



Titel Stepmotor Controller EL734

Ersetzt:

Autoren/  
Autorinnen H. Senn

Erstellt: 24. Nov. 2006  
Druckdatum: 22. Dezember 2006

**Zusammenfassung:**

Der Stepmotorcontroller EL734 wurde entwickelt für Positionierantriebe in Neutronen-Spektrometern an der SINQ.

- modulares System für 1 .. 12 Schrittmotor-Achsen in 1 .. 4 Einheiten 19`` / 3HE
- 2-, 3,- oder 5-Phasen-Schrittmotoren
- Kommunikation mit Host über RS232-Schnittstelle
- Bedienungstasten und eine zweite RS232-Schnittstelle auf Gerätefront für Service- und Einstellarbeiten
- Positionserfassung wahlweise mit dem Motorschritt-Zähler oder einem zusätzlichen Positionsgeber
- Inkremental-Geber mit RS422-Schnittstelle, Absolut-Digitizer mit SSI-Schnittstelle ( 1 .. 25 Bit ) oder Absolutgeber mit TTL-Parallelschnittstelle ( 1 .. 17 Bit )
- automatische Luftkissen- oder Magnetbremsen-Steuerung
- pro Achse zusätzlich je ein unabhängiger digitaler Ein- und Ausgang ( 24V )
- Maximale Motor-Schrittfrequenz 33 .. 20000 Hz
- die Synchronisation mehrerer Achsen ist nicht vorgesehen

Verteiler	Abt.	Empfänger / Empfängerinnen	Expl.	Abt.	Empfänger / Empfängerinnen	Expl.		
								Expl.
							Bibliothek	
							Reserve	
							Total	
							Seiten	
							Beilagen	
							Informationsliste	
							D 1 2 3 4 5 8 9 A	
							Visum Abt./Laborleitung:	

# Inhalt

<b>INHALT</b> .....	<b>2</b>
<b>ALLGEMEINES</b> .....	<b>5</b>
<b>SPEZIFIKATIONEN</b> .....	<b>6</b>
<b>BETRIEBSARTEN</b> .....	<b>7</b>
BEDIENUNG .....	7
POSITIONIER-BETRIEBSARTEN .....	7
<b>MECHANISCHER AUFBAU</b> .....	<b>8</b>
<b>ELEKTRISCHE ANSCHLÜSSE</b> .....	<b>9</b>
VME-STEUEREINHEIT .....	9
LEISTUNG-EINHEIT 4 x 2PH/5PH.....	9
LEISTUNG-EINHEIT 8 x 3PH.....	9
KOMPAKTSTEUERUNG .....	9
<b>INBETRIEBNAHME</b> .....	<b>10</b>
VORBEREITUNG .....	10
GRUNDEINSTELLUNGEN.....	11
KONFIGURATION CLOSEDLOOP .....	11
SETUP INKREMENTALSYSTEM.....	13
PARAMETEREINSTELLUNGEN.....	13
<b>BETRIEB</b> .....	<b>15</b>
SYSTEMSTART .....	15
STEUERUNG ÜBER SERIELLE SCHNITTSTELLE .....	15
LOKALE TASTENBEDIENUNG .....	16
<b>INSTALLATION</b> .....	<b>17</b>
GERÄTE-EINBAU .....	18
BLOCKSCHEMA.....	19
ANSCHLUSS VME-STEUERTEIL .....	20
ANSCHLUSS LEISTUNGSTEIL.....	21
<b>STECKERBELEGUNG</b> .....	<b>22</b>
VME-STEUERTEIL.....	22
<i>RS232-Schnittstellen</i> .....	22
<i>Testbuchsen</i> .....	22
<i>Display</i> .....	22
LEISTUNGSTEIL.....	23
<i>Motorstecker</i> .....	23
<i>Stecker 24V</i> .....	23
<i>Stecker Steuerung Input</i> .....	24
<i>Stecker Steuerung Output</i> .....	25
<i>Klemmenblock Steuerung Input</i> .....	26
<i>Klemmenblock Steuerung Output</i> .....	26
ENCODER-ANSCHLÜSSE .....	27
<i>Verteilerblock Inkrementalgeber</i> .....	27
<i>Verteilerblock Absolutgeber m. SSI-Interface</i> .....	27
<i>Anschluss am DMS SSI-Interface</i> .....	27
<i>Anschluss am SSI4-Interface</i> .....	28

<i>Verteilerblock Absolutgeber Litton</i> .....	28
<b>VME-STEUREINHEIT</b> .....	<b>29</b>
VME-MODULE .....	30
<i>CPU</i> .....	31
<i>VMIO</i> .....	33
<i>MOT</i> .....	34
<i>DSP</i> .....	34
<i>INC</i> .....	36
<i>AEC</i> .....	37
<i>SSI4</i> .....	38
<i>SSI</i> .....	39
MODULADRESSEN .....	41
<i>Moduladressen Tabelle</i> .....	42
<b>LEISTUNGSTEIL</b> .....	<b>43</b>
LEISTUNGSEINHEIT 4 x 2PH / 5PH .....	43
LEISTUNGSEINHEIT 8 x 3PH .....	45
KOMPAKTSTEUERUNG .....	46
MODULE DER LEISTUNGSEINHEIT .....	47
<i>IO</i> .....	47
<i>3Ph</i> .....	50
<i>5Ph</i> .....	51
<i>TWINLINE</i> .....	52
VERBINDUNGSKABEL .....	54
<i>Netzanschluss</i> .....	54
<i>Motoranschluss</i> .....	54
<i>Anschluss Steuerung Ausgänge</i> .....	54
<i>Anschluss Steuerung Eingänge</i> .....	54
<i>24V</i> .....	54
<b>SOFTWARE</b> .....	<b>55</b>
FUNKTION DER SOFTWARE .....	55
<i>Systemstart</i> .....	56
<i>Monitorloop</i> .....	56
<b>BEFEHLSSATZ</b> .....	<b>57</b>
BEFEHLSSYNTAX .....	58
BEFEHLE IM DETAIL .....	59
<i>A</i> .....	59
<i>AC</i> .....	59
<i>ACS</i> .....	60
<i>AM</i> .....	60
<i>CFG</i> .....	60
<i>CLS</i> .....	61
<i>D</i> .....	61
<i>DE</i> .....	61
<i>E</i> .....	62
<i>EC</i> .....	62
<i>ECHO</i> .....	63
<i>EP</i> .....	63
<i>ERR</i> .....	64
<i>F</i> .....	64
<i>FB</i> .....	64
<i>FF</i> .....	64
<i>FD</i> .....	65
<i>FM</i> .....	65
<i>G</i> .....	65
<i>H</i> .....	66

<i>ID</i> .....	66
<i>J</i> .....	66
<i>K</i> .....	67
<i>L</i> .....	67
<i>M</i> .....	68
<i>MEM</i> .....	68
<i>MN</i> .....	68
<i>MSR</i> .....	68
<i>N</i> .....	69
<i>OPT</i> .....	69
<i>P</i> .....	70
<i>PD</i> .....	70
<i>PR</i> .....	71
<i>Q</i> .....	71
<i>R</i> .....	71
<i>RF</i> .....	71
<i>RI</i> .....	72
<i>RMT</i> .....	72
<i>S</i> .....	73
<i>SA</i> .....	73
<i>SN</i> .....	73
<i>SP</i> .....	73
<i>SR</i> .....	73
<i>ST</i> .....	73
<i>SW</i> .....	73
<i>SB</i> .....	74
<i>SF</i> .....	74
<i>SO</i> .....	74
<i>SS</i> .....	75
<i>SYR</i> .....	75
<i>T</i> .....	76
<i>U</i> .....	76
<i>UD</i> .....	76
<i>UU</i> .....	77
<i>V</i> .....	77
<i>W</i> .....	77
<i>Z</i> .....	78
<i>?</i> .....	78
FEHLER- UND STATUS-MELDUNGEN .....	79
<b>ANHANG .....</b>	<b>80</b>
LUFTKISSENSTEUERUNG .....	80
MAGNETBREMSSTEUERUNG .....	81
GETRIEBEFAKTOREN BESTIMMEN .....	81
REFERENZVERFAHREN .....	86
VARIANTEN UND SPEZIALAUSFÜHRUNGEN DER MOTORSTEUERUNG .....	90
<i>μEI</i> .....	90
EXTERNE DOKUMENTE .....	91

## Allgemeines

Mit dem Stepmotorcontroller-System EL734 können bis 12 Schrittmotorachsen über eine RS232-Schnittstelle gesteuert werden. Je nach Ausrüstung können 2-, 3- oder 5-Phasenschrittmotoren eingesetzt werden. Die Positionserfassung kann wahlweise über den Motorschritt-Zähler und Endschalter bzw. Referenzpunkt-Schalter, oder durch verschiedene Typen von Encodern erfolgen. Die Synchronisation mehrerer Achsen ist nicht möglich.

Für jede Motorachse steht eine automatische Luftkissen- oder Magnetbremsen-Steuerung, sowie je ein unabhängiger Steuer-Eingang und ein Schalt-Ausgang (24V) für beliebige Aufgaben zur Verfügung.

Für die Kommunikation mit einem Hostcomputer ist der Stepmotorcontroller mit einer RS232-Schnittstelle ausgerüstet. Über diese Verbindung werden Steuerbefehle übermittelt, Parameterwerte eingestellt, sowie Parameter oder Betriebszustände abgefragt.

Für Service- und Einstell-Arbeiten steht eine zweite RS232-Schnittstelle zum Anschluss eines Terminals oder Notebooks mit identischen Funktionen auf der Frontseite zur Verfügung.

Für die Kommunikation sind zwei Betriebsarten für jede Schnittstelle separat einstellbar

- Hostmode für den Anschluss an Host-Computer in automatisierten Systemen
- Terminalmode ( default ) für den manuellen Betrieb mit Terminal oder Notebook  
Ausgabe mit Zeilenvorschub und Zeichen-Echo, zus. Textinformationen

Im Offline-Zustand der Steuerung ( Lokalbetrieb ) kann eine vorwählbare Achse über die eingebaute Tastenbedienung frei bewegt werden. Die Vorgabe von Zielpositionen mit den Tasten ist nicht vorgesehen.

Eine komplette Anlage besteht aus einer VME-Steuereinheit und, je nach Anzahl benötigter Motorachsen, 1 bis 3 Leistungseinheiten

Ausrüstung mit Interface-Modulen und Motor-Endstufen nach Bedarf

Die Motoren werden direkt an den Motorsteckdosen der Leistungseinheiten angeschlossen.

Für den Anschluss von Encodern sind im Schrank montierbare Steckverteiler vorgesehen.

Die Ein- und Ausgangs-Steuerleitungen der Steuerung (Endschalter, Luftkissensteuerung etc.) werden auf im im Schaltschrank montierte Klemmenverteiler geführt, was einen individuellen Anschluss der externen Apparate erlaubt.



## Betriebsarten

### Bedienung

- Der Stepmotorcontroller wird über die RS232 Host-Schnittstelle von einem Host-Computer aus bedient.  
Es können Positionier- und Fahrbefehle gegeben, Parameterwerte gesetzt, oder Positions-, Status- und Parameter-Werte ausgelesen werden.
- Alternativ ist die Bedienung über die auf der Frontseite befindliche Service-Schnittstelle mittels Terminal oder Computer möglich. Die Service-Schnittstelle bietet die selben Bedienungs-möglichkeiten wie die Host-Schnittstelle.
- Die aktuellen Achspositionen und Parameterwerte werden laufend in einem batteriegestützten RAM netzausfallsicher gespeichert.
- Im Offline-Modus (unmittelbar nach dem Aufstarten oder nach Offlinesetzen durch Kommando) kann eine vorwählbare Achse über Tasten an der Steuerung im Lokalbetrieb bewegt werden.

### Positionier-Betriebsarten

- **Openloop** Ohne Positionsmessung durch Encoder.  
Durch Referenzlauf an einen Referenz- oder Endschalter muss der Nullpunkt der Positionsmessung definiert werden.  
Störungen durch mechanische Überlastung oder Blockierung des Antriebs werden durch die Steuerung nicht automatisch erkannt.  
Netzausfälle während laufender Positionierung oder Schrittverlust durch Überlastung haben den Verlust des korrekten Ist-Positionswertes zur Folge.
- **Closedloop** Mit Positionsmessung durch Winkel- oder lineare Mess-Geber.  
Im ClosedLoop vergleicht das System nach dem Positionierlauf den erreichten Positionswert mit dem Sollwert und führt bei Bedarf selbständig Korrekturen durch. ( keine kontinuierliche Nachführung oder Regelung wie bei Servosystemen )  
Störungen durch mechanische Überlastung oder Blockierung des Antriebs kann die Steuerung automatisch erkennen.

Beim Einsatz eines **Inkrementalgebers** ist zur Initalisierung ein Referenzlauf zu einem Referenz- oder Endschalter notwendig.

Ein Netzunterbruch während einem laufenden Positionierlauf kann eine Abweichung des Positionswertes zur Folge haben.

Beim Einsatz eines **Absolutgebers** ist kein Referenzlauf notwendig.

Bei der Definition eines Positionswertes wird automatisch die Position des Encodernullpunktes berechnet und als Parameter gespeichert. Als Absolutgeber können mechanische Pin-Encoder (Litton) mit V-Abtastung und Parallel-Schnittstelle oder elektronische Encoder mit SSI-Schnittstelle (Heidenhain, Baumer, Stegmann etc.) eingesetzt werden.

**Anmerkung** : Die Litton Pin-Encoder sind nicht mehr lieferbar. Es ist kein Ersatz verfügbar !



## Elektrische Anschlüsse

### VME-Steuereinheit

Netzanschluss	Gerätestecker 2P + E 230V mit Netzschalter und Sicherung	Rückseite
Host-Schnittstelle	D-Sub Stecker 25P m	Rückseite
Service-Schnittstelle	D-Sub Stecker 25P m	Frontseite
Testbuchsen	2mm Buchsen sw 0V, rt +5V	Frontseite

### Leistungseinheit 4 x 2Ph/5Ph

Netzanschluss	Gerätestecker 2P + E 230V mit Netzschalter und Sicherung	Rückseite
Motor-Stecker	Burndy Metalok Bantam 12P w	Rückseite
24V-Stecker	Burndy Metalok Bantam 4P w	Rückseite
Steuer-Ausgänge	Flachband-Stecker 50P m	Rückseite
Steuer-Eingänge	Flachband-Stecker 64P m	Rückseite

### Leistungseinheit 8 x 3Ph

Netzanschluss	Gerätestecker 2P + E 230V mit Netzschalter und Sicherung	Rückseite
Motor-Stecker	LEMO2 4P	Rückseite
Steuerung	je ein steckbarer Klemmenblock pro Achse	Rückseite

### Kompaktsteuerung

Die Kompaktsteuerung ist die Kombination einer reduzierten VME-Steuereinheit mit einer reduzierten Leistungseinheit für 4 x 2Ph bzw. 2 x 5Ph.

