalten. Der Befehl "OutLS" ist aber im Satz der Automatikbefehle enthalten.

Das folgende Beispiel in AWL ermöglicht M2 aus dem Endschalterbereich herauszufahren.

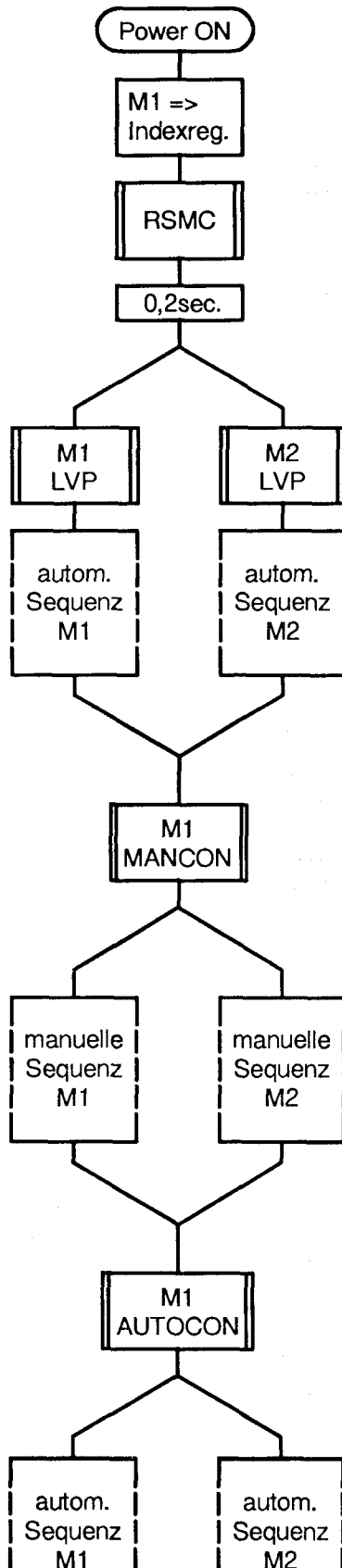
```

"
"
STH      I 1           ;Befehlstaste für OUTLS
DYN      F 500        ;Dynamisierung des Befehls
SET      F 501        ;M für Rückschalten auf Hand
SEI      M2           ;Wahl des Motors
CFB      H AUTOCON    ;Hand => Automatik
CFB      H OUTLS      ;Motorbefehl
STH      F 501        ;\
CFB      H RBUSY      ; | Rückschalten auf Hand-
ANH      F BUSYFLAG   ; > betrieb, wenn "OUTLS"
CFB      H MANCON     ; | beendet ist
RES      F 501        ;/
"
"

```

Initialisierung des H2-Moduls und Umschaltung der Betriebsarten

z.B. für das Modul zu den Motoren M1 und M2



Modulvorwahl

Mit "RSMC" werden beide Systeme X und Y eines Moduls bereitgestellt.

min. 0,1sec warten

Als nächster Schritt ist die Übergabe der Motorparameter erforderlich.

Die automatischen Sequenzen können jetzt gefahren werden.

Die Umschaltung von Automatik zum Handbetrieb erfolgt mit dem Befehl "MANCON" für beide Systeme X und Y gemeinsam.

Die handgesteuerten Sequenzen können jetzt gefahren werden.

Die Umschaltung von Handbetrieb zu Automatik erfolgt mit dem Befehl "AUTOCON" für beide Systeme gemeinsam.

Weiterfahrt in Automatik.

7. Die einzelnen Funktionsbausteine

7.1 Uebersicht der Funktionsbausteine

Funktionsbausteine für den automatischen Betrieb

Initialisierungsbefehle:

| | | |
|-------------|----------------------------------------------------------------|-----|
| RSMC | Initialisierung des H2-Moduls | 7-4 |
| LVP | Parameter eingeben und umrechnen | 7-5 |
| CAV | Parameter umrechnen unter Einhaltung der genauen Soll-Frequenz | 7-6 |

Positioneingaben:

| | | |
|----------------|----------------------------------------|------|
| LAPos * | Istposition eingeben | 7-7 |
| LDest | Sollposition eingeben | 7-8 |
| LDF | Laden der Positionsdifferenz vorwärts | 7-10 |
| LDB | Laden der Positionsdifferenz rückwärts | 7-11 |

Fahrbefehle:

| | | |
|-------------|-------------------------------------------------|------|
| SiSF | Einzelsschritt vorwärts | 7-12 |
| SiSB | Einzelsschritt rückwärts | 7-13 |
| PSSF | Positionieren mit Start/Stop-Frequenz | 7-14 |
| PHF | Positionieren mit Hochlauf-Frequenz (mit Rampe) | 7-15 |

End- und Referenzschalter anfahren:

| | | |
|--------------|----------------------------------------------------------|------|
| Ref | Nullpunkt suchen | 7-16 |
| RefLS | Nullpunkt suchen und aus Endschalterbereich herausfahren | 7-18 |
| LSRef | Endschalter LS1 anfahren und Nullpunkt suchen | 7-20 |
| OutLS | Aus Endschalterbereich herausfahren | 7-22 |
| OutOR | Aus Overrun-Bereich herausfahren | 7-24 |

*) Für den Aufruf der FB's können grosse oder kleine Buchstaben verwendet werden. Um besser lesbar zu sein, werden sie hier in gemischter Schrift dargestellt.

Lesebefehle:

| | | |
|----------------|---------------------------------------------|------|
| RBusy | Busy-Status des SM-Controllers abfragen | 7-26 |
| RAPos | Istposition auslesen | -27 |
| RStatus | Betriebszustand des SM-Controllers auslesen | -28 |

Anzeigebefehle für Modul PCA2.D14:

| | | |
|--------------|------------------------|------|
| DAPos | Istposition anzeigen | 7-30 |
| DPB14 | Anzeige löschen | -31 |
| DP14 | Registerwerte anzeigen | -32 |

Hilfsbefehle:

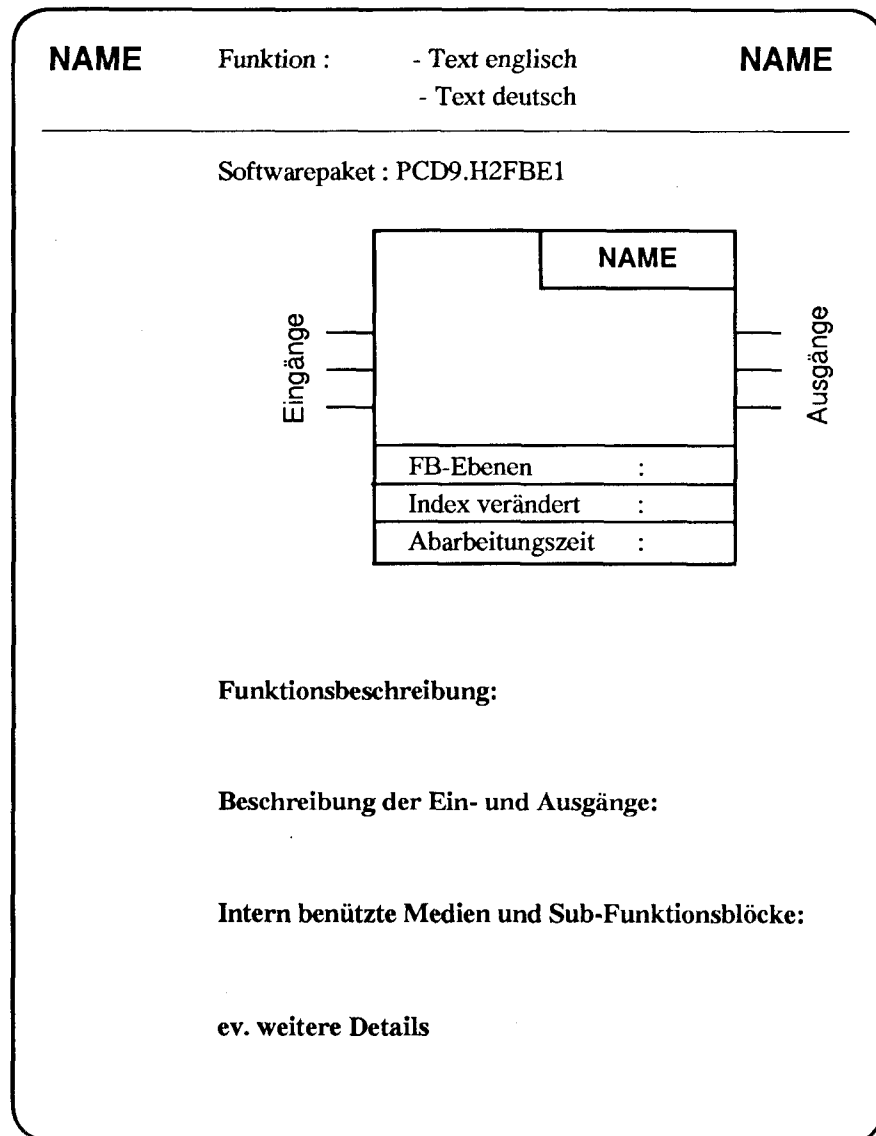
| | | |
|----------------|----------------------------------|------|
| ManCon | Umschaltung auf Handbetrieb | 7-34 |
| AutoCon | Umschaltung auf Automatikbetrieb | -35 |
| HltM | Halt-Befehl | -36 |
| Cont | Freigabe nach Halt | -37 |

Funktionsbausteine für den Handbetrieb**Fahrbefehle:**

| | | |
|-------------|-----------------------------------|------|
| OSF | Einzel-schritt vorwärts | 7-38 |
| OSB | Einzel-schritt rückwärts | -38 |
| SSFF | Mit Start/Stop-Frequenz vorwärts | -39 |
| SSFB | Mit Start/Stop-Frequenz rückwärts | -39 |
| HFF | Mit Hochlauf-Frequenz vorwärts | -40 |
| HFB | Mit Hochlauf-Frequenz rückwärts | -40 |

7.2 Beschreibung der Funktionsbausteine (Befehle)

Um eine gute Übersicht zu bewahren, sind alle folgenden Funktionsbausteine nach dem gleichen Gestaltungsschema aufgebaut:



Es wird im folgenden die gleiche Reihenfolge benützt wie in der Übersicht Abschnitt 7.1.

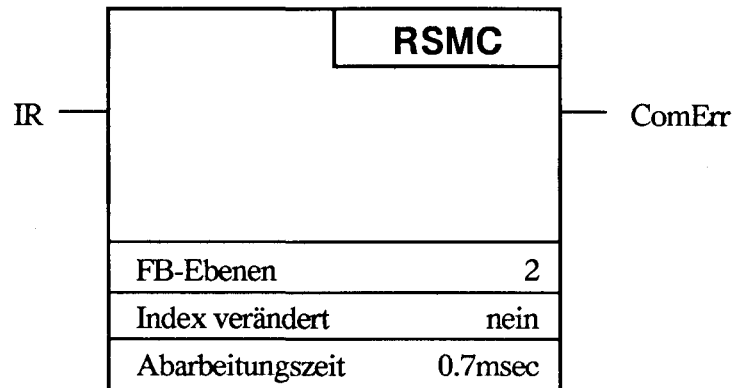
Am Schluss dieses Handbuches finden Sie zudem eine alphabetische Auflistung aller hier benützten Namen und Symbole.

RSMC

Funktion : - Reset Stepping Motor Controller
- Initialisieren des H2-Moduls

RSMC

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Beim Einschalten der PCD4 werden die Schrittmotormodule PCD4.H2xx zwangsläufig rückgestellt und verharren in diesem Zustand. Ein unabsichtliches Fahren der Motoren bei rückgestellten Modulen ist damit ausgeschlossen. Der Befehl "RSMC" macht die X und Y SM-Prozessoren betriebsbereit und prüft auch die Bereitschaft zum empfangen von Befehlen. Kann einer der beiden Controller nicht bereitgestellt werden, so setzt dieser Funktionsbaustein den Merker "ComErr" auf "1". Verläuft hingegen die Initialisierung erfolgreich ab, so bleibt der Merker "ComErr" im Zustand "0".

Nach RSMC muss eine Zeit von mind. 0,1sec abgewartet werden bis zum nächsten Befehl. (Programmierung mit 2 bei Zeitbasis 0,1sec)

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M_1, M_2, \dots, M_n |
| ComErr | Error-Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

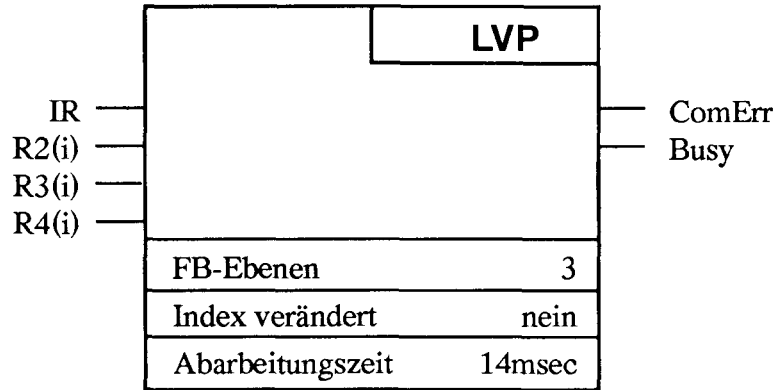
| | |
|----------|----------------|
| FB's | CONTAD, SETAD |
| Register | RINDEX |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG |

LVP

Funktion : - Load Velocity Profile
- Parameter eingeben und umrechnen

LVP

Softwarepaket : PCD9.H2FB E1



Funktionsbeschreibung:

Mit diesem Befehl werden die Bewegungsparameter an den angesprochenen SM-Controller übertragen. Der Befehl "LVP" kann natürlich nur dann erfolgen, wenn der zum Motor gehörende Parameterblock mit den Werten für Start/Stop-Frequenz, Hochlauf-Frequenz und Beschleunigung geladen wurde. Nach der Übertragung der Parameter erfolgt automatisch die Berechnung des Frequenz-Zeit-Profiles. Dieses Profil kann während der Bewegung nicht mehr geändert werden. Eine Änderung der Geschwindigkeit bzw. der Schrittfrequenzen kann somit nur bei Stillstand erfolgen. Werden sämtliche Bewegungen eines Motors mit gleicher Geschwindigkeit gefahren, so muss der Befehl "LVP" nur einmal in der Initialisierungsphase des Antriebes aufgerufen werden.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |
| R2(i) | Parameterblock SSF | | W | Integer | 30 - 10'000 |
| R3(i) | Parameterblock HF | | W | Integer | 30 - 20'000 |
| R4(i) | Parameterblock ACCEL | | W | Integer | 1 - 1'000 |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

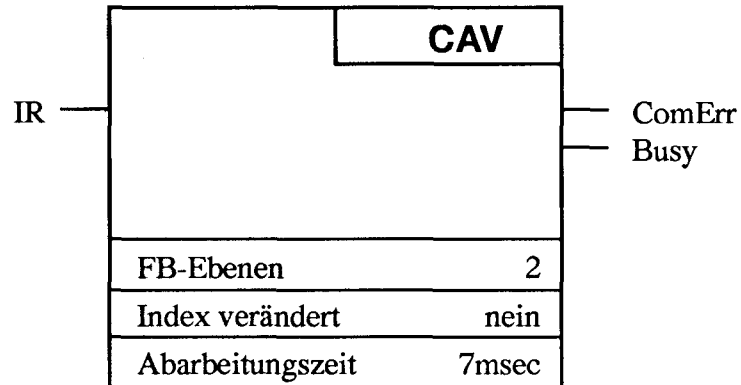
FB's CONTAD, SETAD, WBYTE, WDATA2
 Register Parameterblöcke, TRAREG, SSF, HF, ACCEL, RINDEX
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAF LAG

CAV

Funktion : - Compute Accurate Velocity
 - Parameter umrechnen unter Einhaltung
 der genauen Soll-Frequenz

CAV

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Geschwindigkeitsprofile mit kleinen Rampenschrittzahlen haben die Eigenschaft, dass infolge der Diskretisierungssprünge (die Rampe ist genau genommen treppenförmig) die Hochlauf-Frequenz nicht immer exakt erreicht werden kann. Dort wo dies störend sein sollte, kann mit dem Befehl "CAV" eine alternative Berechnungsroutine durchlaufen werden. Dabei wird mit Priorität die exakte Hochlauf-Frequenz eingehalten und die vorgegebene Beschleunigung auf den nächstfolgenden diskreten Wert korrigiert.