Die Funktion und die Anschlüsse sind mit identisch mit der Normalsteuerung.

# Inbetriebnahme

## Vorbereitung

Fehlerhafte Anschlüsse können Schäden an Motoren, Endstufen, Interface-Modulen oder Encodern verursachen. Daher müssen alle Verbindungen der Installation auf korrekten Anschluss überprüft werden, bevor die Netzspeisung erstmals eingeschaltet wird.

Insbesondere muss darauf geachtet werden, dass die Endstufenmodule ihren Typen entsprechend gesteckt sind ( keine Verwechslung von 3Ph- oder 5Ph- mit 2Ph-Endstufen oder umgekehrt ) .

Für die Inbetriebnahme der Steuerung wird die Verbindung zum Host abgetrennt.

Die einzelnen Achsen können wahlweise über den Terminalemulator eines PCs am Serviceanschluss oder direkt mit den Tasten der Displaymodule bedient werden.

Für die Inbetriebnahme werden zweckmässigerweise für alle Achsen Display-Module im VME-Steuerteil eingesetzt. ( sie können später ev. wieder entfernt werden )

- Alle Schalter auf den IO-Modulen **Pwr. Off** stellen
- Phasenstrom an allen 3Ph- und 5Ph-Endstufenmodulen auf Minimumwert einstellen
- Übrige Einstellungen der Endstufenmodule überprüfen
- Netzspeisung einschalten
- Nach der Systemstart zeigen alle alle Displays 0000.000
- An den IO-Interfaces leuchten nur die roten **Disabel-** und die **Pwr Off\_LEDs**

## Grundeinstellungen

- Endschalter einzeln manuell ( direkt mechanisch, induktive Näherungsschalter mit einem Metallteil z.B. Schraubenzieherklinge ) betätigen und die Schaltzustände an den LEDs der IO-Module überprüfen
- Motorstrom der aktuellen Achse einschalten ( Schalter auf dem IO-Modul => **Normal** )
- Durch Magnetbremsen oder Luftkissen blockierte Antriebe müssen vor dem bewegen freigemacht werden durch aktivieren des Luftkissen-Schaltausgangs mit Befehl **AC m m** oder durch Konfiguration der automatischen Funktion mit dem Parameter **W m 1**. Bei Luftkissen Verkettungsregeln beachten !  
Die Anwendung der Luftkissen- oder Magnetbremsensteuerung ist im **Anhang Anhang**
- **Luftkissensteuerung** bzw. **Magnetbremsensteuerung** beschrieben.
- Achse vorsichtig in Einzelschritten od. mit Start/Stop-Frequenz bewegen und die Bewegungsrichtung des Antriebs feststellen.  
Wenn das Motordrehmoment zum bewegen nicht ausreicht, muss der Phasenstromwert am Endstufenmodul höher eingestellt werden ( Maximalstrom des Motors beachten ! )
- Wenn sich der Antrieb in Bezug zu den Endschaltern in falscher Richtung bewegt, muss der Motordreh Sinn geändert werden
  - 5Ph-Motor                      DIP-Switch 3 auf dem Endstufenmodul umschalten
  - 2Ph- oder 3Ph-Motor      2 Phasenanschlüsse tauschen oder den zugehörigen Richtungs-Jumper im IO-Modul umstecken
- Test der Endschaltefunktionen durch Anfahren und Freifahren der Schalter.  
Ev. müssen vorher die Softwarelimiten verändert werden. ( Parameter **H** )
- Motor-Faktor zur Umrechnung von Längen- bzw. Winkel-Einheiten in Motorschritten ( **MS** ) ermitteln und eingeben ( Parameter **FM** ).  
Berechnungsbeispiele im **Anhang Getriebefaktoren**
- Ist-Position ( Parameter **U** ) und Softwarelimiten ( Parameter **H** ) eingeben
- Positionier-Test mit Ausmessen der Fahrdistanz

Der Antrieb ist nun bereit für OpenLoop-Betrieb

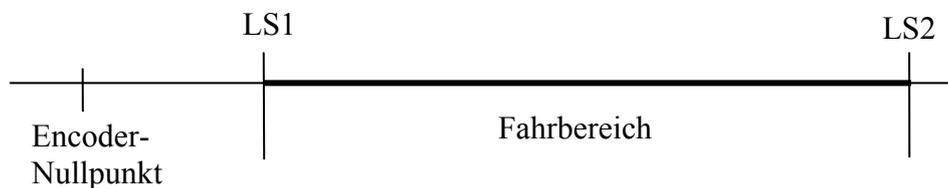
## Konfiguration ClosedLoop

Damit die Positionsmessung unabhängig vom Antrieb konfiguriert und getestet werden kann, wird der Encoder vorerst einer freien Achse  $m_e$  zugeordnet ( z. B. wird der Geber der zu testenden Antriebsachse **M2** auf die unbenützte Achse **M4** gelegt ).

- Parameter der Encoder-Achse  $m_e$  default setzen **DE  $m_e$**
- **EC  $m_e$  t n** ordnet der Motorachse  $m_e$  den Encoder **n** vom Interface-Typ **t** zu
 

<b>m</b>	1 .. 12
<b>n</b>	1 .. 12
<b>t</b>	0 kein Encoder, 1 <b>INC</b> , 2 <b>ABS</b> , 3 <b>SSI</b> , 4 <b>SSI4</b>

- Interfacemapping und Geberzuordnung anzeigen und prüfen mit **CFG**
- Für Geber mit **SSI**- od. **SSI4**-Interface Bit-Zahl und Codeart setzen ( Parameter **EP** )
- Der für die Achse  $m_e$  angezeigte Positionswert ist, ohne Dezimalpunkt gelesen, die Position Encoders in Encoder-Steps ( **DS** )  
Bedingt durch die beschränkte Stellenzahl sind ev. niederwertige Dekaden in der Displayzeile nicht sichtbar. Die Position muss in diesem Fall über die Schnittstelle abgefragt werden ( **U  $m_e$**  )
- Beim Einsatz von Absolutgebern muss gewährleistet sein, dass der Encodernullpunkt ( Sprung von Minimal- zu Maximalwerten u. u. ) nicht im nutzbaren Fahrbereich liegt. Das Überfahren des Encodernullpunkts würde zu Fehlfunktionen und Anzeigefehlern führen.  
Zum Verschieben des Nullpunkts muss die Encoderwelle nach Lösen der Kupplung entsprechend gedreht werden.



Zur Kontrolle den Antrieb über den ganzen Fahrbereich laufen lassen und den Encodermesswert auf dem Display beobachten. Es dürfen dabei keine Wertesprünge auftreten.

- Den Encoder-Faktor zur Umrechnung von Längen- bzw. Winkel-Einheiten in Encoderschritte ermitteln und eingeben für Achse  $m_e$  ( Parameter **FD** ).  
Berechnungsbeispiele im **Anhang Getriebefaktoren**
- Positionswerte der Antriebsachse und der Encoderachse null setzen ( Parameter **U** )
- Zur Kontrolle verschiedene Ziele anfahren und die beiden für Antrieb und Encoder angezeigten Werte vergleichen.  
Wenn die Vorzeichen der beiden angezeigten Werte nicht übereinstimmen, muss das Vorzeichen des Encoderfaktors geändert werden ( Parameter **FD  $m_e$**  )
- Nach positiv verlaufenem Test die des Encoderzuordnung für Achse  $m_e$  löschen mit **EC  $m_e$  0 0** und die für den Encoder ermittelten Parameterwerte ( Parameter **EC , EP , FD** ) neu eingeben für die Achsnummer der Antriebsachse (  $m$  )
- An eine bekannte ( ausgemessene ) Position fahren und den zugehörigen Positionswert eingeben ( Parameter **U** )
- Softwarelimiten eingeben ( Parameter **H** )
- Zur Prüfung verschiedene Positionen anfahren.  
Wenn Positionierungsfehler gemeldet werden, muss ev. der Toleranzparameterwert ( **M  $m$**  ) höher gesetzt werden.

## Setup Inkrementalsystem

Für Achse mit Inkrementalgeber oder ohne Geber

- Referenzlaufmodus setzen ( Parameter **K** )  
Details im **Anhang Referenzverfahren**
- Gebernullpunkt null setzen ( **V m 0** )
- Referenzfahren ( **R m** )
- Nach erfolgreichem Referenzlauf steht der Antrieb auf dem Referenzpunkt  
angezeigter Positionswert ist 0.000
- An bekannte (ausgemessene) Position fahren und Positionswert eingeben (Parameter **U** )
- Softwarelimiten eingeben ( Parameter **H** )

## Parametereinstellungen

Solange keine neuen Parameterwerte eingegeben werden, sind die Defaultwerte gesetzt

Nachfolgend eine Liste der wichtigsten Parameter

- 1) **EC** Encoder zuordnen  
Definition des Encodertyps und der Encodernummer einer Motorachse
- 2) **FM** Motor-Getriebefaktor  
Skalierungsfaktor aus Motor-Schrittauflösung und Getriebe-Übersetzung
- 3) **FD** Encoder-Getriebefaktor  
Skalierungsfaktor aus Encoder-Auflösung und Messgetriebe- Übersetzung
- 4) **EP** Programmiercode für SSI-Encoder, ( default 24Bit, Graycode )  
Definition der Auflösung und der Codeart für Encoder mit SSI-Schnittstelle
- 5) **U** momentane Position des Antriebs  
die absolute Position des Encodernullpunkt ( Param. V ) wird daraus  
automatisch errechnet
- 6) **H** Software-Limiten  
Definition des Fahrbereichs
- 7) **W** Magnetbremsenfunktion oder Luftkissenverkettung  
Definition der Achse, an welcher der Output AC aktiviert werden soll  
( z.B. für Magnetbremsen die eigene Achse **W m m** )
- 8) **J** Positionierfrequenz  
Motor-Schrittfrequenz beim Positioniervorgang
- 9) **G** Start/Stop-Frequenz  
Schrittfrequenz beim Freifahren von Schaltern

- 10) **E** Beschleunigung  
Beschleunigungs- und Verzögerungs-Rate der Schrittfrequenz beim Positionieren
- 11) **OPT** Optionen-Register  
Freifahr-Modus, Direktfahr-Modus, Luftkissen-Rückmeldung
- 12) **K** Referenzlauf-Modus für Betrieb mit Inkremental-Positionsmessung  
( mit Inkremental-Geber od. ohne Geber )
- 13) **M** Positioniertoleranz
- 14) **A** Anzahl Nachkommastellen  
Auf dem Display angezeigte Nachkommastellen
- 15) **MN** Motorname 1 Wort, max. 15 Zeichen  
Bezeichnung des Antriebs
- 16) **MEM** Kurztext 1 Wort, max. 15 Zeichen
- 17) **R** Referenzlauf  
Anfahren der Referenzposition und automatische Bestimmung des Parameters V  
(Skalen-Nullpunkt) bei inkrementaler Positionsmessung oder geberlosem Betrieb
- 18) **Q** Referenzparameter  
Parameter zur Reduktion der Fahrzeit beim Referenzlauf  
( Vorsicht bei zu grossem Wert wird die Referenzposition überfahren.  
Der Referenzpunkt kann nicht gefunden werden )

**Die eingestellten Parameter anschliessend auslesen und archivieren !**

# Betrieb

## Systemstart

Beim Aufstarten des Schrittmotor-Controllers wird auf beiden seriellen Schnittstellen und dem 1. Displaymodul ( Display für die Achsen 1 – 4 ) eine Einschaltmeldung ausgegeben. Danach ist der Controller ist bereit für Lokalbetrieb mit den Tastenbedienung, was durch die leuchtenden gelben Bewegungstasten an den Displaymodulen angezeigt wird. Beide Schnittstellen sind Offline geschaltet

**Power On** oder **Warmstart** mit dem CPU-Schalter ( nach unten )  
die zuletzt gültigen Parameter- und Positions-Werte werden aus dem im Betrieb laufend nachgeführten CMOS-RAM geladen

**Parameter-Reset** am 1. Displaymodul ( Achsen 1 – 4 ) die quadratische grüne Sytemtaste und die unterste grüne runde Taste ( Zeile 4 ) gedrückt halten  
CPU-Schalter nach oben  
die Default-Parameterwerte werden geladen

## Steuerung über serielle Schnittstelle

Die beiden Schnittstellen ( Rückseite **Host 1**, Frontseite **Service 2** ) sind in ihrer Funktion gleichwertig. Es gibt keine Priorität.

Durch Online-Schalten einer Schnittstelle mit **RMT 1** oder **RMT 2** wird die andere Offline geschaltet. Es ist keine gegenseitige Verriegelung möglich. Eine Schnittstelle kann jederzeit die Kontrolle übernehmen oder übergeben ( keine Priorität )

Onlineschalten **RMT 1** bzw. **RMT 2**

Offlineschalten **RMT 0**

Nach dem Systemstart sind beide Schnittstellen in den für manuellen Betrieb mit Terminalemulator geeigneten Terminal- oder Echo-Modus geschaltet  
Eingabezeichen werden zurückgesendet, Textmeldungen und Statusmeldungen werden ausgegeben.

Für den automatischen Betrieb mit Hostcomputer muss zur Unterdrückung von Echo, Text und Status-Meldungen in den Computermode gewechselt werden

### ECHO 0

Die Einstellung wirkt nur an der momentan aktiven Schnittstelle

Da im Computermode keine Statusmeldungen ausgegeben werden ( z. B. nach beendetem Positionier-Lauf ) müssen alle Zustände im Pollingverfahren abgefragt werden.

## Lokale Tastenbedienung

Im Offline-Zustand der Steuerung ( erkennbar ist an den leuchtenden gelben Richtungstasten ) kann eine ausgewählte Antriebsachse über die Tasten der Display-Module bewegt werden

Durch Drücken der grünen Systemtaste des ersten Displaymoduls ( Achsen 1 .. 4 ) während 3 sec. wird die Steuerung manuell in den Offline-Zustand geschaltet.

Wie im Normalbetrieb sind die aktuellen Betriebsparameter ( Stepfrequenzen, Beschleunigung, Position etc.) wirksam.

Eine Motorachse wird durch Betätigen einer Achswahltaste ausgewählt. Die Tasten-LED zeigt die Selektion an, die Richtungstasten-LEDs der nicht aktiven Displaymodule werden ausgeschaltet.

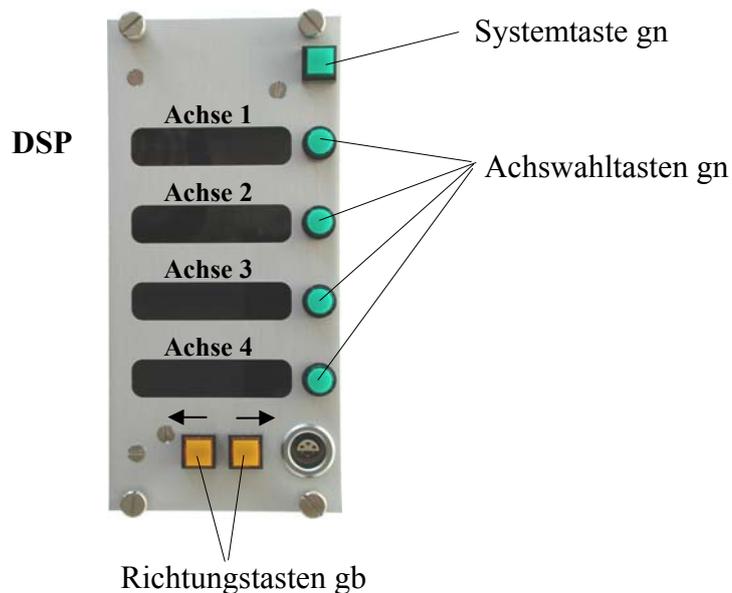
Folgende Bewegungsarten sind möglich:

- Einzelschritt vor- oder rückwärts durch Antippen der entsprechenden Richtungstaste.
- Vor- oder Rückwärtslauf mit der programmierten Start-Stop-Frequenz durch Drücken und Halten der entsprechenden Richtungstaste. Der Motor dreht, solange die Taste gedrückt bleibt.
- Vor- oder Rückwärtslauf mit der programmierten Maximal-Frequenz und Beschleunigung durch gleichzeitiges Drücken und Halten einer Achswahltaste und der entsprechenden Richtungstaste. Lösen der Richtungstaste leitet die Bremsrampe ein.

Durch gleichzeitigen Druck auf zwei benachbarte Achswahltasten ( 1 + 2, 2 + 3 oder 3 + 4 ) wird die Achs-Selektion zurückgesetzt.

Die Achswahltaste erlischt.

Die Richtungstasten aller Displaymodule leuchten.



### **Achtung !**

Im OpenLoop-Betrieb ( ohne Encoder ) ändert die Positions-Anzeige nicht, solange der Motor in Bewegung ist.

## **Installation**

Die Motorsteuerungs-Einheiten sind mit ihrer offenen Bauweise, welche eine optimale Belüftung der Geräte erlaubt, zum Einbau in 19“-Schränken konzipiert.

Zur Wärmeabfuhr ist für genügende Luftzirkulation zu sorgen.

Für den Einsatz als Tischgerät müssen als Berührungsschutz Deck- und Bodenbleche, und zur Gewährleistung der freien Luftzirkulation Gerätefüsse montiert werden.

Die Lüftungsschlitze in den Blechen müssen offen bleiben.

Keine Gegenstände auf den Geräten deponieren!

Jede Einheit muss über den Erdungs-Gewindebolzen in der Geräterückwand mit der Struktur des Schaltschranks elektrisch verbunden werden.

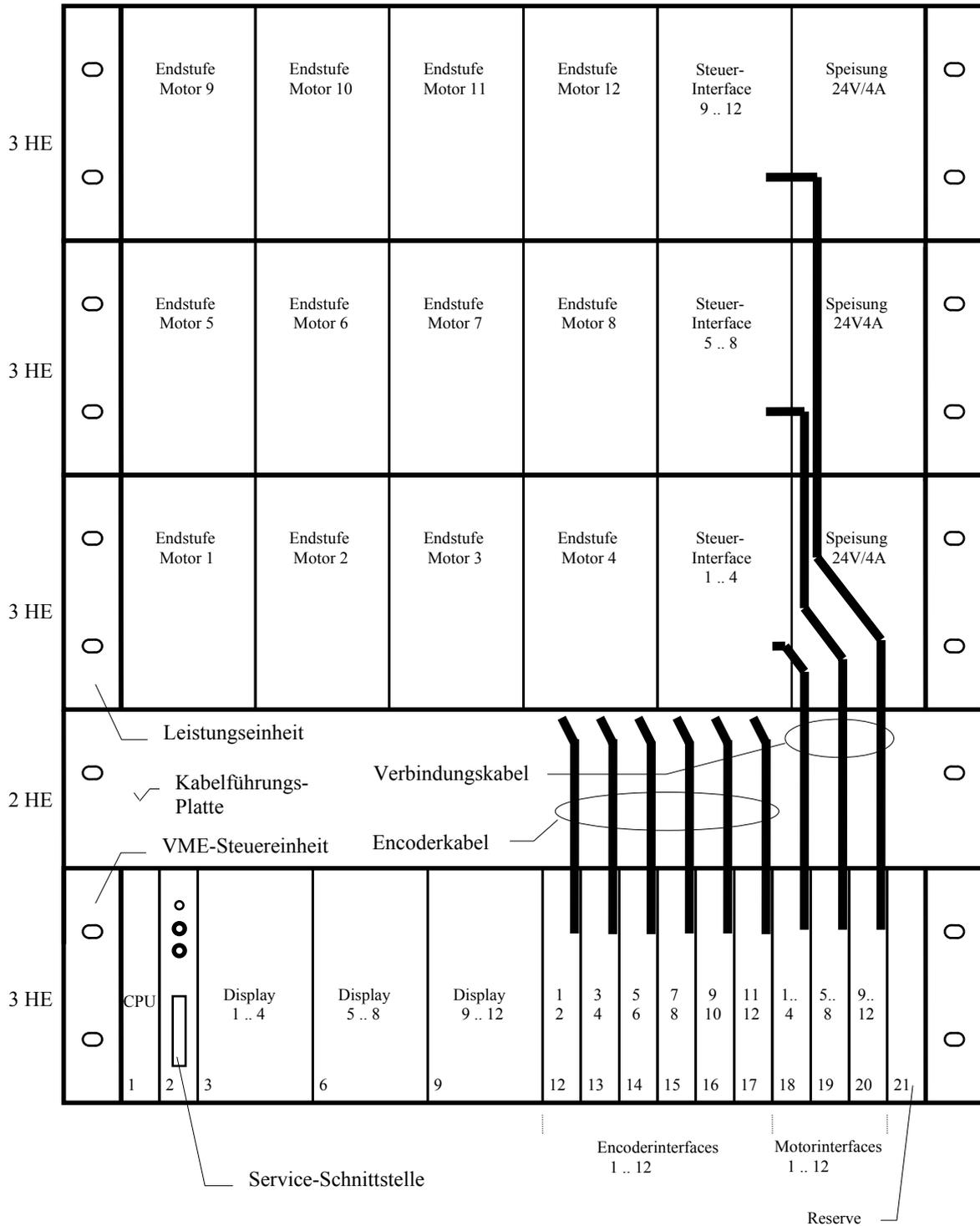
Alle Motor-, Encoder- und Signalkabel müssen abgeschirmt sein. Die Schirme sind beidseitig mit Masse zu verbinden ( direkt oder über die Gehäuse der Kabelstecker ) .

Die Lage der Endschalter und ihr Anschluss an die Klemmen LS1 und LS2 definieren den Anfang und das Ende des Fahrbereichs. Ein Austausch durch Steckbrücken oder Software-Konfiguration ist nicht möglich.

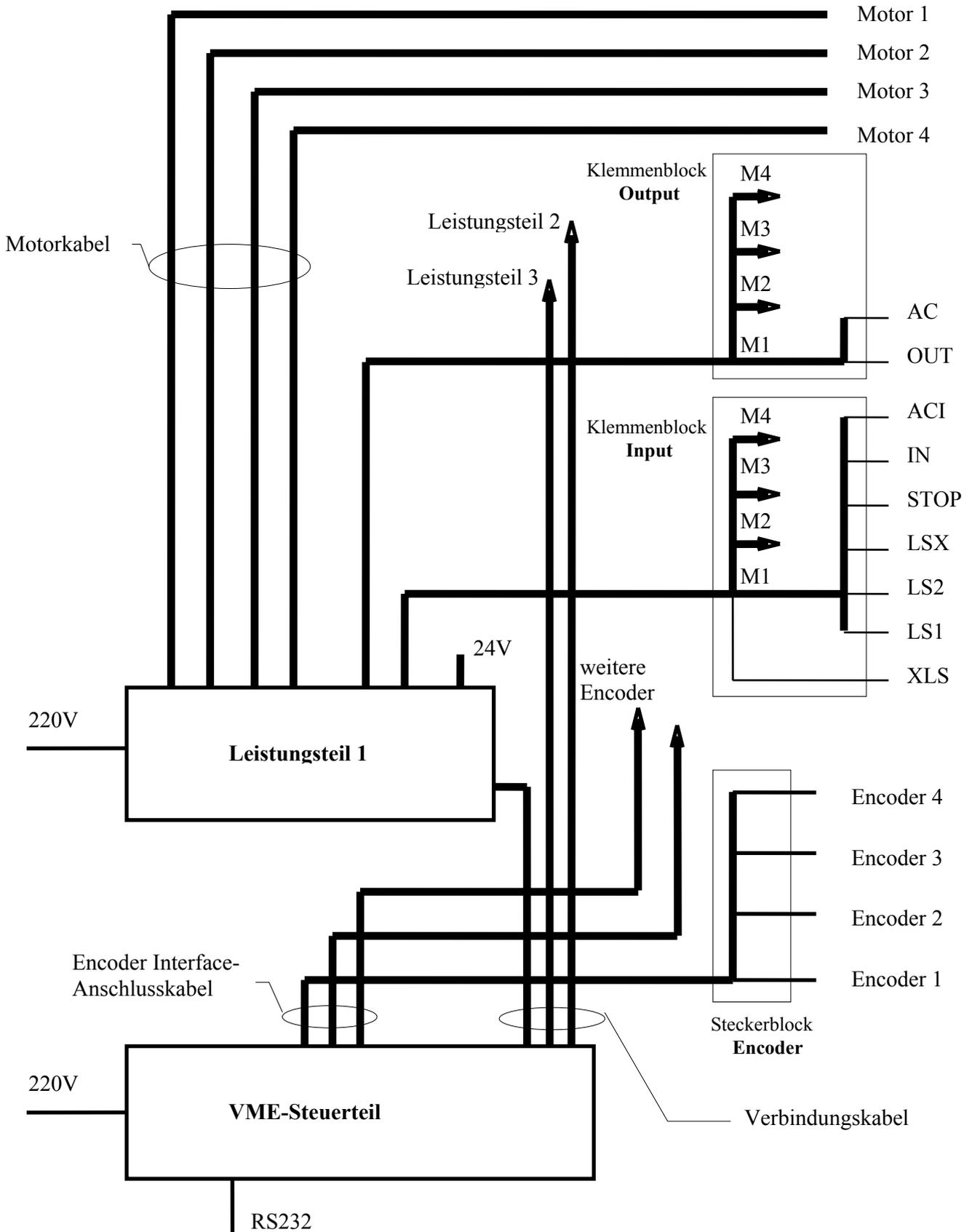
Nicht benutzte Endschalter-, Stop- Notstop- und Ref-Eingänge können mit Jumperbuchsen oder Drahtbrücken auf den Anschluss-Klemmenblöcken gebrückt werden, um ihre Status-LEDs auf den IO-Interfaces dunkel zu schalten.

# Geräte-Einbau

## Standardaufbau für 12 Achsen



# Blockschema

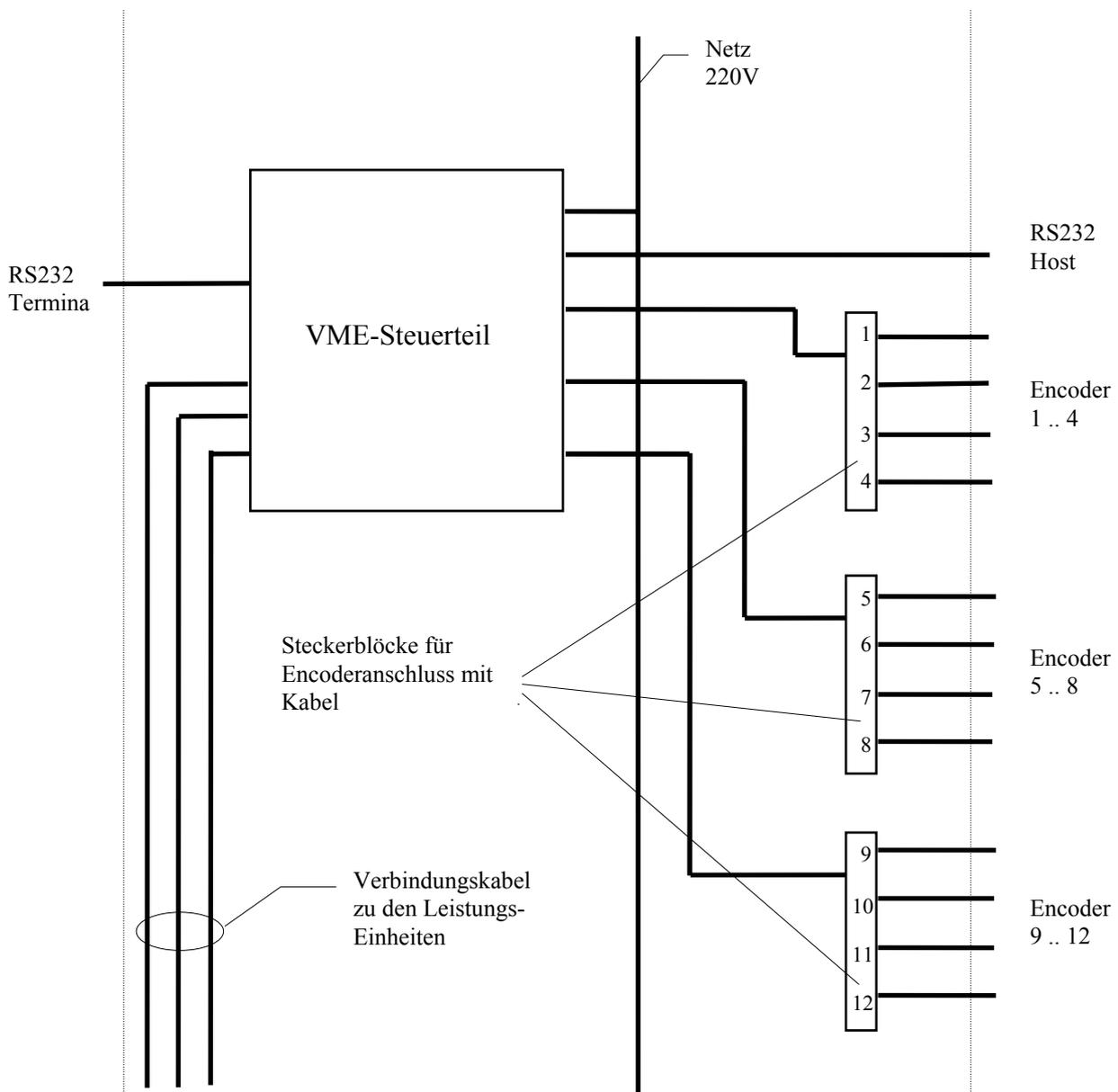


## Anschluss VME-Steuerteil

Bei der Verkabelung ist darauf zu achten, dass die Bedienungs- und Anzeige-Elemente auf den Gerätefronten von Kabeln frei gehalten werden.