Der Befehl "CAV" ist immer nachfolgend zum Befehl "LVP" anzuwenden.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

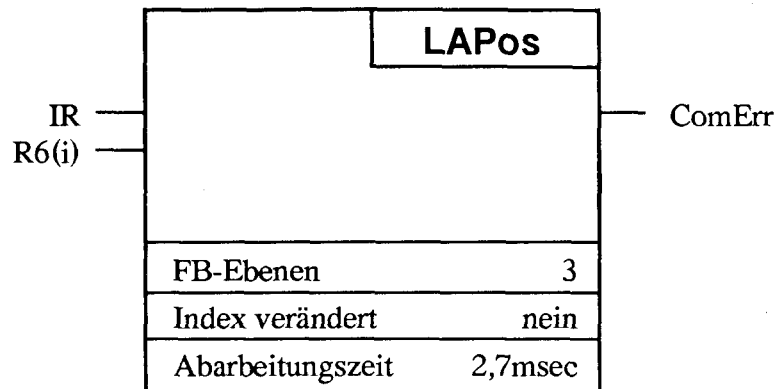
| | |
|----------|-----------------------------|
| FB's | INSTR, CONTAD, SETAD, WBYTE |
| Register | RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFKAG |

LAPos

Funktion : - Load Actual Position
- Istposition eingeben

LAPos

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Bei der Initialisierung des H2-Moduls mit dem Befehl "RSMC" wird der Positionszähler des SM-Controllers in die Mitte der maximalen Positionierdistanz mit dem Wert 8'388'608 geladen. Fährt man nun ausschliesslich Differenzwege oder der Schrittmotor wird mit manuellen Befehlen betrieben, dann braucht man sich um die absolute Istposition nicht zu kümmern. Vorausgesetzt wird, dass die max. Positionierdistanz von 16'777'215 nicht überfahren und der minimale Wert 0 nicht unterschritten wird. In einigen Anwendungen hingegen werden die anzufahrenden Positionen mit absoluten Werten (immer positiv) definiert. In diesem Fall muss nach der Suche des Referenzpunktes eine Istposition eingegeben werden. Der Befehl "LAPos" überträgt die Istposition vom Register 6 des Parameterblocks zum SM-Controller.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| R6(i) | Parameterblock APOS | | W | Integer | $0 \dots 2^{24} - 1$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

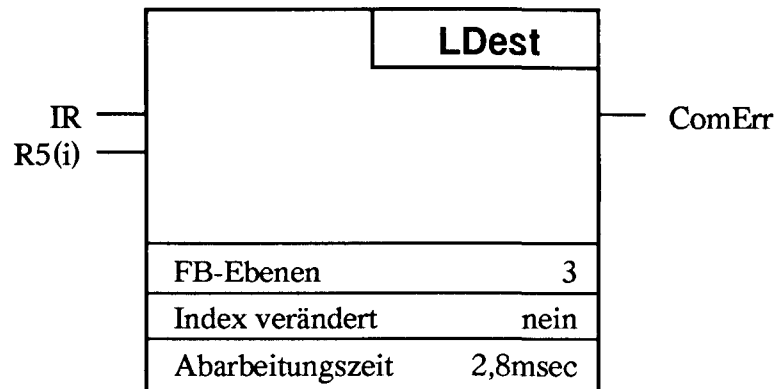
FB's CONTAD, SETAD, WDATA3, WBYTE
 Register Parameterblocks, APOS, RINDEX, TRAREG
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAFKLAG

LDest

Funktion : - Load Destination
- Sollposition eingeben

LDest

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Beim Positionieren mit absoluten Distanzen bezogen auf eine Referenzposition wird der Wert der anzufahrenden Position mit dem Befehl "LDest" vom Register 5 des Parameterblocks zum SM-Controller übertragen. Die Verwendung von absoluten Positionierwerten hat den Vorteil, dass die Wahl der Fahrriichtung im SM-Controller automatisch erfolgt. Sollpositionen sind immer positiv und durch die maximale Distanz von 16'777'215 begrenzt.

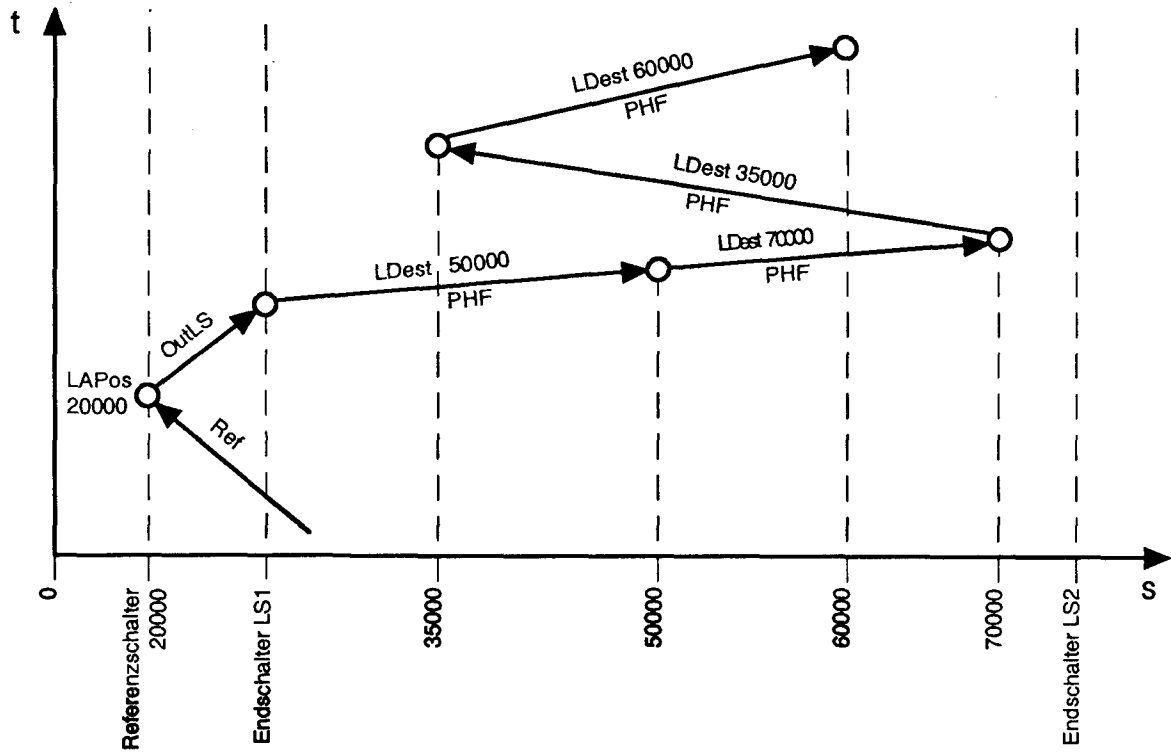
Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| R5(i) | Parameterblock POS | | W | Integer | $0 \dots 2^{24}-1$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|--------------------------------------|
| FB's | CONTAD, SETAD, WDATA3, WBYTE |
| Register | Parameterblocks, POS, RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFKAG |

Beispiel

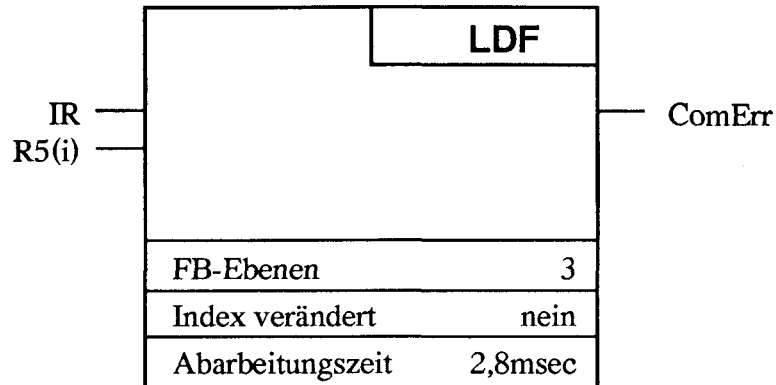


LDF

Funktion : - Load Difference Forward
 - Laden der Positionsdifferenz vorwärts

LDF

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Dieser Befehl überträgt den im Register 5 des Parameterblocks eingetragenen Weg zum SM-Controller. Gleichzeitig wird die Richtung des zu fahrenden Wegs mitgeteilt.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| R5(i) | Parameterblock POS | | W | Integer | $0 \dots 2^{24} - 1$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

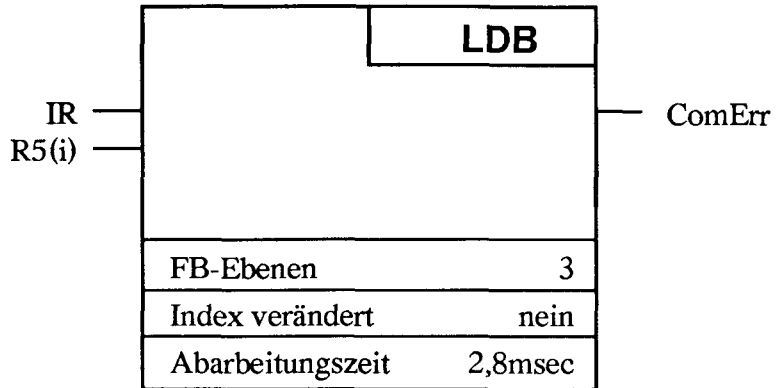
| | |
|----------|--------------------------------------|
| FB's | CONTAD, SETAD, WDATAB, WBYTE |
| Register | Parameterblocks, POS, RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFKAG |

LDB

Funktion : - Load Difference Backwards
- Laden der Positionsdifferenz rückwärts

LDB

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Dieser Befehl überträgt den im Register 5 des Parameterblocks eingetragenen Weg zum SM-Controller. Gleichzeitig wird die Richtung des zu fahrenden Wegs mitgeteilt.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| R5(i) | Parameterblock POS | | W | Integer | $0 \dots 2^{24}-1$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

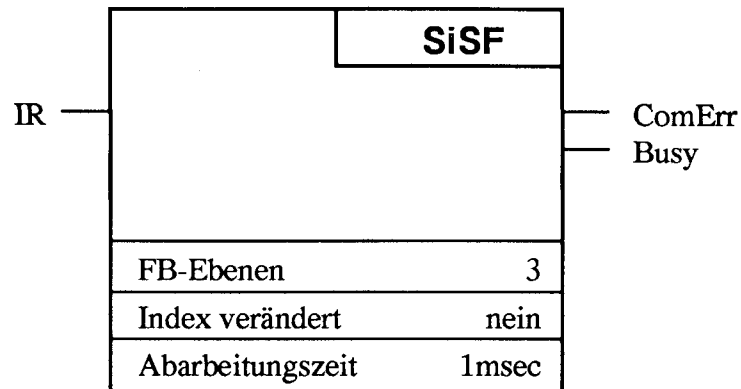
- FB's CONTAD, SETAD, WDATAB, WBYTE
- Register Parameterblocks, POS, RINDEX, TRAREG
- Timer -
- Counter -
- Flag ComErr, XYFLAG, TRAFLEG

SiSF

Funktion : - Single Step Forward
- Einzelschritt vorwärts

SiSF

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Normalerweise werden Einzelschritte mittels Betriebsart manuell gefahren, z.B. Anfahren einer Marke, Teach-in usw.

Im Automatikbetrieb müssen manchmal auch einzelne Schritte gefahren werden, sei es für eine Korrektur oder für Fahrten mit sehr niedriger Frequenz. Für diesen Zweck stehen die Befehle "SiSF" und "SiSB" zur Verfügung. Sie ergeben pro Aufruf einen Schritt vorwärts bzw. rückwärts.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

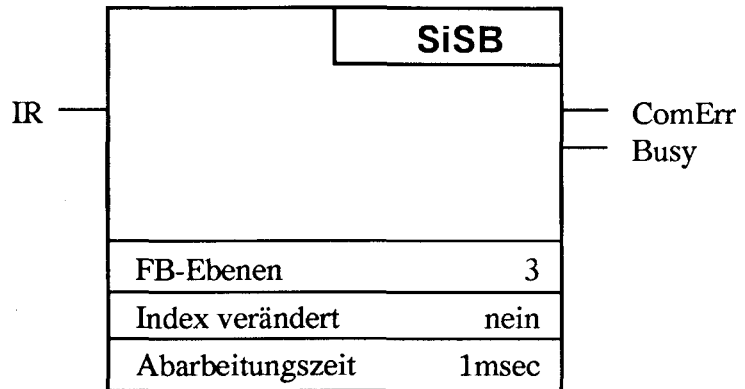
| | |
|----------|-----------------------------|
| FB's | INSTR, CONTAD, SETAD, WBYTE |
| Register | RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFLEG |

SiSB

Funktion : - Single Step Backwards
- Einzelschritt rückwärts

SiSB

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Normalerweise werden Einzelschritte mittels Betriebsart manuell gefahren, z.B. Anfahren einer Marke, Teach-in usw.