Die Leistungseinheiten werden durch gradlinig geführte, längenangepasste Flachrundkabel mit den MOT-Modulen des Steuerteils verbunden.

Der Anschluss der Encoder erfolgt über Interface-Adapterkabel mit im Schrank montierten Steckerblöcken. ( Ausnahme: beim **SSI4**-Interface werden die Encoderkabel direkt auf der Interface-Frontplatte gesteckt )



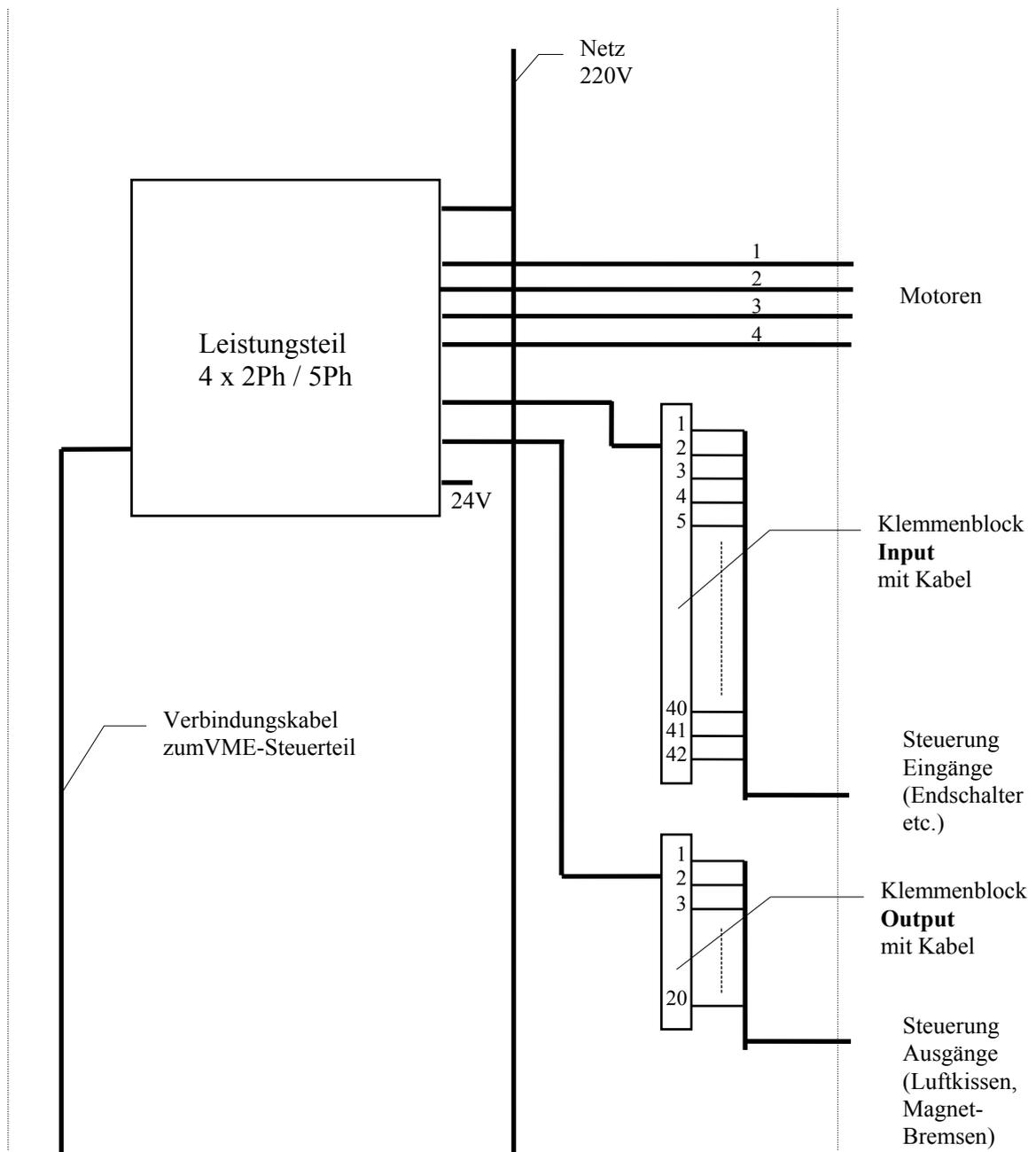
## Anschluss Leistungsteil

Die Motorkabel werden direkt am Leistungsteil gesteckt.

Die Steuersignale (Endschalter, Referenzpunkt-Schalter etc., sowie die Steuer-Ausgänge werden über im Rack montierte Anschluss-Klemmenblöcke geführt, was einen individuellen Anschluss der Anlagenverdrahtung ermöglicht.

Bei der 8-fach 3Ph Leistungseinheit stecken Klemmenblöcke direkt in der Geräterückwand

Die VME-Stuereinheit wird mit den Leistungseinheiten durch angepasste Verbindungskabel verbunden. ( verschiedene Längen )



# Steckerbelegung

## VME-Steuerteil

### RS232-Schnittstellen

- Host-Anschluss Rückwand
- Service-Anschluss Frontseite

Stecker D-Sub 25P m, DCE

Pin	Signal
1	NC
2	TxD
3	RxD
4	RTS *
5	CTS *
6	NC
7	GND
8 .. 25	NC

\* nicht benutzt

### Testbuchsen

2mm Buchsen fixe Frontplatte Modulsteckplatz 2

Buchse	Signal
rot	+5V
schwarz	GND

### Display

Steckdose LEMO 1 Frontplatte Displaymodul

Pin	Signal
1	Analog-Output
2	Analog Input
3	Bit 0
4	Bit 1
5	Bit 2
6	GND

## Leistungsteil

### Motorstecker

Steckdose Burndy Metalok Bantam 12P w in allen **4fach Leistungseinheiten** für alle Motoren

	<b>2-Phasen-Motor</b>	<b>5-Phasen-Motor</b>	<b>3-Phasen-Motor</b>
<b>Pin</b>	<b>Signal</b>	<b>Signal</b>	<b>Signal</b>
1	A	W 1A	V
2	B	W 1E	U
3	C	W 2A	
4	D	W 2E	U
5		W 3A	
6		W 3E	
7		W 4A	W
8		W 4E	
9		W 5A	W
10		W 5E	V
11			
12	Masse	Masse	Masse

### 8-fach Leistungseinheit für 3Ph-Motoren ( MARS )

Steckdose LEMO3 4P

	<b>3-Phasen-Motor</b>
<b>Pin</b>	<b>Signal</b>
1	U
2	V
3	W
4	Masse

### Stecker 24V

Steckdose Burndy Metalok Bantam 4P w

<b>Pin</b>	<b>Signal</b>
1	+24V
2	+24V
3	0V
4	0V

## Stecker Steuerung Input

Am Inputstecker wird ein im Rack montierter Klemmenblock mit Verbindungskabel angeschlossen, welcher die individuelle Verteilung der Eingangssignale ermöglicht.

Flachbandstecker 64P m

Pin	Signal	Funktion	Pin	Signal
1	GND		2	GND
3	GND		4	GND
5	LS1-1	- Endschalter 1	6	+24V
7	LS2-1	+Endschalter 1	8	+24V
9	LSX-1	Referenzschalter 1	10	+24V
11	STOP-1	Stopschalter 1	12	+24V
13	ACI-1	Luftkissenrückmeldung 1	14	+24V
15	IN-1	Steuereingang 1	16	+24V
17	GND		18	GND
19	LS1-2	- Endschalter 2	20	+24V
21	LS2-2	+Endschalter 2	22	+24V
23	LSX-2	Referenzschalter 2	24	+24V
25	STOP-2	Stopschalter 2	26	+24V
27	ACI-2	Luftkissenrückmeldung 2	28	+24V
29	IN-2	Steuereingang 2	30	+24V
31	GND		32	GND
33	LS1-3	- Endschalter 3	34	+24V
35	LS2-3	+Endschalter 3	36	+24V
37	LSX-3	Referenzschalter 3	38	+24V
39	STOP-3	Stopschalter 3	40	+24V
41	ACI-3	Luftkissenrückmeldung 3	42	+24V
43	IN-3	Steuereingang 3	44	+24V
45	GND		46	GND
47	LS1-4	- Endschalter 4	48	+24V
49	LS2-4	+Endschalter 4	50	+24V
51	LSX-4	Referenzschalter 4	52	+24V
53	STOP-4	Stopschalter 4	54	+24V
55	ACI-4	Luftkissenrückmeldung 4	56	+24V
57	IN-4	Steuereingang 4	58	+24V
59	GND		60	GND
61	XLS	Notstop	62	+24V
63	GND		64	GND

## Stecker Steuerung Output

Am Outputstecker wird ein im Rack montierter Klemmenblock mit Verbindungskabel angeschlossen, welcher die individuelle Verteilung der Ausgangssignale ermöglicht

Flackbandstecker 50P m

Pin	Signal	Funktion	Pin	Signal
1	OUT-1	Steuer-Ausgang 1	2	GND
3			4	GND
5			6	GND
7	ACO-1	Luftkissen-Ausgang 1	8	GND
9			10	GND
11			12	GND
13	OUT-2	Steuer-Ausgang 2	14	GND
15			16	GND
17			18	GND
19	ACO-2	Luftkissen-Ausgang 2	20	GND
21			22	GND
23			24	GND
25	OUT-3	Steuer-Ausgang 3	26	GND
27			28	GND
29			30	GND
31	ACO-3	Luftkissen-Ausgang 3	32	GND
33			34	GND
35			36	GND
37	OUT-4	Steuer-Ausgang 4	38	GND
39			40	GND
41			42	GND
43	ACO-4	Luftkissen-Ausgang 4	44	GND
45			46	GND
47			48	GND
49	+24V		50	+24V

## Klemmenblock Steuerung Input

Die Steuerleitungen werden über die im Rack montierten Klemmenblöcke angeschlossen.

Pin	Signal	Funktion	Pin	Signal	Funktion
1	+24V	Speisung	23	+24V	Speisung
2	LS1-1	- Endschalter 1	24	LS2-1	+Endschalter 1
3	LSX-1	Referenzschalter 1	25	GND	
4	STOP-1	Stopschalter 1	26	GND	
5	ACI-1	Luftkissenrückmeldung 1	27	IN-1	Steuereingang 1
6	+24V	Speisung	28	+24V	Speisung
7	LS1-2	- Endschalter 2	29	LS2-2	+Endschalter 2
8	LSX-2	Referenzschalter 2	30	GND	
9	STOP-2	Stopschalter 2	31	GND	
10	ACI-2	Luftkissenrückmeldung 2	32	IN-2	Steuereingang 2
11	+24V	Speisung	33	+24V	Speisung
12	LS1-3	- Endschalter 3	34	LS2-3	+Endschalter 3
13	LSX-3	Referenzschalter 3	35	GND	
14	STOP-3	Stopschalter 3	36	GND	
15	ACI-3	Luftkissenrückmeldung 3	37	IN-3	Steuereingang 3
16	+24V	Speisung	38	+24V	Speisung
17	LS1-4	- Endschalter 4	39	LS2-4	+Endschalter 4
18	LSX-4	Referenzschalter 4	40	GND	
19	STOP-4	Stopschalter 4	41	GND	
20	ACI-4	Luftkissenrückmeldung 4	42	IN-4	Steuereingang 4
21	+24V	Speisung	43	+24V	Speisung
22	XLS	Notstop	44	GND	

## Klemmenblock Steuerung Output

Pin	Signal		Pin	Signal
1	OUT-1	Steuer-Ausgang 1	11	GND
2	ACO-1	Luftkissen-Ausgang 1	12	GND
3	OUT-2	Steuer-Ausgang 2	13	GND
4	ACO-2	Luftkissen-Ausgang 2	14	GND
5	OUT-3	Steuer-Ausgang 3	15	GND
6	ACO-3	Luftkissen-Ausgang 3	16	GND
7	OUT-4	Steuer-Ausgang 4	17	GND
8	ACO-4	Luftkissen-Ausgang 4	18	GND
9	+24V	Speisung	19	+24V
10	+24V	Speisung	20	+24V

## Encoder-Anschlüsse

### Verteilerblock Inkrementalgeber

Stecker D-Sub 9P m Crimp- oder Löt-Montage

Pin	Farbe	Signal	Pin	Farbe	Signal
1		UA2+	6		UA0+
2		UA2-	7		/
3		UA1-	8		VCC
4		UA1+	9		GND
5		UA0-			
Gehäuse	Schirm				

### Verteilerblock Absolutgeber m. SSI-Interface

Encoder **ROC424.2** GS5-12-12 mit 24 Bit ohne Parität, 4096/360°, 4096 Umdr. VCC 10.5V .. 26V

### Anschluss am DMS SSI-Interface

Stecker D-Sub 15P m Crimp- oder Löt-Montage

Pin	Farbe	Signal	Pin	Farbe	Signal
1			9		
2	rosa	-Daten	10	grau	+Daten
3	gelb	-Takt	11	violett	+Takt
4			12		
5	blau	VCC*	13	braun/grün	VCC
6	weiss	GND*	14	weiss/grün	GND
7			15		
8					
Gehäuse	Schirm				

\* Fühlleitung

Typ **ROC425** mit 25 Bit ( Nachfolge-Typ von ROC424 )

Pin 3 und Pin 11 werden wie folgt vertauscht angeschlossen:

3	violett	+Takt	11	gelb	-Takt
---	---------	-------	----	------	-------

Alle andern Anschlüsse sind identisch mit ROC424.