Im Automatikbetrieb müssen manchmal auch einzelne Schritte gefahren werden, sei es für eine Korrektur oder für Fahrten mit sehr niedriger Frequenz. Für diesen Zweck stehen die Befehle "SiSF" und "SiSB" zur Verfügung. Sie ergeben pro Aufruf einen Schritt vorwärts bzw. rückwärts.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

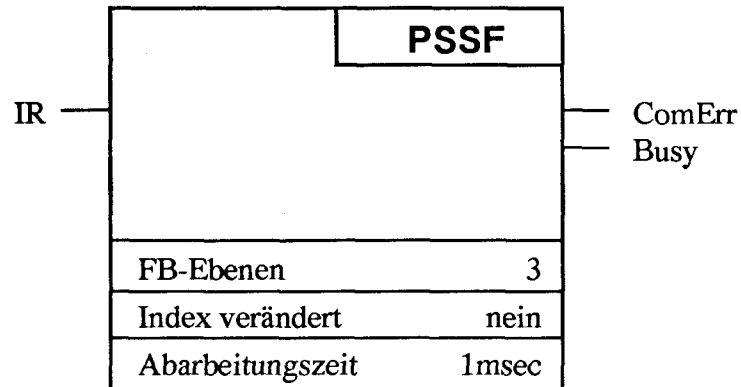
- FB's : INSTR, CONTAD, SETAD, WBYTE
- Register : RINDEX, TRAREG
- Timer : -
- Counter : -
- Flag : ComErr, XYFLAG, TRAFLEG

PSSF

Funktion : - Positioning with Start/Stop-Frequency
 - Positionierung mit Start/Stop-Frequenz

PSSF

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

"PSSF" ist der Startbefehl für das Fahren mit Start/Stop-Frequenz. Die Bewegung erfolgt unmittelbar nach Ausgabe des Befehls. Das "Busy" des SM-Controllers geht mit dem Befehl auf "0" und wird erst dann wieder "1" gesetzt, wenn die gewünschte Position erreicht ist. Beim Überfahren des Endschalters hält der Motor innerhalb des nächsten Schrittes an.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M_1, M_2, \dots, M_n |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

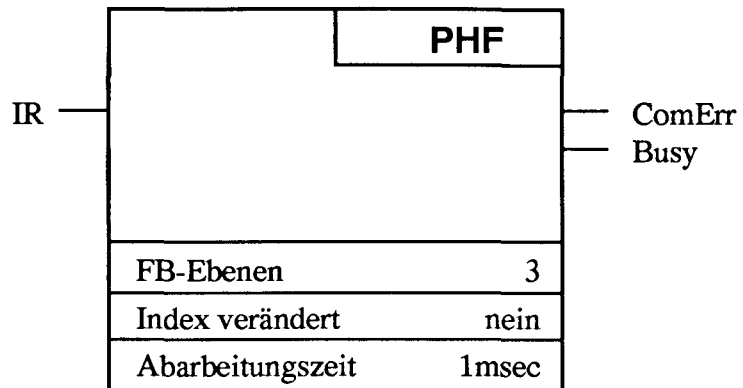
FB's INSTR, CONTAD, SETAD, WBYTE
 Register RINDEX, TRAREG
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAFKAG

PHF

Funktion : - Positioning with High Frequency
 - Positionierung mit Hochlauf-Frequenz (mit Rampe)

PHF

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

"PHF" ist der Startbefehl für das Fahren mit Hochlauf-Frequenz. Die Bewegung erfolgt unmittelbar nach Ausgabe des Befehls und verläuft quatz-genau nach dem vorausberechneten Geschwindigkeits-Zeit-Profil ab. Das "Busy" des SM-Controllers bleibt solange "0" gesetzt, bis die Position erreicht wird. Beim Überfahren des Endschalters hält der Motor mit kontrollierter Bremsung (Rampe) an. Dabei bleibt die Istposition ohne Schrittverlust erhalten.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|-----------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ , .. M _n |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

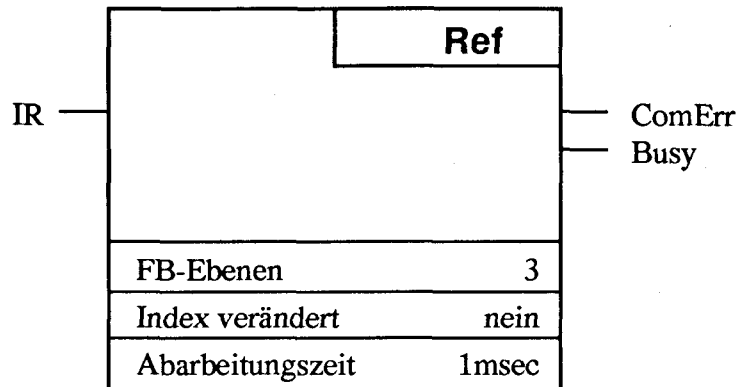
FB's INSTR, CONTAD, SETAD, WBYTE
 Register RINDEX, TRAREG
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAFLAG

Ref

Funktion : - Move to Reference Switch
- Nullpunkt suchen

Ref

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Durch diesen Befehl wird mit Start/Stop-Frequenz rückwärts gefahren, bis zum Referenzschalter. Sobald dieser betätigt wird stoppt der Motor schrittgenau. Das Überfahren des Endschalters LS1 hält die Bewegung nicht an. Das Fahren aus dem Endschalterbereich hingegen verlangt den Befehl "OutLS".

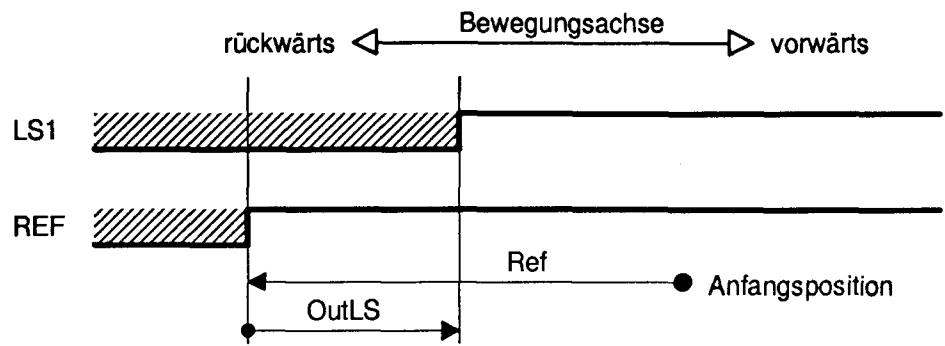
Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|-----------------------------|
| FB's | INSTR, CONTAD, SETAD, WBYTE |
| Register | RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFKAG |

Ref

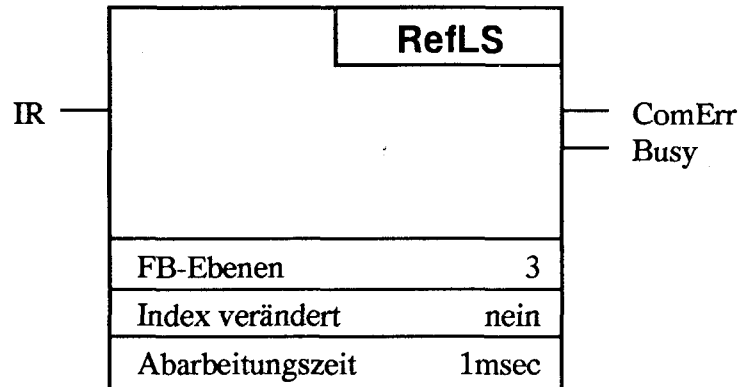


RefLS

Funktion : - Move to **R**eference, then to **L**imit Switch
 - Nullpunkt suchen und aus Endschalter
 herausfahren

RefLS

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Mit diesem Befehl wird mit Start/Stop-Frequenz rückwärts bis zum Referenzschalter gefahren, Drehrichtung gewechselt und vorwärts bis zum Austritt aus dem aktiven Bereich des Endschalters LS1 gefahren und gestoppt.

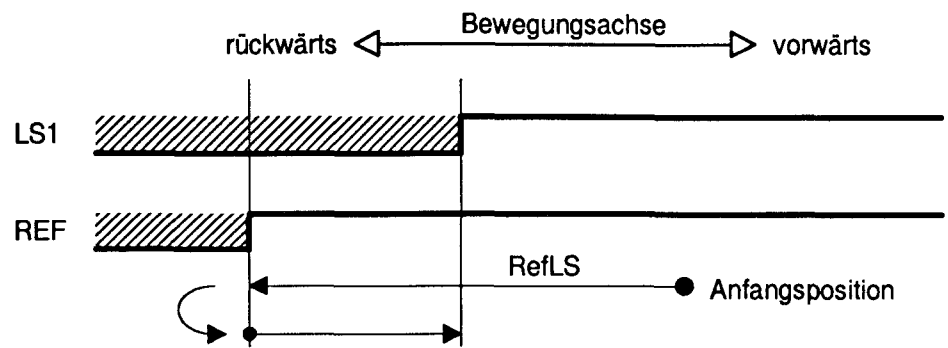
Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

FB's INSTR, CONTAD, SETAD, WBYTE
 Register RINDEX, TRAREG
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAFKAG

RefLS

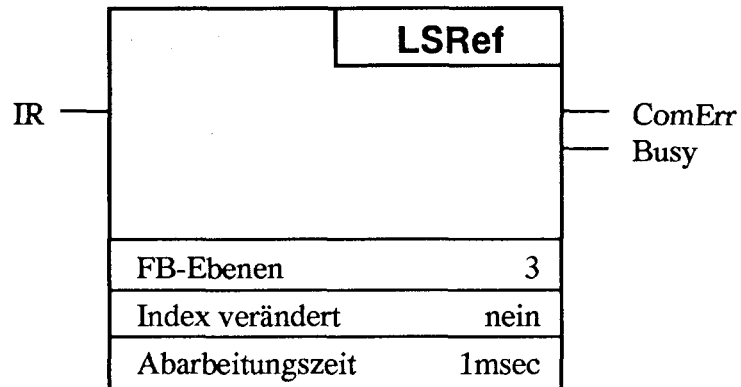


LSRef

Funktion : - Move to Limit Switch then to Reference Switch
 - Endschalter LS1 anfahren und zum Nullpunkt zurück fahren

LSRef

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Dieser Befehl dient einerseits zum präzisen Anfahren eines Nullpunktes mit einem auf der Motorwelle angebrachten Markersignal, das auf den Hardwareeingang REF geführt wird. Andererseits dient es zum Anfahren eines Referenzschalters, der sich innerhalb des gültigen Fahrbereiches d.h. zwischen LS1 und LS2 befindet. Dabei wird von einer Ausgangsposition im gültigen Fahrbereich rückwärts gefahren. Beim Auftreffen auf LS1 wird die Bewegungsrichtung umgekehrt, bis zur ersten Referenzmarke bzw. zum Referenzschalter und gestoppt. Alle weiteren Impulse auf den Hardwareeingang REF werden ignoriert.

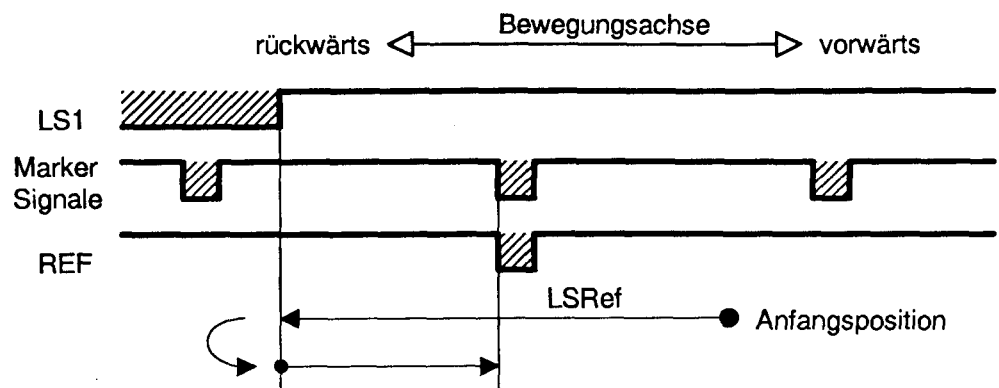
Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|-----------------------------|
| FB's | INSTR, CONTAD, SETAD, WBYTE |
| Register | RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFKAG |

LSRef

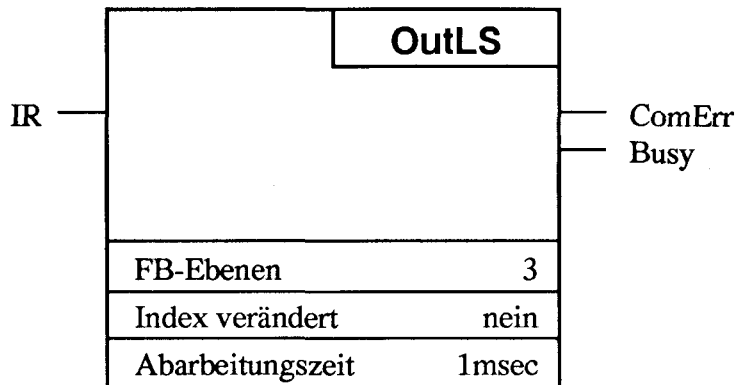


OutLS

Funktion : - Move Out of Limit Switch
 - Aus Endschalterbereich herausfahren

OutLS

Softwarepaket : PCD9.H2FB E1

**Funktionsbeschreibung:**

Mit diesem Befehl wird mit Start/Stop-Frequenz aus dem aktiven Bereich des entsprechenden Endschalters herausgefahren.