### Anschluss am SSI4-Interface

Stecker LEMO 2 8p Crimpmontage

alle Encoder-Typen werden normal  
angeschlossen  
( Takt-Anschlüsse für ROQ425 nicht  
vertauscht )

Pin	Farbe	Signal
1	rosa	-Daten
2	grau	+Daten
3	gelb	-Takt
4	violett	+Takt
5	braun/grün	VCC
6	blau	VCC sense
7	weiss/grün	GND
8	weiss	GND sense
Gehäuse	Schirm	

### Verteilerblock Absolutgeber Litton

Typ VNB11-13P1 mit 13 Bit, 256/360°, 32 Umdr.

Typ VNB11-17P1 mit 17 Bit, 256/360°, 512 Umdr.

Stecker D-Sub 37P m Crimpmontage

DELCONEX Typ DT 37PX Distrelec Art. Nr. 12 47 44

Pin	Farbe	Bit	Pin	Farbe	Bit
1	braun	LD1	20	weiss/blau-schwarz	0
2	rot	LD2	21	weiss/braun	LG1
3	orange	LD3	22	weiss/rot	LG2
4	gelb	LD4	23	weiss/orange	LG3
5	grün	LD5	24	weiss/gelb	LG4
6	blau	LD6	25	weiss/grün	LG5
7	violett	LD7	26	weiss/blau	LG6
8	grau	LD8	27	weiss/violett	LG7
9	weiss	LD9	28	weiss/grau	LG8
10	orange/schwarz	LD10	29	weiss/schwarz	LG9
11	gelb/braun	LD11	30	weiss/orange-schwarz	LG10
12	braun/rot	LD12	31	weiss/gelb-braun	LG11
13	weiss/rot-orange	LD13	32	weiss/braun-rot	LG12
14	weiss/rot-gelb	LD14	33	rot/orange	LG13
15	weiss/rot-grün	LD15	34	rot/gelb	LG14
16	weiss/rot-blau	LD16	35	rot/grün	LG15
17	NC		36	rot/blau	LG16
18		C	37		C
19	schwarz	C			

Der Typ VNB11-13P1 (13 Bit) ist nur an  angeschlossen. Die übrigen Pins bleiben frei.

Der Kollektor ist in jedem Fall an Pin 37 angeschlossen. 17 bis 19 bleiben frei.

## VME-Steuereinheit

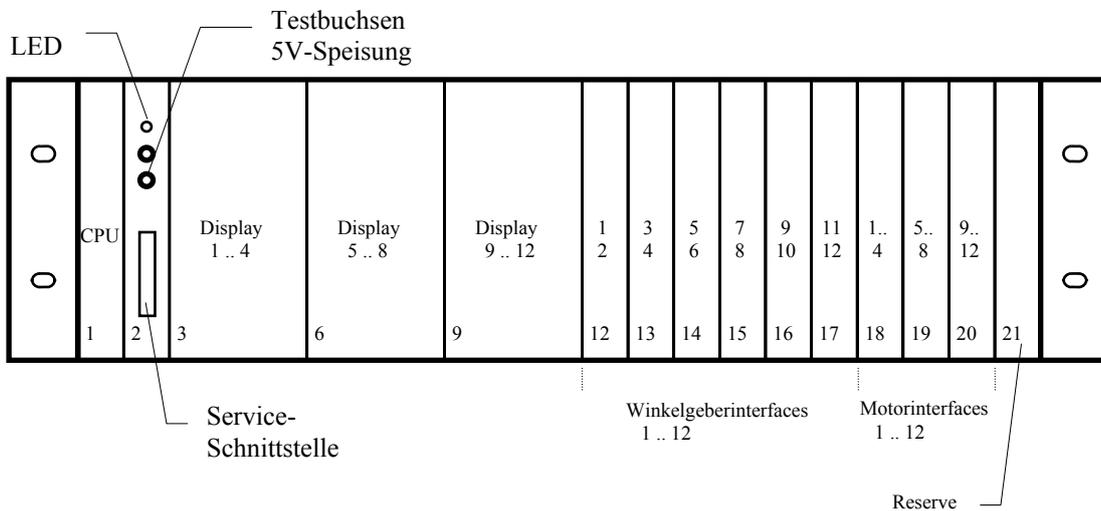
Die VME-Steuereinheit trägt die VME-Busplatine mit 21 frontseitig zugänglichen Modulsteckplätzen und zur Stromversorgung ein 5V-Schaltnetzteil

Slot 2 ist durch eine fest montierte Frontplatte belegt

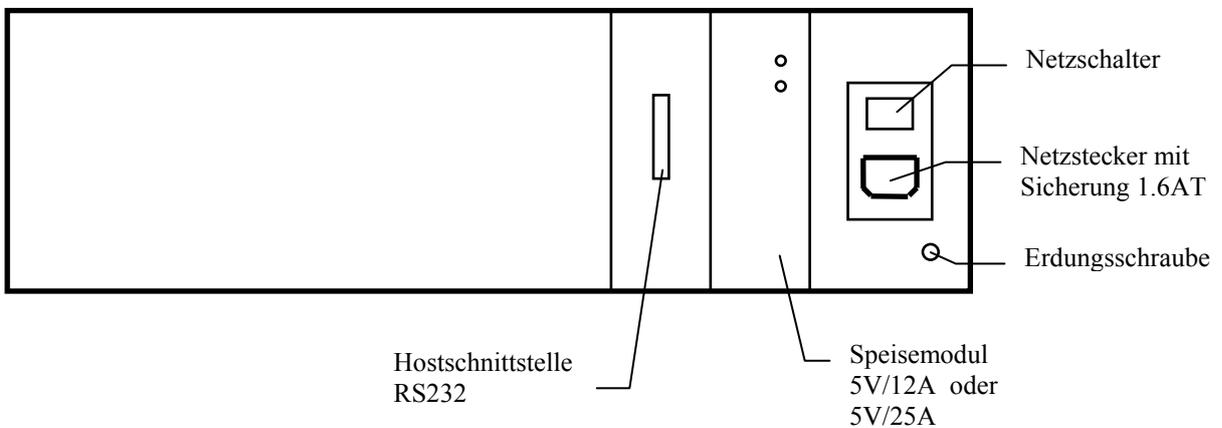
Um die Displaymodule stecken zu können, dürfen in den Slots 3 und 4, 7 und 8, 10 und 11 oben und unten keine Kartenführungen eingesetzt sein

Die Netzanschlussdose mit Schalter und Sicherung und ein DSub 25p-Stecker für die Verbindung mit dem Host-Computer über RS232 sind in die Rückwand eingebaut

### Ansicht Frontseite



### Ansicht Rückseite



## VME-Module

Mit Ausnahme des Displaymoduls belegt jedes Modul einen einzelnen Slot

Für die Displaymodule sind je 3 Slots reserviert

Im 2. und im 3. durch ein Displaymodul belegten Slot dürfen keine Kartenführungen eingesetzt sein

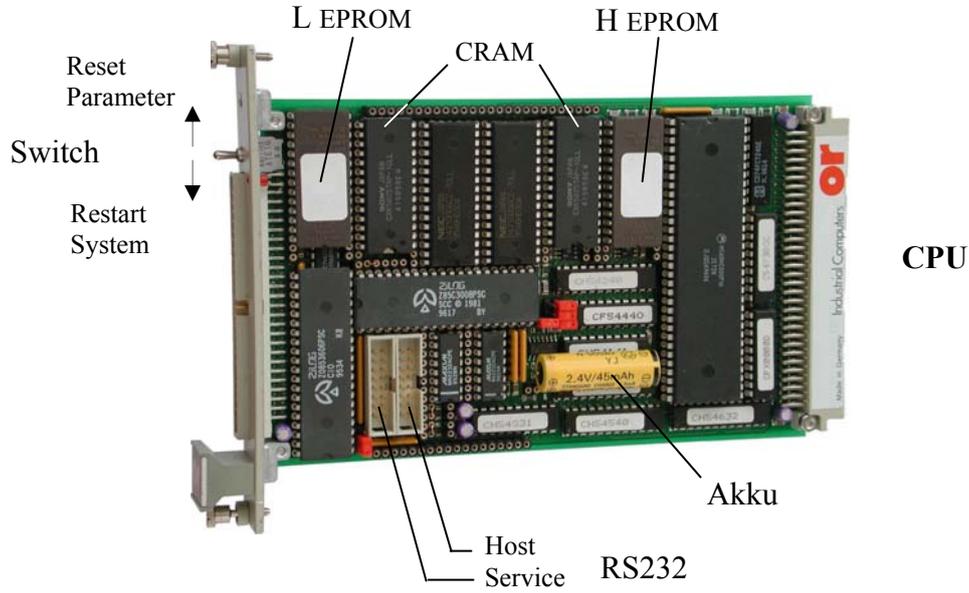
Die Steckplatz-Numerierung im VME-Bus verläuft von links nach rechts

Den verschiedenen Modultypen sind in der VME-Einheit Steckplatzbereiche zugeordnet

Andere Anordnungen sind möglich ( Ausnahmen CPU, Display )

1	CPU-Modul
2	Fest montierte Platte mit dem Service-Schnittstellenstecker, sowie Power-LED und zwei 2mm-Prüfbuchsen zur Kontrolle der 5V Bus-Speisung
3, 6, 9	Display-Module
12 .. 17	Encoder-Interfaces, je nach den verwendeten Encodern.
18 .. 20	Motor-Interfaces
21	Reserveplatz

**CPU** VCPU04-C16 VMEBus-CPU mit 68HC000, 16MHz,  
256kB RAM, 2 x RS232C, Z8536  
ausgebaut mit 64kB CMOS-RAM mit Puffer-Akku

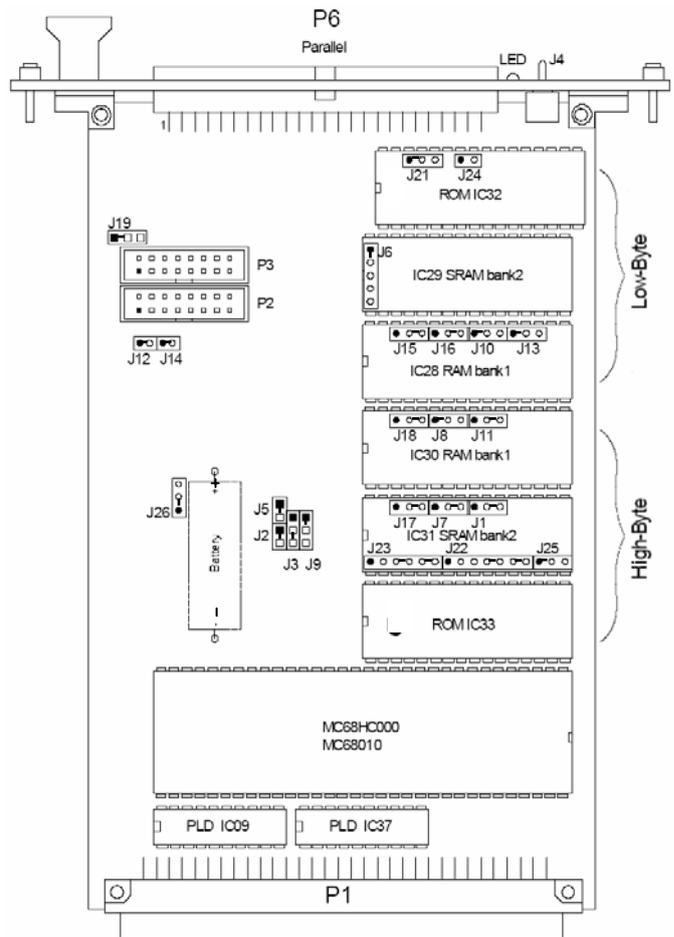


Grösse 3HE 4TE  
Steckplatz Pos. 1 fix

EPROM	2 x 27C512	Programmspeicher
CRAM	2 x 64Kx8	Parameterspeicher
Akku	Sanyo N-50SB2	2.4V/50mA

### Jumper-Configuration CPU

Jumper	Position	Funktion
J1	2-3	CMOS RAM Battery powered
J2	1-2	SYSCCLK enabled
J3	2-3	Slot 1 Funktion enabled
J5	1-2	VMEbus Timeout enabled
J6	4-5   1-2	CPU-Typ VCPU0xxC12 xxC16
J7	2-3	Battery Power Source onboard
J8	1-2	no Wait States
J9	1-2	Dual Port RAM Int. disabled
J10	1-2	EPROM Type 27C512 IC32, IC33
J15	2-3	
J16	2-3	
J21	1-2	
J24	open	external Interrupts enabled
J11	2-3	
J12	1-2	RS232 Selection
J14	1-2	RAM2 Size 32Kx8 IC29, IC31 ( CRAM )
J13	1-2	
J17	2-3	
J22	4-5 6-7	RAM1 Size 128Kx8 IC28, IC30
J23	3-4 5-6	
J25	1-2	no Write protection RAM2
J18	2-3	
J26	1-2	
J19	1-2	