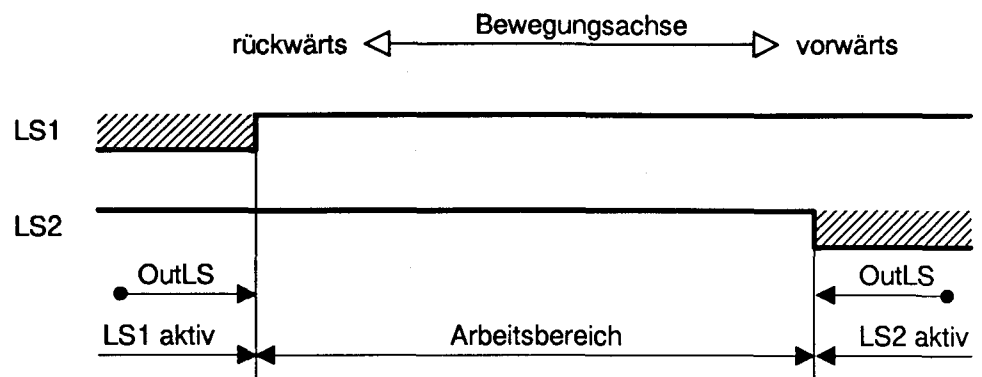
Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|-----------------------------|
| FB's | INSTR, CONTAD, SETAD, WBYTE |
| Register | RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAF LAG |

OutLS

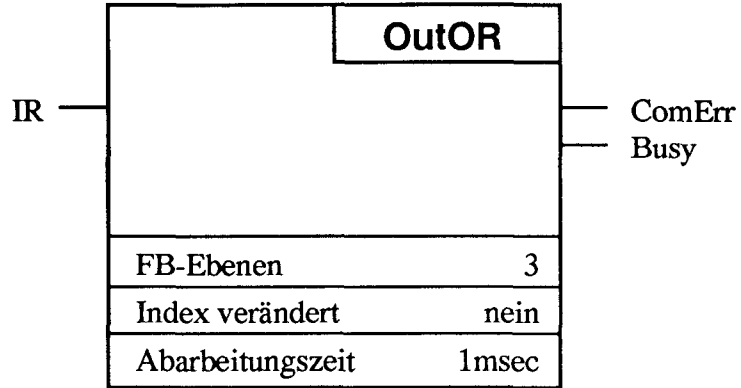


OutOR

Funktion : - Move Out of Overrun Range
 - Aus Overrun-Bereich herausfahren

OutOR

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Dieser Befehl bewirkt, dass mit Start/Stop-Frequenz zum Eintrittspunkt des Endschalters LS1 oder LS2 zurückgekehrt wird, nachdem dieser vorher überfahren wurde (Overrun). Im Gegensatz zu "OutLS" spielt es hier keine Rolle, ob der Motor im aktiven oder nicht-aktiven Bereich des Endschalters zum Stillstand kam.

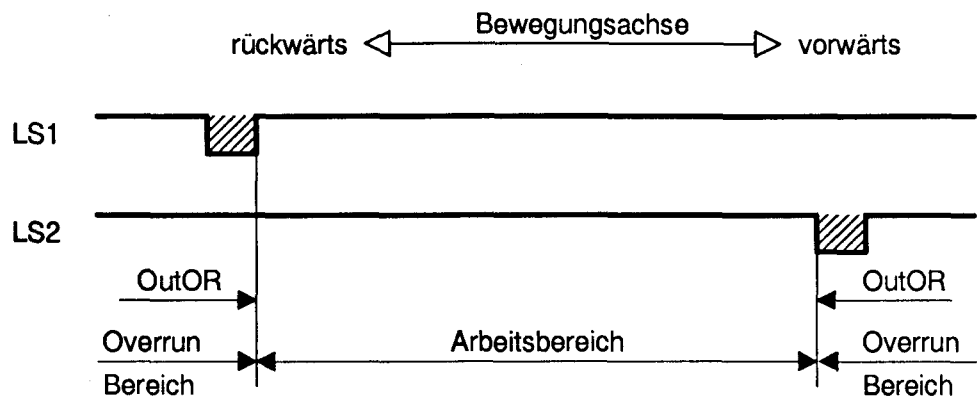
Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|-----------------------------|
| FB's | INSTR, CONTAD, SETAD, WBYTE |
| Register | RINDEX, TRAREG |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFAG |

OutOR

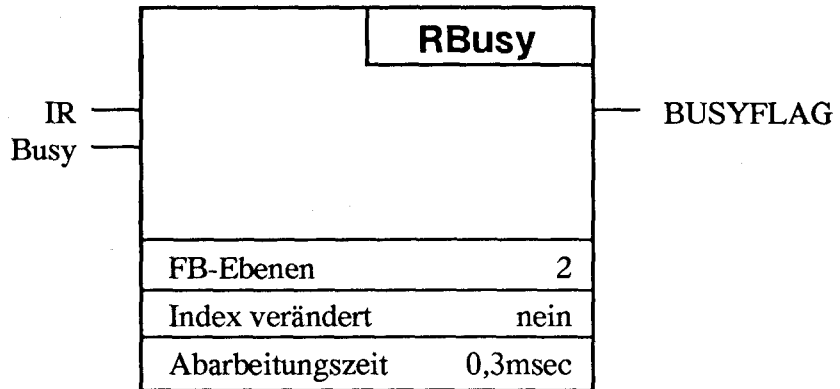


RBusy

Funktion : - Read **Busy**-Status of SM-Controller
 - **Busy**-Status des SM-Controllers abfragen

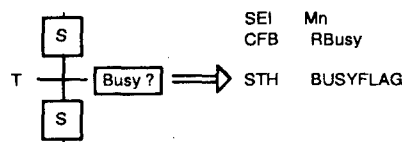
RBusy

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Mit "RBusy" wird die Bereitschaft des SM-Controllers geprüft und der Merker "BUSYFLAG" entsprechend gesetzt.
 "BUSYFLAG" = 0 bedeutet SM-Controller besetzt oder anzufahrende Position noch nicht erreicht. Während der Ausführung eines Befehls kann der SM-Controller keinen weiteren Befehl annehmen. Mit "RBusy" wird der sequentielle Prozessablauf bei Schrittmotorantrieben gesteuert. Das Ende einer Tätigkeit des Motorsystems wird in einem Graftec-Programm folgendermassen in der Transition abgefragt:



Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|----------|------------------------|-----------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |
| Busy | Input | | I | Binary | 0, 1 |
| BUSYFLAG | Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

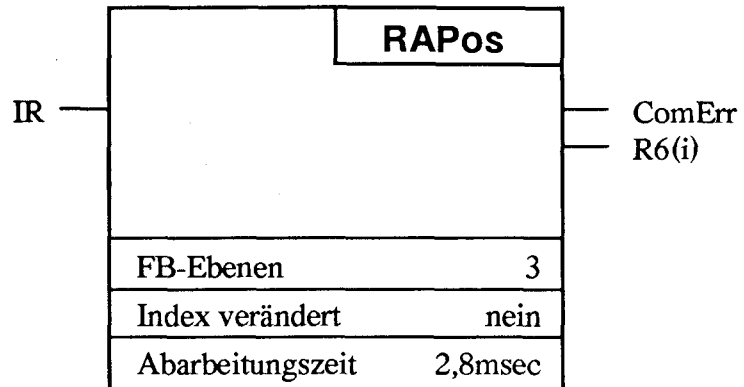
| | |
|----------|------------------|
| FB's | CONTAD, SETAD |
| Register | RINDEX |
| Timer | - |
| Counter | - |
| Flag | XYFLAG, BUSYFLAG |

RAPos

Funktion : - Read Actual Position
- Istposition auslesen

RAPos

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Mit diesem Befehl wird die aktuelle Istposition des Motors gelesen und in das Register 6 des Parameterblocks übertragen. Der Befehl "RAPos" ist nur bei Stillstand des Motors ausführbar. Normalerweise ist das Lesen der Istposition nicht notwendig, weil sie dem vorgegebenen Sollwert immer entspricht. Doch gibt es Anwendungsfälle, wo das Lesen der Istposition erforderlich ist:

- Beim vorzeitigen Abbruch einer Bewegung im automatischen Betrieb
- Lesen einer Position die manuell angefahren wurde (z. B. bei "Teach-in")

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| R6(i) | Parameterblock APOS | | R | Integer | $0 \dots 2^{24}-1$ |

Intern benützte Medien und Sub-Funktionsblöcke

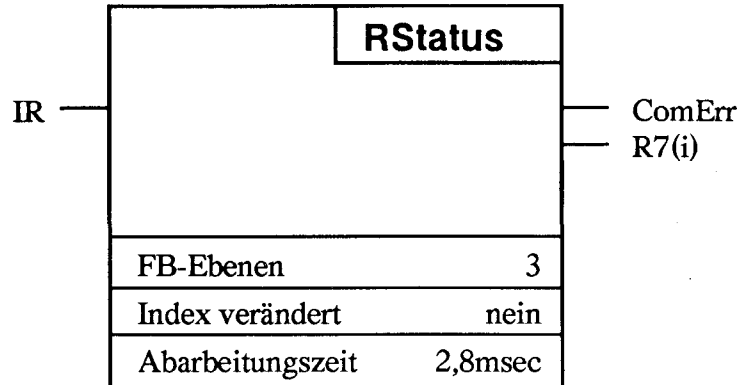
| | |
|----------|-----------------------------------------------|
| FB's | CONTAD, SETAD, WBYTE, RDATA |
| Register | Parameterblock, TEMPREG, RINDEX, TRAREG, APOS |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFKAG |

RStatus

Funktion : - Read Status of SM-Controller
 - Betriebszustand des SM-Controllers lesen

RStatus

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Der Betriebszustand des angesprochenen Motor-Systems wird in das Register 7 "STATUS" des Parameterblocks übertragen.

Beschreibung der Ein- und Ausgänge:

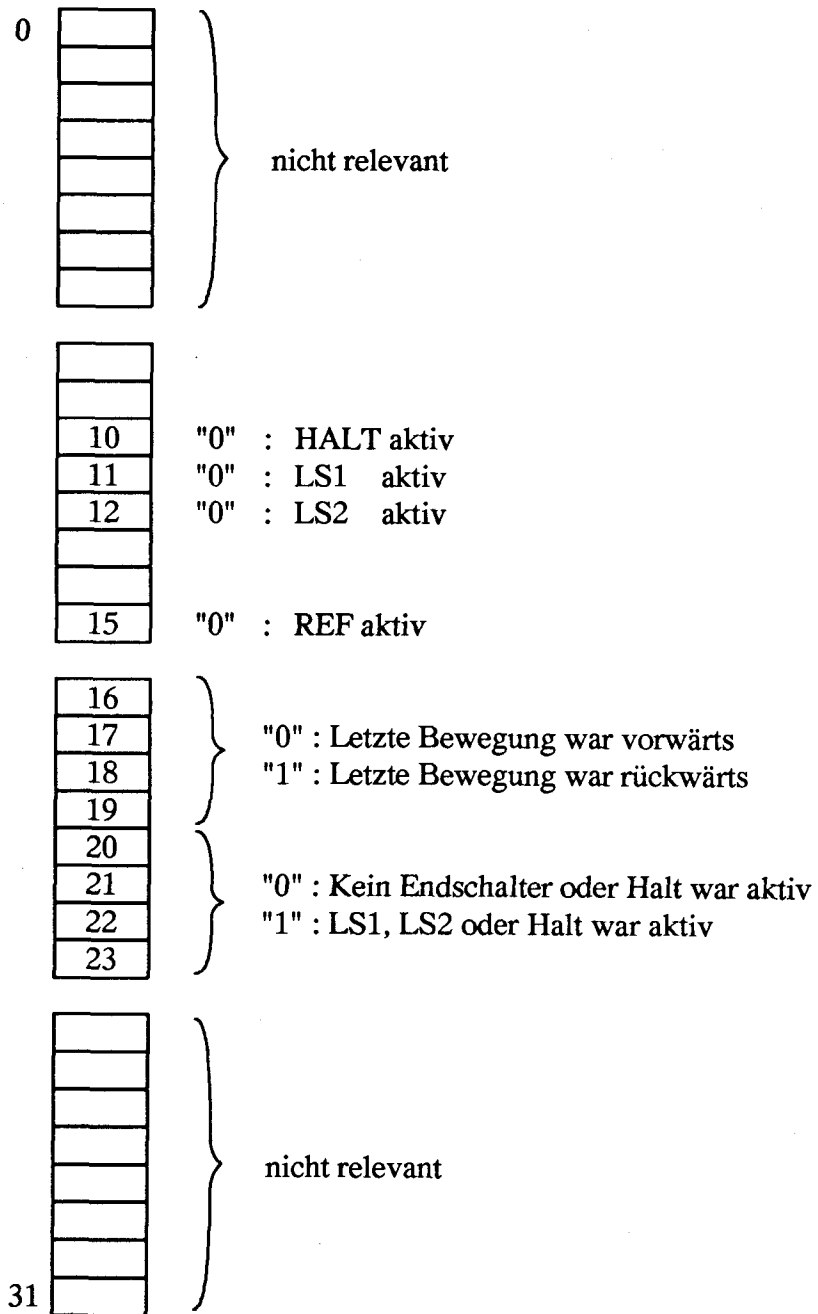
| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| R7(i) | Parameterblock STATUS | | R | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

FB's CONTAD, SETAD, WBYTE, RDATA
 Register Parameterblock, TEMPREG, RINDEX, TRAREG
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAFLAG

Bedeutung der Statusinformation

Register 7 "STATUS"

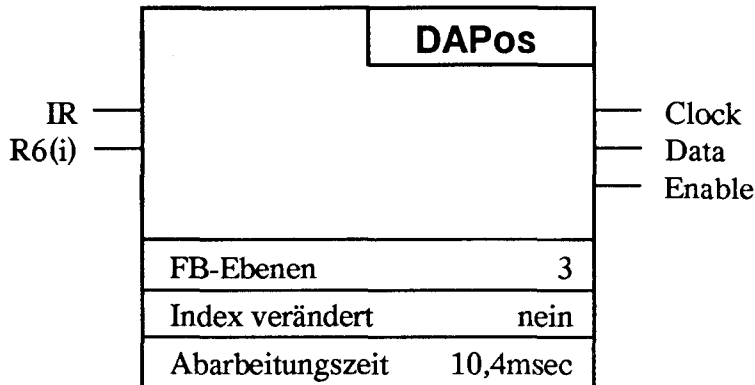


DAPos

Funktion : - Display Actual Position with PCA2.D14
 - Istposition mittels PCA2.D14 anzeigen

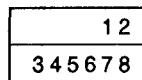
DAPos

Softwarepaket : PCD9.H2FB E1



Funktionsbeschreibung:

Mit diesem Befehl wird der Inhalt des Registers 6 des Parameterblocks in das Anzeigemodul PCA2.D14 übertragen und angezeigt. Die Anzeige erfolgt 8-stellig, die 6 niederwertigen Stellen auf der unteren Anzeigeebene und die 2 höheren Stellen auf der oberen Anzeigeebene.