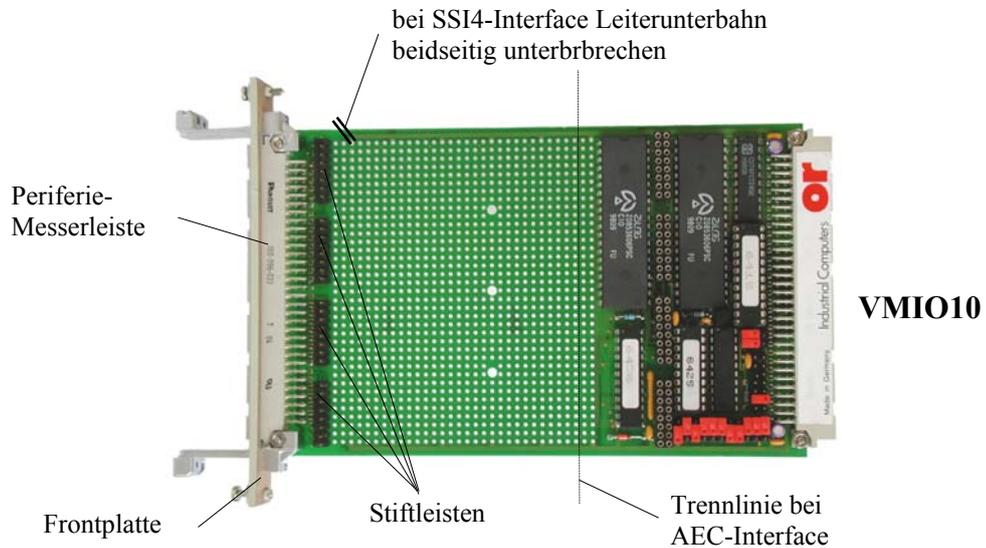
### Epromtausch

- Halteschrauben des CPU-Moduls lösen
- CPU-Modul zur Hälfte herausziehen
- Schnittstellen-Flachbandstecker ausziehen
- CPU-Modul aus der VME-Einheit vorsichtig entfernen  
auf Flachbandkabel achten, nicht einklemmen zw. Akku und Nachbarmodul !!
- beide EPROMs entfernen
- neue EPROMs einsetzen
- Steckposition kontrollieren ( L an Frontseite, Pin 1 gegen Kartenmitte, am Kartenrand bleiben keine Buchsen frei, alle Pins stecken in den Sockelbuchsen )

### Einsetzen des CPU-Moduls in die VME-Einheit

- Schnittstellen-Flachbandkabel etwas aus der VME-Einheit herausziehen
- CPU-Modul zur Hälfte in die Printführungen einführen
- Schnittstellen-Flachbandverbinder stecken ( Steckposition kontrollieren )
- CPU-Modul vorsichtig ganz einschieben ( Flachbandkabel nicht verklemmen! )
- Frontplatten-Halteschrauben festziehen

**VMIO** VMIO10-C universelle VMEBus IO-Interfacekarte mit 2 x Z8536  
wird als Basis für alle VMEBus-Interfaces ausser SSI eingesetzt



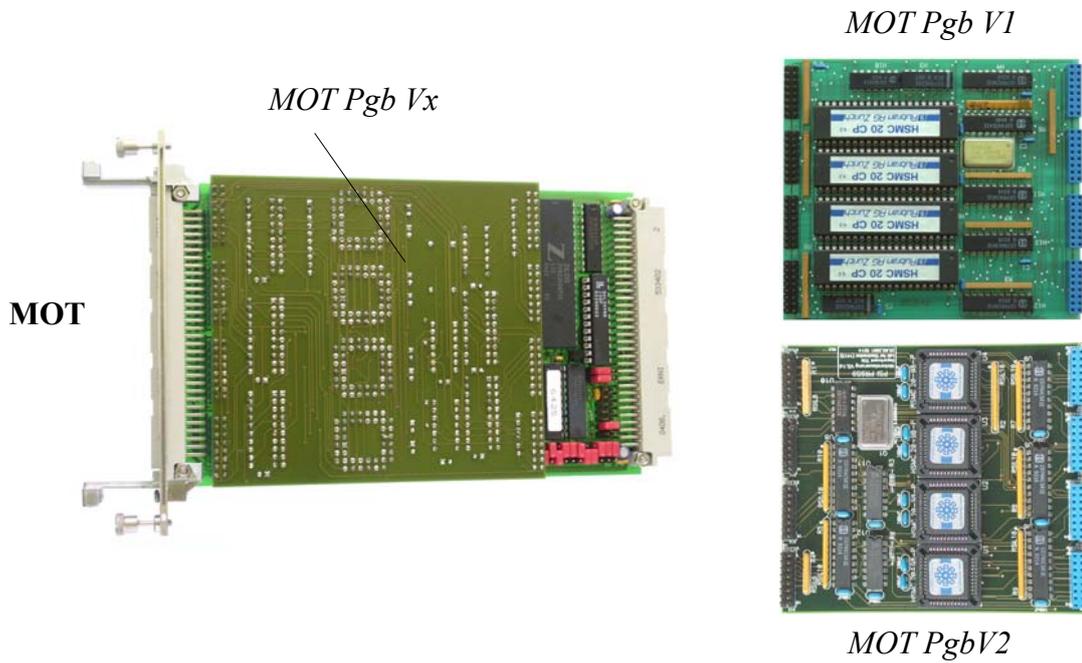
Die zur Sicherung der Piggybacks vorgesehenen Distanzbolzen werden bei allen VMIO10-Karten entfernt.

Anwendungsspezifische Modifikationen

Modul	Modifikation
MOT INC	Originalzustand
DSP	ohne Periferie-Messerleiste und Frontplatte
AEC	Karte gekürzt
SSI4	ohne Periferie-Messerleiste, Stiftleisten und Frontplatte zus. 36p-Stiftleiste, Spezialfrontplatte m. Anschlüssen GND und 5V zur Periferie unterbrechen

Doku ext. [Technische Dokumentation VMIO10](#)

**MOT** Stepmotor-Interface



Original VMIO10-C mit Piggyback *MOT Pgb V1* oder *MOT Pgb V2*

- Kanalzahl 4
- Steckplatz variabel
- Anschluss Messerleiste DIN41612 Bauform C 3 x 32p
- Konfiguration Moduladresse

Interface Nr.	Moduladresse	Achse
MOT 1	1800	1..4
MOT 2	1900	5..8
MOT 3	1A00	9..12

- Funktion Controller für 1 .. 4 Stepmotoren  
Beschleunigungsrampe
- Signale Endschalter, Stop, Ref. Punkt, Schnellstop  
pro Achse je ein frei verwendbarer Input und Output

Es existieren 2 Varianten der MOT Pgb-Karte, welche sich durch ihre Controllerbausteine unterscheiden.

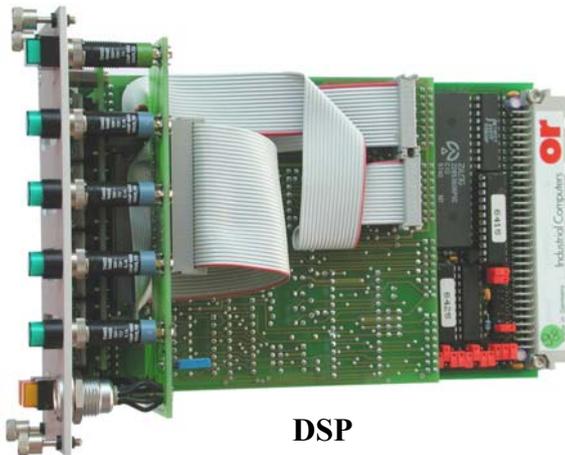
- V1 Typ HSMC20CP mit Sockel DIP40
  - V2 Typ HSMC-20-98-P mit Sockel PLC44 neue Version
- Die beiden Versionen sind gegeneinander austauschbar

**Doku** ext. Technische Dokumentation [HSMC20CP](#) oder [HSMC-20-98-P](#)

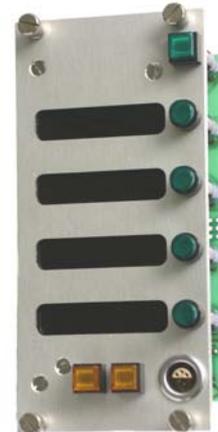
**DSP** Displaymodul mit zusätzlichen Funktionen

Tasten mit Leuchtfunktion für lokale Bedienung

Je 2 analoge Ein- und Ausgänge sowie 3 digitale Eingänge TTL



**DSP**



Frontansicht

VMIO10-C ohne Periferie-Messerleiste,  
mit Piggyback *DSP Pgb* und Displayfront

- Kanalzahl 4
- Grösse 3HE, 12TE
- Steckplatz 3 res. Positionen, die 2 linken Slots im Bereich ohne Printführungen !
- Displaytyp alfanumerisch, 8 Stellen, LED
- Anschluss 1 Buchse LEMO1 6p
- Tasten
  - grün  Systemtaste Offline bzw. Parameter-Reset
  - grün  Achswahl
  - gelb  Bewegungstasten vor- bzw. rückwärts
- Konfiguration Moduladresse

Interface Nr.	Moduladresse	Achse
DSP 1	1B00	1..4
DSP 2	1C00	5..8
DSP 3	1D00	9..12

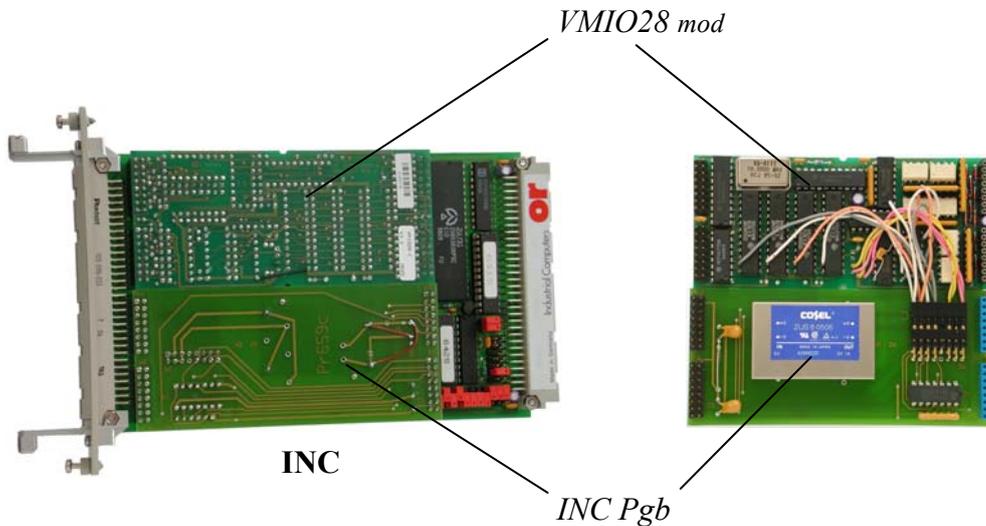
Mit den Tasten des DSP kann eine gewählte Achse im Einzelschritt, mit Start/Stop- oder mit Positionierfrequenz vor- oder rückwärts bewegt werden.

*Achtung !* Im Openloop-Betrieb ( ohne Encoder ) und während dem Freifahren von Schaltern bleibt die Positions-Anzeige unverändert, solange der Motor in Bewegung ist.

Die Endschalter-, Stop- und Notstop-Funktionen sind ohne angeschlossenen Periferiestecker frei. Die Bewegungen sind nur durch die Software-Limiten begrenzt.

Für einen vollen Funktionstest ist der Anschluss muss ein IO-Interface samt Periferiebeschaltung angeschlossen werden.

**INC** Interface für Inkremental-Encoder mit RS422 Schnittstelle



Original VMIO10-C mit Piggyback *VMIO28 mod* und *INC Pgb*

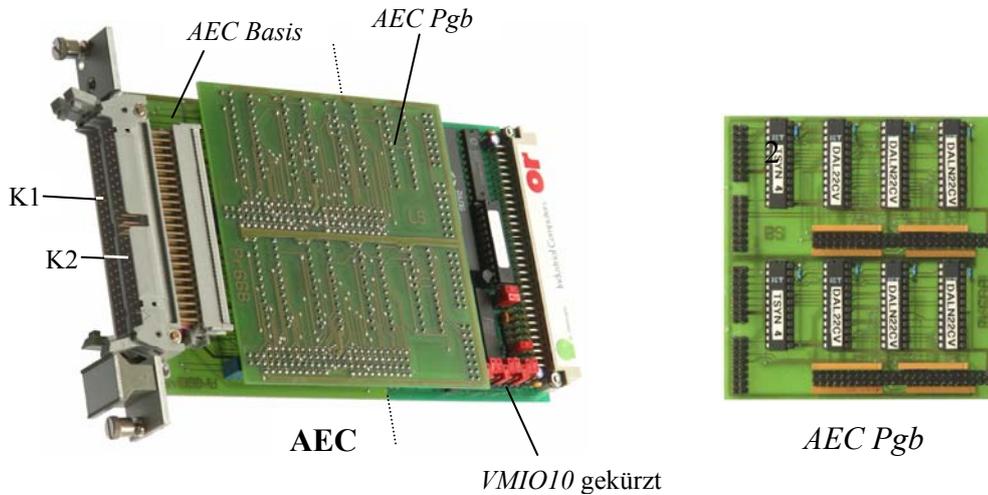
Typencode	1
Kanalzahl	4
Grösse	3HE 4TE
Steckplatz	variabel
Schnittstelle	4 x RS422, A, B und Index
Encoderspeisung	5V total 1A, 4 Kan. gemeinsam
Kopplung	Potentialtrennung
Interpolation	x 2
Anschluss	Messerleiste DIN41612 Bauform C 3 x 32p
Konfiguration	Moduladresse

Interface Nr.	Moduladresse	Encoder
INC 1	1E00	1..4
INC 2	1F00	5..8
INC 3	2000	9..12

- Funktionstest** VME-Einheit mit CPU, DSP und INC mit Testencoder benutzen  
 Verbindung mit PC für Parameter-Eingabe  
 ( Leistungseinheit und Motor nicht notwendig )
- Steuerung Online schalten ( z.B. RMT 2 )
  - Encoderkanäle dem Display zuordnen ( z.B. EC 1 1 1 ... )
  - Konfiguration anzeigen ( CFG )
  - Display zeigt Encoder-Position in Encoder-Steps ( DS )