Bei Positionierdistanzen < 1'000'000 können die Werte beider Motoren X und Y eines Moduls auch mit einem einzigen PCA2.D14-Modul angezeigt werden. Zu diesem Zweck ist dann der Befehl "DP14" zu verwenden.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| Clock | Ausgang zu D14 | | O | Binary | 0, 1 |
| Data | " " | | O | Binary | 0, 1 |
| Enable | " " | | O | Binary | 0, 1 |
| R6(i) | Parameterblock APOS | | R | Integer | $0.. 2^{24}-1$ |

Intern benützte Medien und Sub-Funktionsblöcke

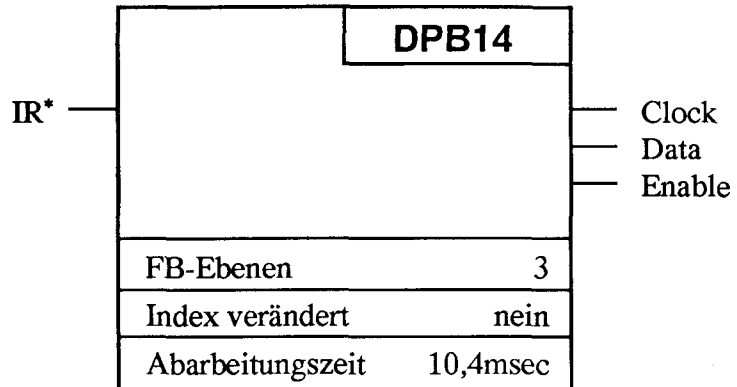
| | |
|----------|----------------------------|
| FB's | RBO, CONTAD, SETAD, DP14 |
| Register | REG_1, RH_0, RINDEX |
| Timer | - |
| Counter | CH_0 |
| Flag | XYFLAG, FH_0 ... FH_0 + 47 |

DPB14

Funktion : - Clear Display on PCA2.D14
 - Betriebszustand des SM-Controllers lesen

DPB14

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Mit dem Befehl "DPB14" werden alle Ziffern der Anzeige PCA2.D14 mit Leerzeichen gefüllt, so dass sie erlöschen.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |
| Clock | Ausgang zu D14 | | O | Binary | 0, 1 |
| Data | " " | | O | Binary | 0, 1 |
| Enable | " " | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

FB's RBO
 Register RH_0
 Timer -
 Counter CH_0
 Flag XYFLAG, FH_0 ... FH_0 + 47

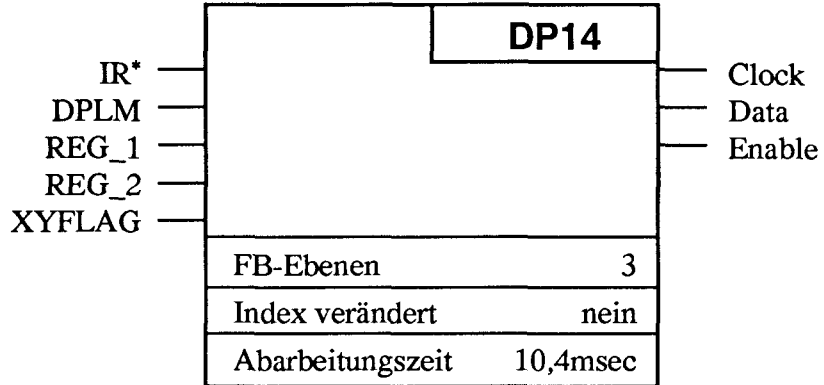
*) Das Indexregister muss hier mit der Basisadresse des H2-Moduls geladen werden.

DP14

Funktion : - Display Contents of Register on PCA2.D14
 - Registerwerte mit PCA2.D14 anzeigen

DP14

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Mit diesem Befehl kann ein Registerwert mit 1*10 Digit oder mit 2*6 Digit auf dem Anzeigemodul PCA2.D14 angezeigt werden.

Folgende Anzeigeformate sind möglich:

1*10 Digits (1Register)

| | | | | | |
|---|---|---|---|---|----|
| b | v | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 |

 b = blank ; v = blank oder "-"
 1 .. 10 = Digits

Anzeigebereich = Wertebereich der Register ± 2'147'483'647

2*6 Digits (2Register)

| | | | | | |
|---|---|---|---|---|---|
| v | x | x | x | x | x |
| v | x | x | x | x | x |

 w = Inhalt des Registers
 0 ≤ w ≤ + 999'999 : v = MSD
 w > + 999'999 : v = "A"
 0 > w ≥ - 99'999 : v = "-"
 w < - 99'999 : v = "U"

Anzeigebereich : - 99'999 ... + 999'999

Mit dem Modul PCD4.H220 können 2 Anzeigemodule betrieben werden. Die Anzeige mit Anschluss auf Klemme 7 erfolgt mit "XYFLAG" = "0", mit Anschluss auf Klemme 15 mit "XYFLAG" = "1".

Der Merker DPLM bestimmt die Art der Anzeige:

DPLM = "0" --> 1*10 Digits
 DPLM = "1" --> 2* 6 Digits

Register: REG_1 : Register für den 1. Anzeigewert
 REG_2 : Register für den 2. Anzeigewert

*) Das Indexregister muss hier mit der Basisadresse des H2-Moduls geladen werden.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| DPLM | Display Modus | | F | Binary | 0, 1 |
| REG_1 | Anzeigewert 1 | | R | Integer | $0 \dots 10^9 - 1$ |
| REG_2 | Anzeigewert 2 | | R | Integer | $0 \dots 10^6 - 1$ |
| Clock | Ausgang zu D14 | | O | Binary | 0, 1 |
| Data | " " | | O | Binary | 0, 1 |
| Enable | " " | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|----------------------------|
| FB's | RBO |
| Register | REG_1, REG_2, RH_0 |
| Timer | - |
| Counter | CH_0 |
| Flag | XYFLAG, FH_0 ... FH_0 + 47 |

Programmbeispiel für die Anzeige der Istposition vom Motor1 (M1) und Motor2 (M2) mit einem einzigen PCA2.D14:

```

;
SEI M1 ;
GETX PARA+6 ;
REG_1 ; Istposition M1
SEI M2 ;
GETX PARA+6 ;
REG_2 ; Istposition M2
SEI Base Addr. ; Basisadresse des H2-Moduls
ACC H ;
RES XYFLAG ; Enable auf X-System
SET DPLM ; 2*6 Digits
CFB DP14 ;
;

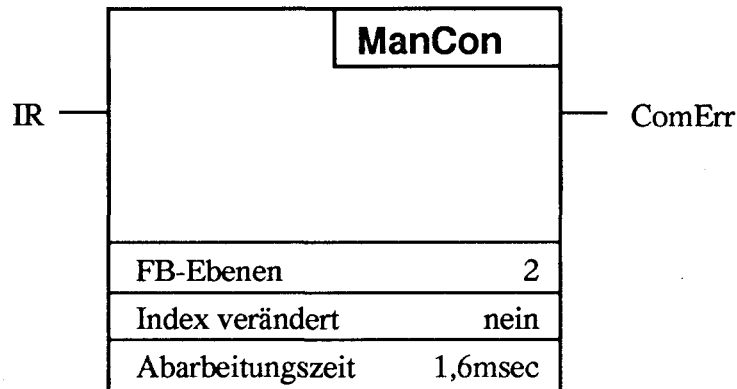
```

ManCon

Funktion : - Manual Control
- Umschaltung auf Handbetrieb

ManCon

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:**

Mit diesem Befehl wird vom automatischen in den manuellen Betrieb umgeschaltet. Der Merker "BUSYFLAG" bleibt von nun an immer "0" und wird in dieser Betriebsart nicht abgefragt. Im manuellen Betrieb können ausschliesslich nur manuelle Befehle ausgeführt werden.

Manchmal wird beim Handbetrieb eine Funktion benötigt, die nur bei den Automatikbefehlen zu finden ist. In diesem Fall wird zuerst mit dem Befehl "AutoCon" auf Automatik geschaltet, der entsprechende Befehl ausgeführt und dann wieder mit "ManCon" in den manuellen Betrieb zurückgekehrt. Der Befehl "ManCon" (sowie "AutoCon") wirkt auf beide Systeme (X und Y) eines Moduls gleichzeitig.

"ManCon" darf frühestens 0,2sec nach Beendigung des letzten Automatikbefehls aufgerufen werden.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| ComErr | Error-Flag | | F | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

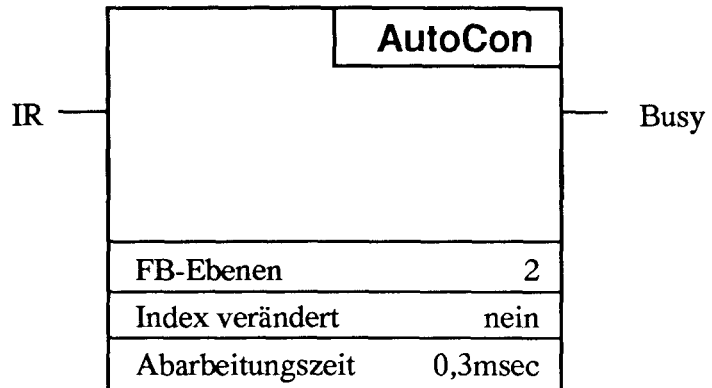
| | |
|----------|-------------------------|
| FB's | CONTAD, SETAD, WBYTE |
| Register | TRAREG, RINDEX |
| Timer | - |
| Counter | - |
| Flag | ComErr, XYFLAG, TRAFLEG |

AutoCon

Funktion : - Switch to **Automatic Control**
 - Umschaltung auf Automatikbetrieb

AutoCon

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Dieser Befehl erlaubt die Rückkehr vom Handbetrieb in den Automatikbetrieb. Es ist zu beachten, dass dieser Befehl auf beide Systeme (X und Y) eines Moduls wirkt.

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |
| Busy | Output | | O | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

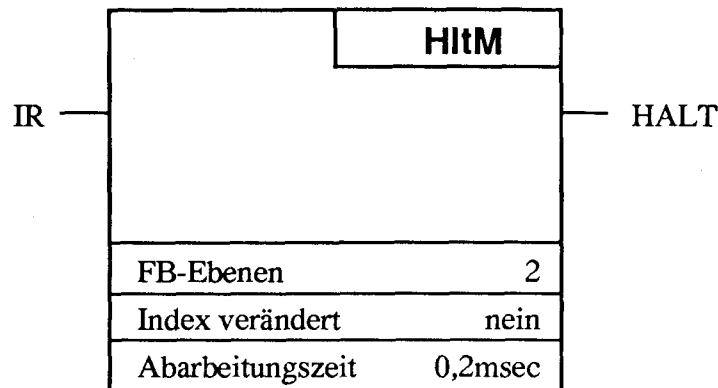
| | |
|----------|---------------|
| FB's | CONTAD, SETAD |
| Register | RINDEX |
| Timer | - |
| Counter | - |
| Flag | - |

HltM

Funktion : - Halt any Motion
- Anhalten einer Bewegung

HltM

Softwarepaket : PCD9.H2FBE1

**Funktionsbeschreibung:****Handbetrieb**

Der Befehl "HltM" hält die von Hand gestartete Bewegung an. Bei der Fahrt mit Hochlauf-Frequenz bewirkt "HltM" ein kontrolliertes Anhalten mit Rampe.

Automatikbetrieb

Der Befehl "HltM" hält die laufende Bewegung an. Eine Weiterfahrt ist nur dann möglich, wenn mit dem Befehl "Cont" der "HltM"-Befehl gelöscht und der Motor mit den Befehlen "PSSF" bzw. "PHF" neu gestartet wird. Nach Empfang des Befehls "HltM" nimmt der SM-Controller während 60 ms keine Befehle entgegen. Vor dem Löschbefehl "Cont" muss daher mind. 60 ms gewartet oder die Eingangsadresse 10 (+ Basisadr.) abgefragt werden (siehe Kap. 5.4).

Beschreibung der Ein- und Ausgänge

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |
| HALT | Status | | I | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

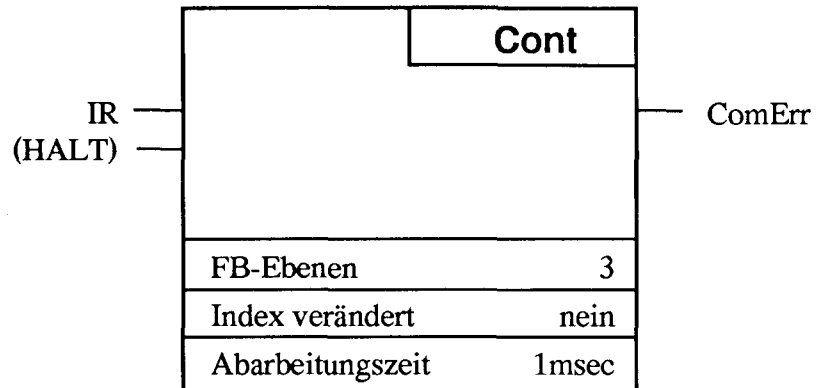
| | |
|----------|---------------|
| FB's | CONTAD, SETAD |
| Register | RINDEX |
| Timer | - |
| Counter | - |
| Flag | XYFLAG |

Cont

Funktion : - Continue after "HltM"
 - Freigabe nach "HltM"

Cont

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Nach Unterbruch einer Bewegung durch den Befehl "HltM" oder nach kurzzeitigem Aktivieren eines Endschalters, kann der im Controller sich befindende Anhaltermerker mit dem Befehl "Cont" gelöscht werden. Danach ist es möglich, die Bewegung mit Fahrbefehlen zu Ende zu führen oder auch die Istposition zu Lesen und die Fahrt abubrechen. Aus Sicherheitsgründen wird der Befehl "Cont" nach "HltM" erst nach mind. 60 ms vom SM-Controller entgegengenommen (siehe Befehl "HltM").

Beschreibung der Ein- und Ausgänge:

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |
| ComErr | Error-Flag | | F | Binary | 0, 1 |
| HALT | Status | | I | Binary | 0, 1 |

Intern benützte Medien und Sub-Funktionsblöcke

FB's INSTR, CONTAD, SETAD, WBYTE
 Register RINDEX, TRAREG
 Timer -
 Counter -
 Flag ComErr, XYFLAG, TRAFLEG

OSF
OSB

Funktion : - One Step manual Forward , Backwards
- Einzelschritt manuell vorwärts , rückwärts

OSF
OSB

Softwarepaket : PCD9.H2FBE1

| | | |
|----|------------------|---------|
| IR | OSF , OSB | |
| | FB-Ebenen | 2 |
| | Index verändert | nein |
| | Abarbeitungszeit | 0,4msec |

Funktionsbeschreibung:

Mit diesen Befehlen wird im manuellen Betrieb ein Schritt vorwärts bzw. rückwärts ausgeführt. Diese Einzelschritte werden auch dann ausgeführt, wenn ein Endschalter überfahren wird.

Beschreibung der Ein- und Ausgänge

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|----------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | $M_1, M_2 \dots M_n$ |

Intern benützte Medien und Sub-Funktionsblöcke

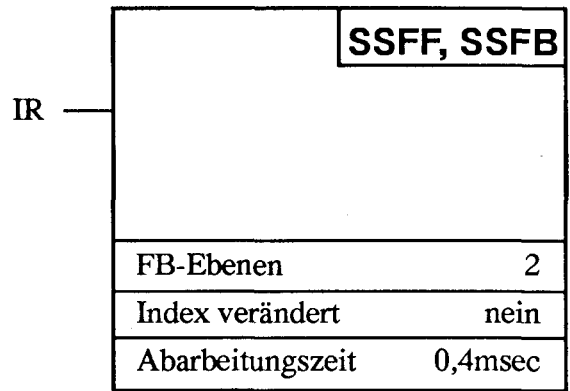
| | |
|----------|---------------|
| FB's | CONTAD, SETAD |
| Register | RINDEX |
| Timer | - |
| Counter | - |
| Flag | XYFLAG |

SSFF
SSFB

Funktion : - Manual with Start/Stop Frequency
Forward , Backwards
- Manuell mit Start/Stop-Frequenz
vorwärts , rückwärts

SSFF
SSFB

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Diese Befehle starten den Motor mit Start/Stop-Frequenz vorwärts bzw. rückwärts. Der Motor dreht dann solange, bis er mit dem Befehl "HltM" angehalten oder der Endschalter angefahren wird.

Beschreibung der Ein- und Ausgänge

| Symbol | Bezeichnung / Funktion | Para- meter | Daten | | |
|--------|------------------------|----------------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |

Intern benützte Medien und Sub-Funktionsblöcke

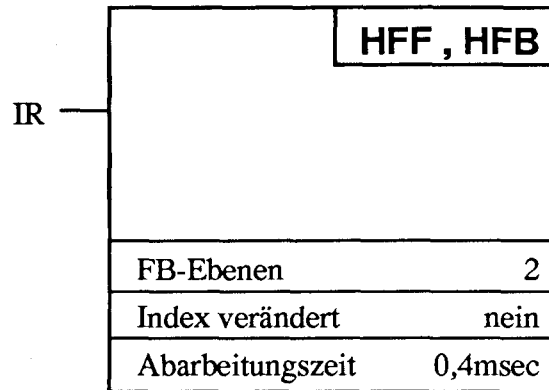
FB's CONTAD, SETAD
Register RINDEX
Timer -
Counter -
Flag XYFLAG

HFF
HFB

Funktion : - Manual with High Frequency Forward, Backwards
-Manuell mit Hochlauf-Frequenz vorwärts,
rückwärts

HFF
HFB

Softwarepaket : PCD9.H2FBE1



Funktionsbeschreibung:

Diese Befehle starten den Motor mit Hochlauf-Frequenz vorwärts bzw. rückwärts. Der Motor dreht dann solange, bis der mit dem Befehl "HltM" angehalten oder der Endschalter angefahren wird. Das Anhalten erfolgt in beiden Fällen via Bremsrampe.

Beschreibung der Ein- und Ausgänge

| Symbol | Bezeichnung / Funktion | Parameter | Daten | | |
|--------|------------------------|-----------|-------|---------|---------------------------------------------------|
| | | | Typ | Format | Wert |
| IR | Index Register | | K | Integer | M ₁ , M ₂ .. M _n |

Intern benützte Medien und Sub-Funktionsblöcke

| | |
|----------|---------------|
| FB's | CONTAD, SETAD |
| Register | RINDEX |
| Timer | - |
| Counter | - |
| Flag | XYFLAG |

8. Fehlererkennung und -Behandlung

Übertragungsfehler

Ein Übertragungsfehler wird dadurch erkannt, dass der SM-Controller die ihm gesendeten Daten nicht quittiert. Die Quittung bleibt dann aus, wenn die Daten nicht verstanden werden oder wenn sie zur unpassenden Zeit übertragen werden. Bei einem Übertragungsfehler wird der Merker "ComErr" = "1" gesetzt.

Nach einem Übertragungsfehler und somit bei "ComErr" = "1" bleibt aus Sicherheitsgründen die Abarbeitung weiterer Schrittmotorbefehle solange unterdrückt, bis der Anwender den "ComErr"-Merker wieder rückstellt. Das Überprüfen eines Übertragungsfehlers kann individuell, unmittelbar nach einem Fahrbefehl erfolgen. Das Erreichen der Zielposition braucht dabei nicht abgewartet zu werden. Zweckmässigerweise erfolgt jedoch die Überprüfung innerhalb der zyklischen Prozessüberwachung, wo im Falle eines Fehlers jede weitere Bewegung verhindert wird. Übertragungsfehler werden nicht durch die ERROR-LED auf der Frontplatte angezeigt.

Störung des SM-Controllers

Im SM-Controller ist eine Überwachung eingebaut, die im Falle einer Störung ein Fehlersignal auf die Eingangsadressen 9 und 13 generiert. Zudem wird die Störung mittels der ERROR-LED auf der Frontplatte des Moduls angezeigt (siehe Kap. 4. Blockschaltbild).

Ein externer STOP für sofortigen Bewegungsabbruch wird vom Controller ebenfalls als Störung aufgefasst. Hat der SM-Controller eine Störung festgestellt, dann schaltet er sich ab und verharrt in diesem Zustand. Dabei dürfte in der Regel ein Schrittverlust entstanden sein oder eine Positionierung wurde gar nicht ausgeführt. Nach jeder Störung des SM-Controllers ist es deshalb erforderlich, eine Neu-Initialisierung des H2-Moduls vorzunehmen.

9. Anwendungsbeispiel

Bei diesem Beispiel handelt es sich um eine typische Anwendung für zwei Achsen.

Aufgabenstellung

Fräsen eines spiralförmigen Vierkants auf einem Rundstab.

Motor M1 (x) bewirkt den Fräsvorschub

Motor M2 (y) bewirkt die Drehbewegung des Rundstabes



Ablauf:

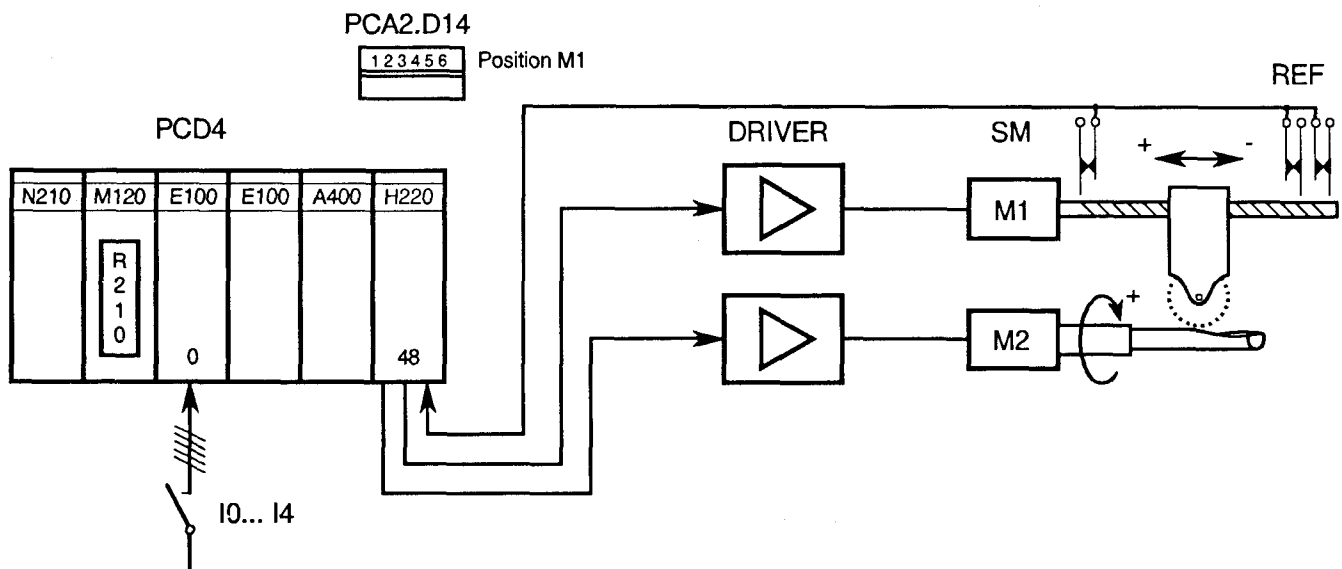
Bei erfolgreichem Start sucht M1 den Referenzpunkt und fährt anschliessend mit hoher Geschwindigkeit zum Ausgangspunkt der Bearbeitung.

M1 und M2 fahren koordiniert die Bearbeitungsstücke ab und halten an.

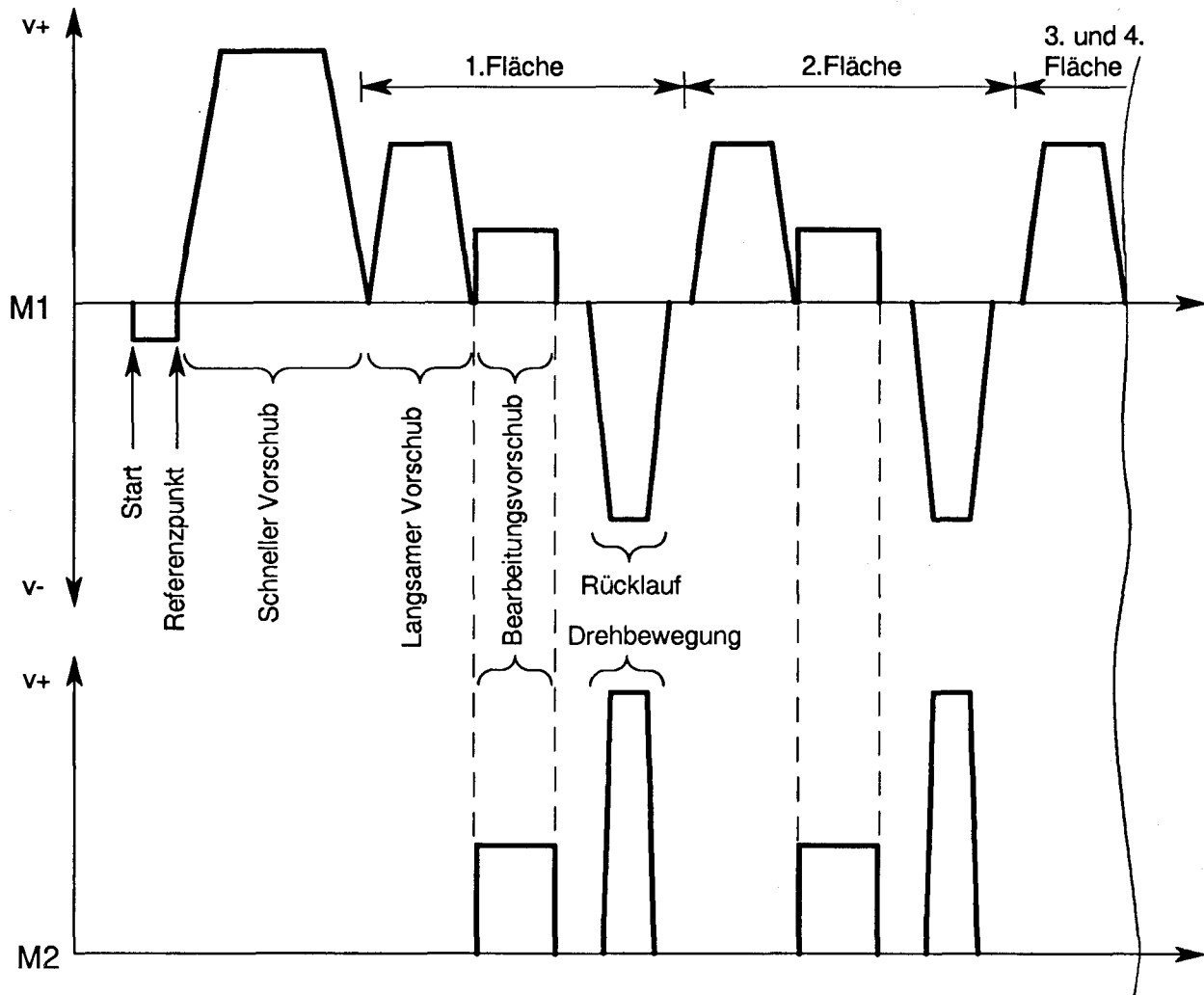
M1 fährt in die Ausgangslage zurück während M2 die Teilung auf 90° ergänzt.

Dieser Bearbeitungsvorgang wird 4 mal wiederholt, um den Vierkant vollständig zu bearbeiten.