**Doku** ext. [Technische Dokumentation](#) **VMIO28** Piggybackmodul zu VMIO10

**AEC** Interface für absolute Pin-Encoder mit V-Abtastung ( Litton )



VMIO10-C gekürzt und verschraubt mit AEC Basis

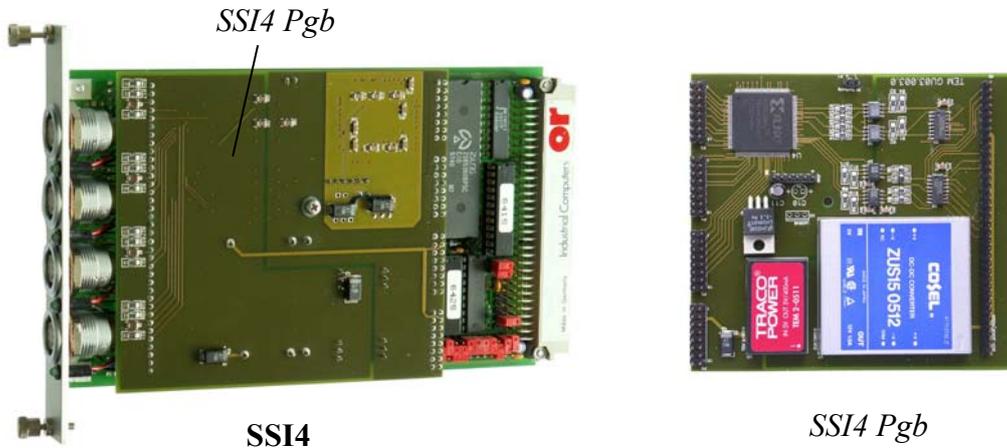
Piggyback *AEC Pgb*

- Typencode 2
- Kanalzahl 2
- Grösse 3HE 4TE
- Steckplatz variabel
- Schnittstelle 5V-Logik, Schalter gegen Common  
V-Abtastung, 1 .. 17 Bit
- Encodertypen Litton 11Bit od. 17Bit
- Kopplung keine Potentialtrennung
- Anschluss 2 x Flachbandkabel-Steckerleisten 50p
- Konfiguration Moduladresse

Interface Nr.	Moduladresse	Encoder
AEC 1	2400	1, 2
AEC 2	2500	3, 4
AEC 3	2600	5, 6
AEC 4	2700	7, 8
AEC 5	2800	9, 10
AEC 6	2900	11, 12

- Funktionstest**
- VME-Einheit mit CPU, DSP und AEC mit Testencoder benutzen
  - Verbindung mit PC für Parameter-Eingabe  
( Leistungseinheit und Motor nicht notwendig )
  - Steuerung Online schalten ( z.B. RMT 2 )
  - Encoderkanäle dem Display zuordnen ( z.B. EC 1 2 1 ... )
  - Display zeigt Encoder-Position in Encoder-Steps ( DS )

**SSI4** Interface für Absolut-Encoder mit SSI-Schnittstelle



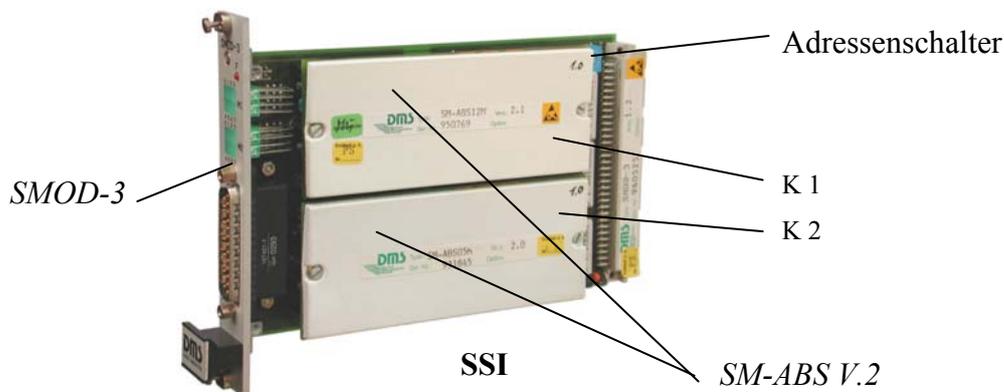
VMIO10-C modifiziert, mit Piggyback *SSI4 Pgb*

- Typencode 4
- Kanalzahl 4
- Grösse 3HE 4TE
- Steckplatz variabel
- Anschluss 4 Buchsen LEMO2 8p
- Schnittstelle SSI, RS422
- Encoderspeisung 12V / 1A total
- Encodertypen 1 .. 25 Bit, Gray | Binär, ohne Parität
- Kopplung Potentialtrennung, alle 4 Kanäle gemeinsam
- Konfiguration Moduladresse

Interface Nr.	Moduladresse	Encoder
SSI4 1	2100	1..4
SSI4 2	2200	5..8
SSI4 3	2300	9..12

- Funktionstest**
- VME-Einheit mit CPU, DSP und SSI4, Verbindung zum PC
  - Steuerung Online schalten ( z.B. RMT 2 )
  - Encoderkanäle dem Display zuordnen ( z.B. EC 1 4 1 ... )
  - Konfiguration anzeigen ( CFG )
  - Display zeigt 0.000 ( ohne Encoder )
  - Positionsanzeige 0.000 grüne LED Off
  - LEMO-Buchse P0 mit P8 verbinden
  - => Anzeige 11184.810 grüne LED On
  - ( gilt mit defaultconfig. 24Bit, Gray )
  - Encoder anschliessen => Display zeigt Messwert
  - in Encoder-Steps ( DS ) , grüne LED On

**SSI** Interface für Absolut-Endoder mit SSI-Schnittstelle  
*DMS* Dorsch Mikrosystem GmbH



- VMEBus-Karte *SMOD-3* mit 1 od. 2 Kanalmodulen *SM-ABS V.2*  
 Typencode 3  
 Kanalzahl 2  
 Grösse 3HE 4TE  
 Steckplatz variabel  
 Anschluss 1 DSub 25p m  
 Schnittstelle SSI, RS422  
 Encoderspeisung 12V / 250mA  
 Kopplung Potentialtrennung für jeden Kanal  
 Encodertypen Software konfigurierbar  
 15Bit, 16Bit, 17Bit, 19Bit, 21Bit, 24Bit  
 mit | ohne Parität, Gray | Binär  
 \*25Bit ( ROQ425 ) mit spezieller Verdrahtung als 24Bit-Typ möglich  
 Konfiguration Moduladresse

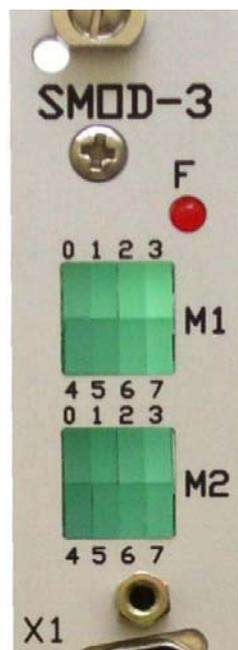
Interface Nr.	Moduladresse	Encoder
SSI 1	<b>80000</b>	1, 2
SSI 2	<b>81000</b>	3, 4
SSI 3	<b>82000</b>	5, 6
SSI 4	<b>83000</b>	7, 8
SSI 4	<b>84000</b>	9, 10
SSI 6	<b>85000</b>	11, 12

Die markierte Ziffer der Adresse ( Fettdruck ) wird am blauen 16 Stufen-Schalter eingestellt

## Funktionstest

VME-Einheit mit CPU, DSP und SSI, sowie Testencoder, Verbindung zum PC  
( Leistungseinheit und Motor nicht notwendig )

- nach Einschalten der Netzspeisung leuchtet die rote Fehler-LED des SSI-Moduls kurz auf.
- Steuerung Online schalten ( z.B. RMT 2 )
- Encoderkanäle dem Display zuordnen ( z.B. EC 1 3 1 ... )
- Konfiguration anzeigen ( CFG )
- Display zeigt ohne angeschlossenen Encoder 11184.8
- Auf *SMOD-3* zeigen LED 2 das Takt-Signal, LED 6 die 12V- und LED 7 die 5V-Betriebsspannung
- Encoder anschliessen => Display zeigt Messwert in Encoder-Steps ( DS )
- LED 0 auf *SMOD-3* zeigt das Encoder-Datensignal ( flackert leicht bei änderndem Messwert )



**Doku** ext. Technische Dokumentation [SM-ABS](#) Piggybackmodul zu **SMOD-3**

## Moduladressen

An jedem Modul eines Stepmotor-Controllers muss eine individuelle Basis-Adresse nach Tabelle eingestellt werden.

Den verschiedenen Modultypen sind Adressbereiche zwingend zugeordnet. Es können also nur die in der Tabelle angegebenen Werte benützt werden.

Die Adressen sind unabhängig von den Steckpositionen der einzelnen Module im VMEbus.

Die aus der Moduladresse resultierenden Encoder-Interfacenummern bestimmen nicht die Zugehörigkeit der Encoder zu den Motor-Achsen. Diese Zuordnung erfolgt durch Software-Konfiguration ( Parameter EC ).

Adressen werden als Hexadezimalzahlen angegeben.

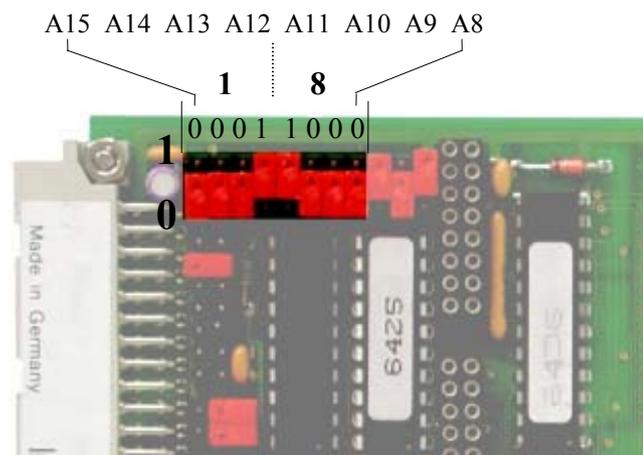
- Die Adressen der nicht auf der VMIO10 - Karte basierenden SSI-Module von DMS werden nach Tabelle am blauen Drehschalter S10 auf der SMOD-3 Karte eingestellt ( markierte Ziffern in der Tabelle )
- Auf der VMIO-Karte aller andern Module ( MOT, DSP, IEC, SSI4, AEC, AUX ) wird nur der in der Tabelle markierte Teil der Adresse bitweise mit 8 Jumpern eingestellt.

**ACHTUNG** nur die 8 im Bild markierten Jumper bilden die Moduladresse.

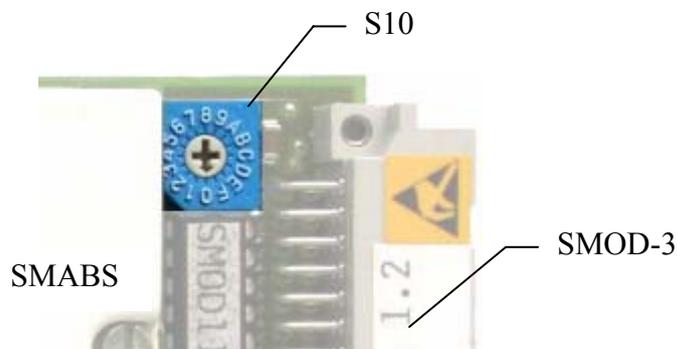
Alle andern Jumper bleiben in der den dargestellten Positionen.

( Gilt für alle im Stepmotor-Controller EL734 eingesetzten VMIO10-Karten )

Beispiel mit Adresse FF1800h  
( Adresse des ersten MOT Moduls )



Das DMS Interface-Modul SSI für 2 Encoder mit SSI-Interface hat als Basis die Buskarte SMOD-3. Die Basisadresse 800000H ist auf dieser Karte durch Lötbrücken fest vorgegeben. Die individuellen Moduladressen 0 .. 5 werden am blauen Drehschalter S10 eingestellt.



### Moduladressen Tabelle

Interface		Achsen	Adresse	Jumper
<b>MOT 1</b>	Step-Motor	1 - 4	FF <b>18</b> 00	0001 1000
<b>MOT 2</b>		5 - 8	FF <b>19</b> 00	0001 1001
<b>MOT 3</b>		9 - 12	FF <b>1A</b> 00	0001 1010
<b>DSP 1</b>	Display	1 - 4	FF <b>1B</b> 00	0001 1011
<b>DSP 2</b>		5 - 8	FF <b>1C</b> 00	0001 1100
<b>DSP 3</b>		9 - 12	FF <b>1D</b> 00	0001 1101
<b>IEC 1</b>	Increment. Encoder	1 - 4	FF <b>1E</b> 00	0001 1110
<b>IEC 2</b>		5 - 8	FF <b>1F</b> 00	0001 1111
<b>IEC 3</b>		9 - 12	FF <b>20</b> 00	0010 0000
<b>SSI4 1</b>	SSI4 4 Kanal Encoder	1 - 4	FF <b>21</b> 00	0010 0001
<b>SSI4 2</b>		5 - 8	FF <b>22</b> 00	0010 0010
<b>SSI4 3</b>		9 - 12	FF <b>23</b> 00	0010 0011
<b>AEC 1</b>	Absolut Encoder (Litton)	1, 2	FF <b>24</b> 00	0010 0100
<b>AEC 2</b>		3, 4	FF <b>25</b> 00	0010 0101
<b>AEC 3</b>		5, 6	FF <b>26</b> 00	0010 0110
<b>AEC 4</b>		7, 8	FF <b>27</b> 00	0010 0111
<b>AEC 5</b>		9, 10	FF <b>28</b> 00	0010 1000
<b>AEC 6</b>		11, 12	FF <b>29</b> 00	0010 1001
<b>AUX</b>	Auxiliary		FF <b>2A</b> 00	0010 1010
<b>SSI 1</b>	DMS Encoder m. SSI- Schnittstelle	1, 2	8 <b>0</b> 0000	0 Switch
<b>SSI 2</b>		3, 4	8 <b>1</b> 0000	1 S10
<b>SSI 3</b>		5, 6	8 <b>2</b> 0000	2
<b>SSI 4</b>		7, 8	8 <b>3</b> 0000	3
<b>SSI 5</b>		9, 10	8 <b>4</b> 0000	4
<b>SSI 6</b>		11, 12	8 <b>5</b> 0000	5

# Leistungsteil

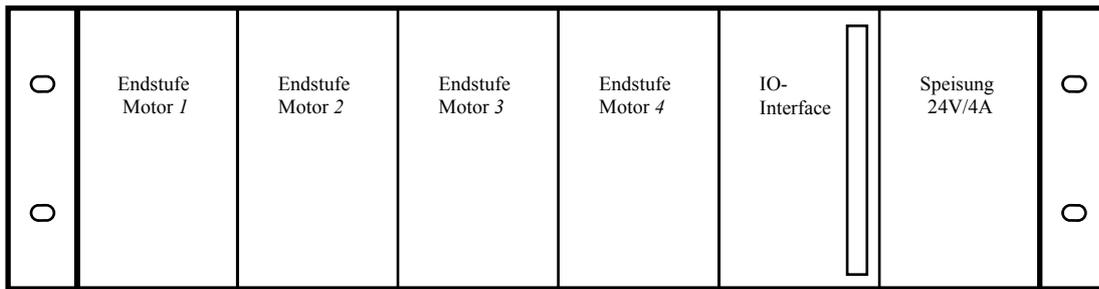
## Leistungseinheit 4 x 2Ph / 5Ph

Die Leistungseinheiten werden im Rack über der VME-Steuereinheit auf Führungsschienen montiert.