Hardware:



Folgendes Bewegungsdiagramm wird ausgeführt



Bedienung über Eingänge

- Automatik Eingang 0 :L = Automatik
Eingang 1 = Start
- Manuell Eingang 0 :H = Manuell
Eingang 2 = vorwärts
Eingang 3 = rückwärts
Eingang 4 = aus Endschalterbereich herausfahren

Anzeige

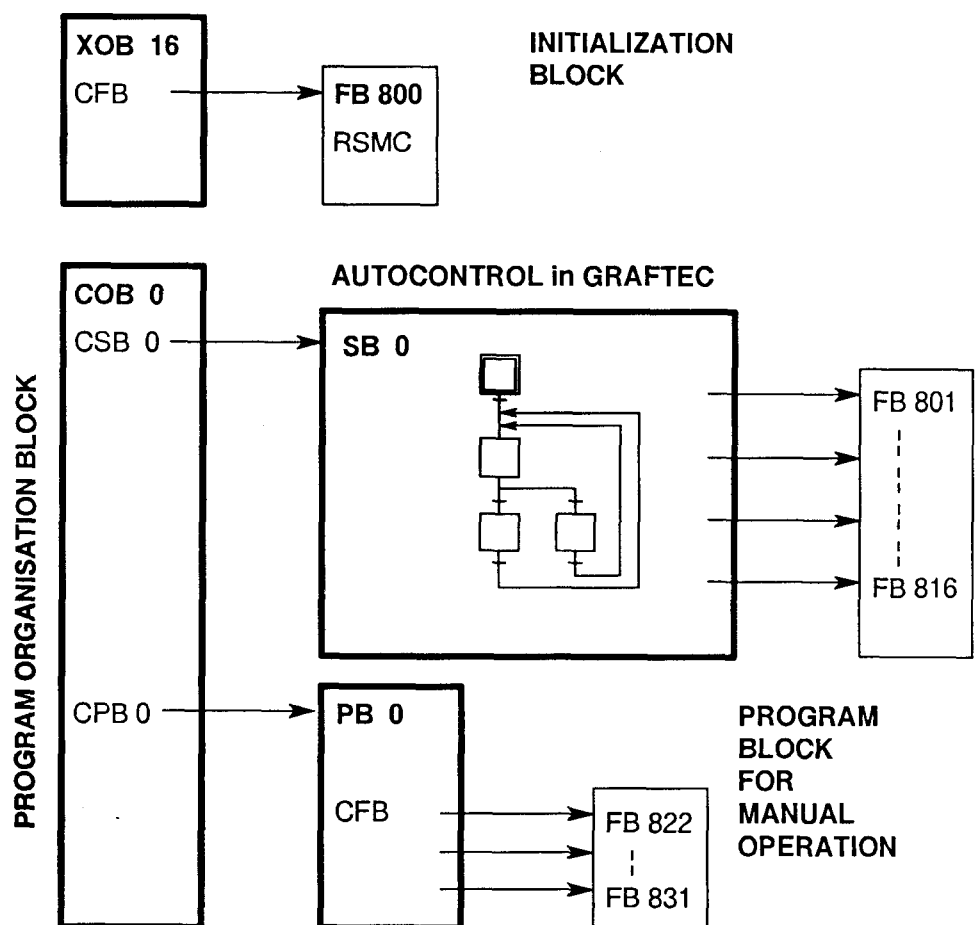
Mit dem Display-Modul PCA2.D14 werden einzelne Positionen des M1-Motors im Automatikablauf angezeigt.

Organisation des Schrittmotorprogramms

Das Laden der Basis-Adresse des H2-Moduls sowie die Kennzeichnung der Systeme X und Y in die entsprechenden Parameterblöcke erfolgt am einfachsten im Kaltstart-Organisationsblock XOB16. Die Initialisierung der SM-Controller mit RSMC sollte ebenfalls in diesem Organisationsblock stattfinden, weil dort ohne Nachteile auch die nötige Zeitverzögerung programmiert werden kann.

Der sequentielle Bewegungsablauf des Automatik-Programmes wird zweckmässigerweise mittels GRAFTEC programmiert. Hingegen ist es vorteilhaft, die manuelle Bedienungsebene in einen zyklischen Programm-block unterzubringen, der nur dann aufgerufen wird, wenn manuelles Steuern der Bewegungen gewünscht wird.

Die Umschaltung von "Automatik" nach "Manuell" erfolgt im Automatik-Programm, weil dort jeweils das Ende einer Automatik-Sequenz abgewartet werden muss. Hingegen erfolgt die Umschaltung von "Manuell" nach "Automatik" im Manuell-Programm, um sicher zu sein, dass die letzte manuell ausgelöste Bewegung tatsächlich beendet wurde, bevor mit automatischen Sequenzen weitergefahren wird.



```

$INCLUDE H2DEF.SRC
;
;INITIALIZATION BLOCK
;=====
;
;IN XOB 16 MODULE BASE ADDRESS AND X/Y IDENTIFICATION ARE
;          LOADED INTO PARAMETER BLOCKS FOR M1 AND M2
;          THEN MODULE IS RESET BY RSMC AND A TIME DELAY IS
;          PROGRAMMED TO ASSURE CORRECT RESETTING
;
;
;*****
      XOB      16
;-----
      SEI      K 1      M1      PARAMETER MOTOR 1 (BLOCK OF 8 REG)
      LDX      R 0      PARA     PARAMETER BLOCK (0-> BASE ADDRESS)
              48
      LDX      R 1      PARA+1   PARAMETER BLOCK (0-> BASE ADDRESS)
              0
;
      SEI      K 9      M2      PARAMETER MOTOR 2 (BLOCK OF 8 REG)
      LDX      R 0      PARA     PARAMETER BLOCK (0-> BASE ADDRESS)
              48
      LDX      R 1      PARA+1   PARAMETER BLOCK (0-> BASE ADDRESS)
              1
;-----
      CFB      800      RSMC     RESET STEPPING MOTOR CONTROLLER
;-----
; WAIT END OF RESETTING H2 MODUL
;
      LD       T 0      TIMERO   TIMER FOR COMMON USE
              2
      STH      T 0      TIMERO   TIMER FOR COMMON USE
      JR       H -1
;-----
      EXOB
;
;DEFINITION OF CONTROL INPUTS
;=====
;
;INPUT 0 :SELECT OPERATION MODE  OFF->AUTO  ON->MANUAL CONTROL
;INPUT 1 :START AUTO SEQUENCE
;INPUT 2 :MANUAL FORWARD
;INPUT 3 :MANUAL BACKWARDS
;INPUT 4 :MANUAL MOVE OUT OF LIMIT SWITCH
;

```

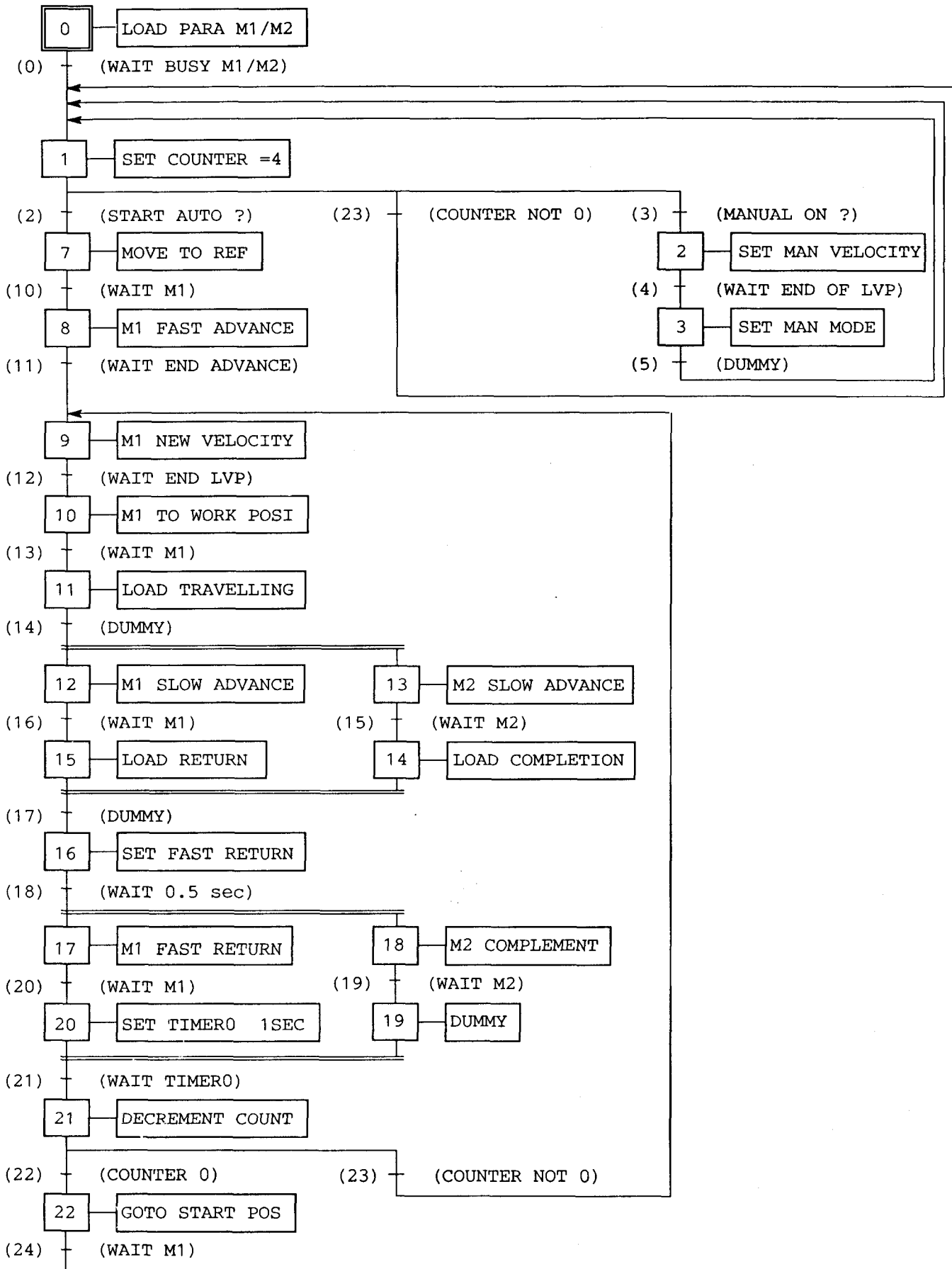


```

;PROGRAM ORGANISATION BLOCK
;=====
;
;- COB 0 READS OPERATION MODE AND CALLS PB 0 FOR MANUAL
; OPERATION AND SB 0 FOR THE AUTO SEQUENCE
;- IN THIS APPLICATION MOTOR PARAMETERS ARE DIFFERENT ON BOTH
; OPERATION MODES
;- MANUAL CONTROL IS ONLY ACTIV FOR MOTOR 1
;
;*****
COB 0
0
;-----
CSB 0 ACONTROL AUTO CONTROL
;-----
STH F 500 MANFLAG FLAG IS SET IF MANUAL OPERATION
;-----
CPB H 0 MCONTROL MANUAL CONTROL
;-----
;
;SWITCH TO AUTO IF NOT MANUAL
;
STH I 0 MANUAL SWITCH FOR MANUAL CONTROL
JR H end
STH F 500 MANFLAG FLAG IS SET IF MANUAL OPERATION
SEI K 1 M1 PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB 824 HLTM HALT ANY MOTION
CFB 823 AUTOCON SWITCH TO AUTO CONTROL
RES F 500 MANFLAG FLAG IS SET IF MANUAL OPERATION
;
;END OF OPERATING MODE SELECTION
;
ECOB

```

SB-NUMBER: 0 AUTO CONTROL in GRAFTEC



```

;***** AUTOCONTROL in GRAFTEC
SB      0      ACONTROL  AUTO CONTROL
;-----
;  STEPS
;-----
IST      0      LOAD PARA M1/M2
          O 0      WAIT BUSY M1/M2
SEI      K 1      M1      PARAMETER MOTOR 1 (BLOCK OF 8 REG)
LDX      R 2      PARA+2  PARAMETER BLOCK (0-> BASE ADDRESS)
          500
LDX      R 3      PARA+3  PARAMETER BLOCK (0-> BASE ADDRESS)
          10000
LDX      R 4      PARA+4  PARAMETER BLOCK (0-> BASE ADDRESS)
          50
CFB      801      LVP      LOAD VELOCITY PROFILE
SEI      K 9      M2      PARAMETER MOTOR 2 (BLOCK OF 8 REG)
LDX      R 2      PARA+2  PARAMETER BLOCK (0-> BASE ADDRESS)
          40
LDX      R 3      PARA+3  PARAMETER BLOCK (0-> BASE ADDRESS)
          5000
LDX      R 4      PARA+4  PARAMETER BLOCK (0-> BASE ADDRESS)
          50
CFB      801      LVP      LOAD VELOCITY PROFILE
RES      F 500    MANFLAG  FLAG IS SET IF MANUAL OPERATION
EST
;-----
ST      1      SET COUNTER =4
          I 0      WAIT BUSY M1/M2
          I 24     WAIT M1
          I 1      MANFLAG ON ?
          I 5      DUMMY
          O 2      START AUTO ?
          O 1      MANFLAG ON ?
          O 3      MANUAL ON ?
LD      C 101    COUNTER  COUNTER FOR CYCLE
          4
EST
;-----
ST      2      SET MAN VELOCITY
          I 3      MANUAL ON ?
          O 4      WAIT END OF LVP
SEI      K 1      M1      PARAMETER MOTOR 1 (BLOCK OF 8 REG)
LDX      R 3      PARA+3  PARAMETER BLOCK (0-> BASE ADDRESS)
          1200
CFB      801      LVP      LOAD VELOCITY PROFILE
EST
;-----

```

```

      ST      3          SET MAN MODE
              I 4          WAIT END OF LVP
              O 5          DUMMY
      SEI     K 1      M1    PARAMETER MOTOR 1 (BLOCK OF 8 REG)
      CFB     822      MANCON SWITCH TO MANUAL CONTROL
      SET     F 500    MANFLAG FLAG IS SET IF MANUAL OPERATION
      EST

;-----
      ST      7          MOVE TO REFERENCE
              I 2          START AUTO ?
              O 10         WAIT M1
      SEI     K 1      M1    PARAMETER MOTOR 1 (BLOCK OF 8 REG)
      CFB     811      REF   MOVE TO REFERENCE SWITCH
      EST

;-----
      ST      8          M1 FAST ADVANCE
              I 10         WAIT M1
              O 11         WAIT END ADVANCE
      SEI     K 1      M1    PARAMETER MOTOR 1 (BLOCK OF 8 REG)
      LDX     R 6      PARA+6 PARAMETER BLOCK (0-> BASE ADDRESS)
              100000
      CFB     819      DAPOS  DISPLAY ACTUAL POSITION
      CFB     803      LAPOS  LOAD ACTUAL POSITION
      LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
              115000
      CFB     804      LDEST  LOAD DESTINATION ABSOLUTE
      CFB     810      PHF    POSITION WITH HIGH FREQUENCY
      EST

;-----
      ST      9          M1 NEW VELOCITY
              I 11         WAIT END ADVANCE
              I 23         COUNTER NOT 0
              O 12         WAIT END LVP
      SEI     K 1      M1    PARAMETER MOTOR 1 (BLOCK OF 8 REG)
      CFB     817      RAPOS  READ ACTUAL POSITION
      CFB     819      DAPOS  DISPLAY ACTUAL POSITION
      LDX     R 3      PARA+3 PARAMETER BLOCK (0-> BASE ADDRESS)
              2000
      CFB     801      LVP    LOAD VELOCITY PROFILE
      EST

;-----
      ST      10         M1 TO WORK POSITION
              I 12         WAIT END LVP
              O 13         WAIT M1
      SEI     K 1      M1    PARAMETER MOTOR 1 (BLOCK OF 8 REG)
      LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
              120000
      CFB     804      LDEST  LOAD DESTINATION ABSOLUTE
      CFB     810      PHF    POSITIONING WITH HIGH FREQUENCY
      EST

```

```

-----
ST      11          LOAD TRAVELLING
        I 13          WAIT M1
        O 14          DUMMY
SEI     K 1      M1   PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     817      RAPOS READ ACTUAL POSITION
CFB     819      DAPOS DISPLAY ACTUAL POSITION
LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
        2500
CFB     805      LDF   LOAD DIFFERENCE FORWARD
SEI     K 9      M2   PARAMETER MOTOR 2 (BLOCK OF 8 REG)
LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
        200
CFB     805      LDF   LOAD DIFFERENCE FORWARD
EST

```

```

-----
ST      12          M1 SLOW ADVANCE
        I 14          DUMMY
        O 16          WAIT M1
SEI     K 1      M1   PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     809      PSSF POSITIONING WITH START/STOP FREQUEN
EST

```

```

-----
ST      13          M2 SLOW ADVANCE
        I 14          DUMMY
        O 15          WAIT M2
SEI     K 9      M2   PARAMETER MOTOR 2 (BLOCK OF 8 REG)
CFB     809      PSSF POSITIONING WITH START/STOP FREQUEN
EST

```

```

-----
ST      14          LOAD COMPLETION
        I 15          WAIT M2
        O 17          DUMMY
SEI     K 9      M2   PARAMETER MOTOR 2 (BLOCK OF 8 REG)
LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
        300
CFB     805      LDF   LOAD DIFFERENCE FORWARD
EST

```

```

-----
ST      15          LOAD RETURN
        I 16          WAIT M1
        O 17          DUMMY
SEI     K 1      M1   PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     817      RAPOS READ ACTUAL POSITION
CFB     819      DAPOS DISPLAY ACTUAL POSITION
LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
        115000
CFB     804      LDEST LOAD DESTINATION ABSOLUTE
EST
-----

```

```

ST      16          SET FAST RETURN
        I 17          DUMMY
        O 18          WAIT 0.5 sec
LD      T 0    TIMER0  TIMER FOR COMMON USE
        5
SEI     K 1    M1      PARAMETER MOTOR 1 (BLOCK OF 8 REG)
LDX     R 3    PARA+3  PARAMETER BLOCK (0-> BASE ADDRESS)
        10000
CFB     801    LVP     LOAD VELOCITY PROFILE
EST
;-----;
ST      17          M1 FAST RETURN
        I 18          WAIT 0.5 sec
        O 20          WAIT M1
SEI     K 1    M1      PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     810    PHF     PSITIONING WITH HIGH FREQUENCY
EST
;-----;
ST      18          M2 COMPLEMENT
        I 18          WAIT 0.5 sec
        O 19          WAIT M2
SEI     K 9    M2      PARAMETER MOTOR 2 (BLOCK OF 8 REG)
CFB     810    PHF     POSITIONING WITH HIGH FREQUENCY
EST
;-----;
ST      19          DUMMY
        I 19          WAIT M2
        O 21          WAIT TIMER0
EST
;-----;
ST      20          SET TIMER0 1SEC
        I 20          WAIT M1
        O 21          WAIT TIMER0
LD      T 0    TIMER0  TIMER FOR COMMON USE
        10
EST
;-----;
ST      21          DECREMENT COUNTER
        I 21          WAIT TIMER0
        O 22          COUNTER 0
        O 23          COUNTER NOT 0
DEC     C 101  COUNTER  COUNTER FOR CYCLE
EST
;-----;

```

```

ST      22          GOTO START POS
        I 22        COUNTER 0
        O 24        WAIT M1
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     817      RAPOS READ ACTUAL POSITION
CFB     819      DAPOS DISPLAY ACTUAL POSITION
LDX     R 5      PARA+5 PARAMETER BLOCK (0-> BASE ADDRESS)
        100250
CFB     804      LDEST LOAD DESTINATION ABSOLUTE
CFB     810      PHF   POSITIONING WITH HIGH FREQUENCY
EST

```

```

;-----
;
; TRANSITIONS
;-----

```

```

TR      0          WAIT BUSY M1/M2
        I 0        LOAD PARA M1/M2
        O 1        SET COUNTER =4
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"- BUSY)
SEI     K 9      M2  PARAMETER MOTOR 2 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
ANH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"- BUSY)
ETR

```

```

;-----
TR      1          MANFLAG ON ?
        I 1        SET COUNTER =4
        O 1        SET COUNTER =4
STH     F 500    MANFLAG FLAG IS SET IF MANUAL OPERATION
ETR

```

```

;-----
TR      2          START AUTO ?
        I 1        SET COUNTER =4
        O 7        MOVE TO REFERENCE
STH     I 1      START START CONDITION
ETR

```

```

;-----
TR      3          MANUAL ON ?
        I 1        SET COUNTER =4
        O 2        SET MAN VELOCITY
STH     I 0      MANUAL SWITCH FOR MANUAL CONTROL
ETR
;-----

```

```

TR      4          WAIT END OF LVP
        I 2          SET MAN VELOCITY
        O 3          SET MAN MODE
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"- BUSY)
ETR

```

```

-----
TR      5          DUMMY
        I 3          SET MAN MODE
        O 1          SET COUNTER =4
ETR

```

```

-----
TR      10         WAIT M1
        I 7          MOVE TO REFERENCE
        O 8          M1 FAST ADVANCE
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR

```

```

-----
TR      11         WAIT END ADVANCE
        I 8          M1 FAST ADVANCE
        O 9          M1 NEW VELOCITY
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR

```

```

-----
TR      12         WAIT END LVP
        I 9          M1 NEW VELOCITY
        O 10         M1 TO WORK POSITION
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR

```

```

-----
TR      13         WAIT M1
        I 10         M1 TO WORK POSITION
        O 11         LOAD TRAVELLING
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816      RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR

```

```

-----
TR      14         DUMMY
        I 11         LOAD TRAVELLING
        O 12         M1 SLOW ADVANCE
        O 13         M2 SLOW ADVANCE
ETR

```



```

;-----
TR      15          WAIT M2
        I 13        M2 SLOW ADVANCE
        O 14        LOAD COMPLETION
SEI     K 9      M2  PARAMETER MOTOR 2 (BLOCK OF 8 REG)
CFB     816     RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"- BUSY)
ETR

```

```

;-----
TR      16          WAIT M1
        I 12        M1 SLOW ADVANCE
        O 15        LOAD RETURN
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816     RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR

```

```

;-----
TR      17          DUMMY
        I 15        LOAD RETURN
        I 14        LOAD COMPLETION
        O 16        SET FAST RETURN
ETR

```

```

;-----
TR      18          WAIT 0.5 sec
        I 16        SET FAST RETURN
        O 17        M1 FAST RETURN
        O 18        M2 COMPLEMENT
STL     T 0      TIMER0  TIMER FOR COMMON USE
ETR

```

```

;-----
TR      19          WAIT M2
        I 18        M2 COMPLEMENT
        O 19        DUMMY
SEI     K 9      M2  PARAMETER MOTOR 2 (BLOCK OF 8 REG)
CFB     816     RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR

```

```

;-----
TR      20          WAIT M1
        I 17        M1 FAST RETURN
        O 20        SET TIMER0 1SEC
SEI     K 1      M1  PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816     RBUSY READ BUSY STATUS
STH     F 10     BUSYFLAG COMMON FLAG FOR BUSY ("0"-> BUSY)
ETR
;-----

```

```

TR      21          WAIT TIMERO
        I 20        SET TIMERO 1SEC
        I 19        DUMMY
        O 21        DECREMENT COUNTER
STL     T 0    TIMER0  TIMER FOR COMMON USE
ETR
-----;
TR      22          COUNTER 0
        I 21        DECREMENT COUNTER
        O 22        GOTO START POS
STL     C 101  COUNTER  COUNTER FOR CYCLE
ETR
-----;
TR      23          COUNTER NOT 0
        I 21        DECREMENT COUNTER
        O 9         M1 NEW VELOCITY
STH     C 101  COUNTER  COUNTER FOR CYCLE
ETR
-----;
TR      24          WAIT M1
        I 22        GOTO START POS
        O 1         SET COUNTER =4
SEI     K 1    M1     PARAMETER MOTOR 1 (BLOCK OF 8 REG)
CFB     816    RBUSY  READ BUSY STATUS
STH     F 10   BUSYFLAG COMMON FLAG FOR BUSY ("0"- BUSY)
ETR
-----;
ESB

```

```

;
;PROGRAM BLOCK FOR MANUAL OPERATION
;=====
;
;*****
      PB      0      MCONTROL  MANUAL CONTROL
;-----
      SEI     K 1     M1        PARAMETER MOTOR 1 (BLOCK OF 8 REG)
;
;MANUAL FORWARD
      STH     I  2     FOR       SWITCH FOR MANUAL FORWARD
      DYN     F 502    DYNF      DYN FLAG 502-520
      CFB     H 830    HFF       MANUAL WITH HF FORWARD
      STL     I  2     FOR       SWITCH FOR MANUAL FORWARD
      DYN     F 503    DYNF+1    DYN FLAG 502-520
      CFB     H 824    HLTM      HALT ANY MOTION
;
;MANUAL BACKWARDS
      STH     I  3     BACK      SWITCH FOR MANUAL BACKWARDS
      DYN     F 504    DYNF+2    DYN FLAG 502-520
      CFB     H 831    HFB       MANUAL WITH HF BACHWARDS
      STL     I  3     BACK      SWITCH FOR MANUAL BACKWARDS
      DYN     F 505    DYNF+3    DYN FLAG 502-520
      CFB     H 824    HLTM      HALT ANY MOTION
;
; MANUAL OUTLS (BECAUSE OUTLS IS AN AUTO FUNCTION THIS ROUTINE
; SWITCHES TO AUTOMATIC, CALLS THE OUTLS FUNCTION AN THEN COMES
; BACK TO MANUAL CONTROL)
;
      STH     I  4     OUTLIMIT  SWITCH TO MOVE OUT OF LIMIT SWITCH
      DYN     F 506    DYNF+4    DYN FLAG 502-520
      SET     F 507    DYNF+5    DYN FLAG 502-520
      CFB     H 823    AUTOCON   SWITCH TO AUTOMATIC CONTROL
      CFB     H 814    OUTLS     MOVE OUT OF LIMIT SWITCH
      STH     F 507    DYNF+5    DYN FLAG 502-520
      CFB     H 816    RBUSY     READ BUSY STATUS
      ANH     F 10     BUSYFLAG  COMMON FLAG FOR BUSY ("0"- BUSY)
      CFB     H 822    MANCON    SWITCH TO MANUAL CONTROL
      RES     F 507    DYNF+5    DYN FLAG 502-520
;-----
      EPB

```


10. Alphabetische Symbol-Übersicht

| Name | Typ | Seite |
|-----------------|-----|-------|
| A ACCEL | R | |
| APOS | R | |
| AutoCon | FB | 7-35 |
| B BASE | R | |
| BUSYFLAG | F | |
| C CAV | FB | 7-6 |
| CH_0 | C | |
| ComErr | F | |
| Cont | FB | 7-37 |
| CONTAD | FB | 6-6 |
| D DAPos | FB | 7-30 |
| DPB14 | FB | 7-31 |
| DP14 | FB | 7-32 |
| DPLM | F | |
| F FH_0 | F | |
| H HF | R | |
| HFB | FB | 7-40 |
| HFF | FB | 7-40 |
| HltM | FB | 7-36 |
| I INSTR | FB | 6-6 |
| L LAPos | FB | 7-7 |
| LDB | FB | 7-11 |
| LDest | FB | 7-8 |
| LDF | FB | 7-10 |
| LSRef | FB | 7-20 |
| LVP | FB | 7-5 |
| M ManCon | FB | 7-34 |
| O OSB | FB | 7-38 |
| OSF | FB | 7-38 |
| OutOR | FB | 7-24 |
| OutLS | FB | 7-22 |
| P PHF | FB | 7-15 |
| POS | R | |
| PSSF | FB | 7-14 |

| Name | Typ | Seite |
|----------|-----|-------|
| R | | |
| RAPos | FB | 7-27 |
| RBO | FB | 6-6 |
| RBusy | FB | 7-26 |
| RBYTE | FB | 6-6 |
| RDATA | FB | 6-6 |
| Ref | FB | 7-16 |
| RefLS | FB | 7-18 |
| REG_1 | R | |
| REG_2 | R | |
| RH_0 | R | |
| RINDEX | R | |
| RSMC | FB | 7-4 |
| RStatus | FB | 7-28 |
| S | | |
| SETAD | FB | 6-6 |
| SiSB | FB | 7-13 |
| SiSF | FB | 7-12 |
| SSFB | FB | 7-39 |
| SSFF | FB | 7-39 |
| SSF | R | |
| STATUS | R | |
| SWPULS | FB | 6-6 |
| T | | |
| TEMPREG | R | |
| TRAFLAG | F | |
| TRAREG | R | |
| W | | |
| WBYTE | FB | 6-6 |
| WDATA2 | FB | 6-6 |
| WDATA3 | FB | 6-6 |
| X | | |
| XYFLAG | F | |
| XY | R | |