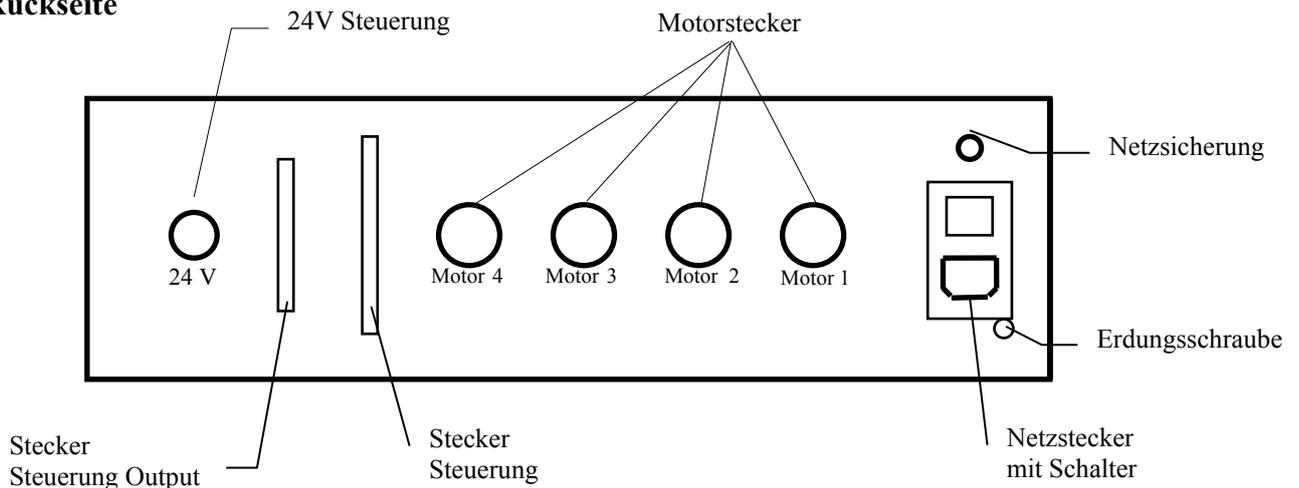
Zwischen der VME-Steuereinheit und der ersten Leistungseinheit ist ein Zwischenraum von 2 HE für die Durchführung der Encoder-Kabel vorgesehen.

Die Leistungseinheiten werden je nach den verwendeten Motortypen ( 5 oder 2 Ph ) individuell mit Endstufenmodulen bestückt. Dazu sind zuvor die Kartenführungen in den entsprechenden Positionen einzusetzen.

### Frontseite



### Rückseite



### Speisung Varianten

Trafo / Gleichrichter	60V unstab	24V/ 4A unstab.
Ringkerntrfo / Gleichrichter	60V unstab	24V/ 4A unstab.

Jede Leistungseinheit ist mit einem IO-Interfacemodul und einem 24V-Netzspeisemodul ausgerüstet.

Die 4fach-Leistungseinheit kann wahlweise mit 5-Phasen- oder 2-Phasenmodulen bestückt werden. Endstufen-Module müssen passend Schrittmotoren der jeweiligen Achsen gewählt werden.

Um Verwechslungen zu vermeiden, an den frei bleibenden Steckpositionen keine Kartenführungen einsetzen!

### Warnung !

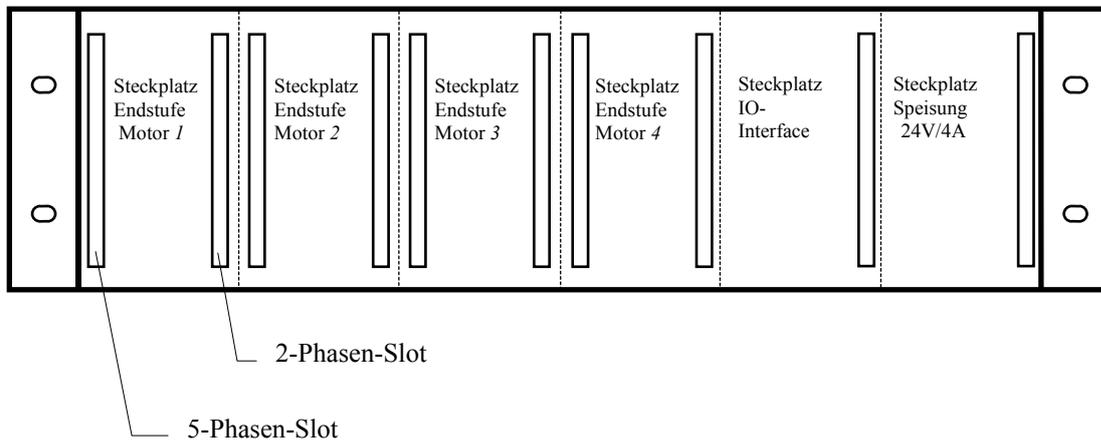
**Die Anschlüsse der Speisung sind beim 2Ph- und beim 5Ph-Modul unterschiedlich.  
Ein in den falschen Slot gestecktes Modul kann nach dem Einschalten der Netzspannung sofort einen irreparablen Defekt erleiden !**

**Endstufen-Module und Motorstecker dürfen nur in stromlosem Zustand der Leistungseinheit ein-oder ausgesteckt werden.**

**Unter Strom können Steck-Kontakte zerstört werden.**

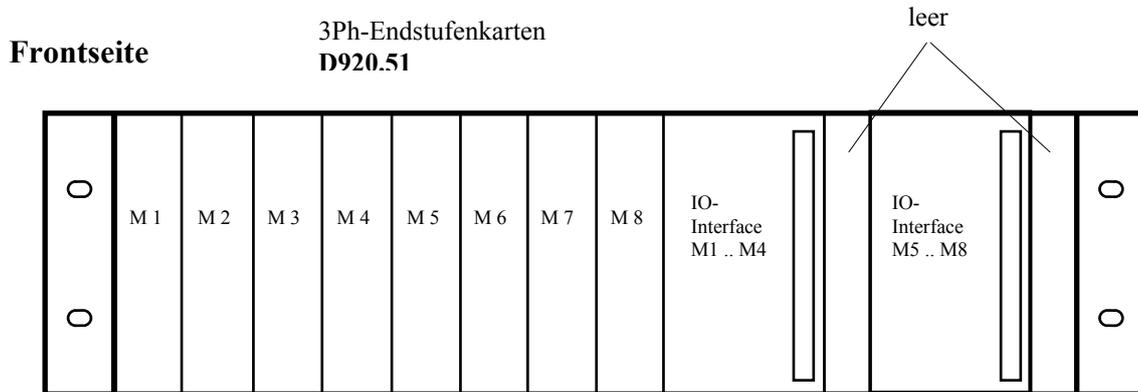
**Zur Entladung der Kondensatoren des Netzteils mindestens 1 Minute abwarten !**

### Frontansicht Leistungseinheit 4 x 2Ph / 5Ph ohne Bestückung

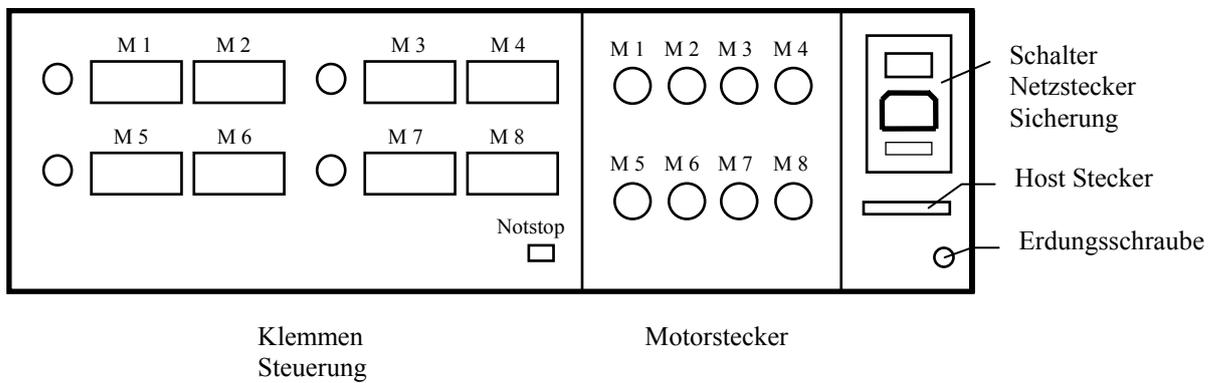


# Leistungseinheit 8 x 3Ph

Nur für 3Ph-Schrittmotoren



## Rückseite



IN	Out	Fuse
1 LS-	7 ACO	4 x 1AT
2 LS+	8 OUT	
3 STOP		
4 REF		
5 ACI		
6 IN		

**Speisung**

PULS SL30 40V stab.

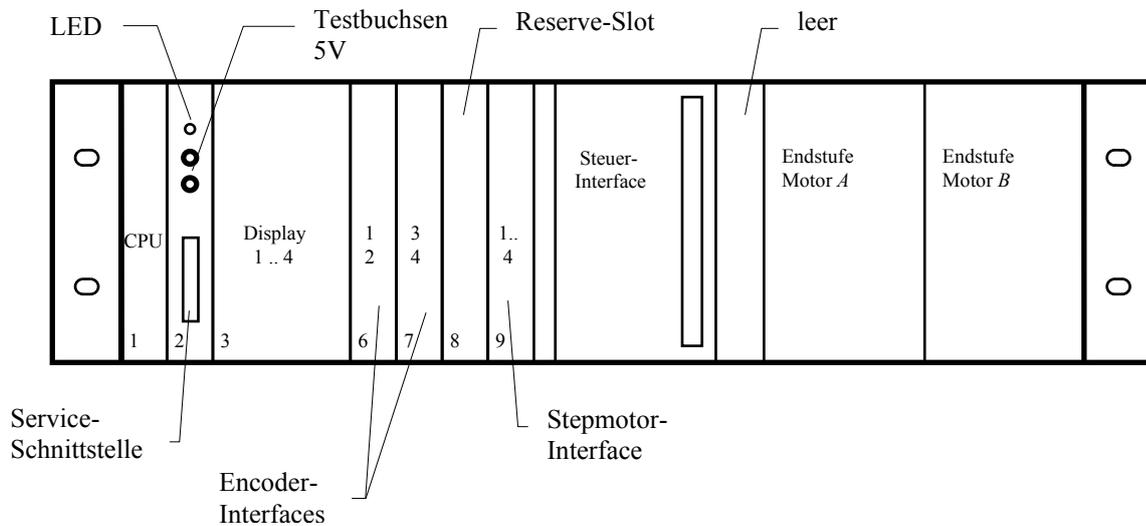
24V/ 4A stab.

## Kompaktsteuerung

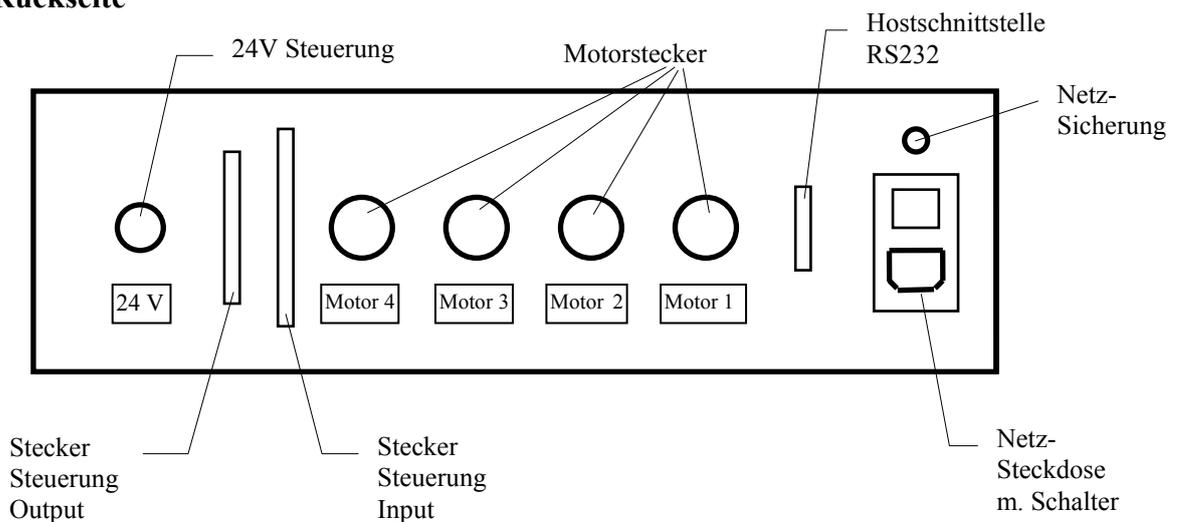
Die Kompaktsteuerung ist Vereinigung von VME-Steuerteil und Leistungsteil für 2 ( 4 für 2Ph-Antriebe ) Achsen in einem einzigen Gerät. ( 19" , 3HE , 550mm Tiefe )

Die Geräte sind mit PULS-Netzteilen ( stabilisiert ) oder teilweise mit Ringkern-Trafo Speisungen zur Versorgung der Motor-Endstufen ausgerüstet. Alle 24V-Speisungen sind stabilisiert.

### Frontseite



### Rückseite



## Module der Leistungseinheit

**IO** Pegelanpassung und Potentialtrennung für alle IO-Signale  
LED-Statusanzeigen und Abschaltmöglichkeit für jede Achse



**IO**

Grösse	3HE 14TE	
Kanalzahl	4	
Steckplatz	im Leistungsteil, fix	
Anschluss Front	Messerleiste DIN41612 Bauform C 3 x 32p	
Anschluss Back	Messerleiste DIN41612 Bauform C 3 x 32p	
Steuersignale	Endschalter	LS1 LS2
	Stop	S
	Referenz	LSX
	Luftkissen	ACI
	Eingang	IN
Schaltausgänge	Luftkissen	AC
	Ausgang	OUT
Stepmotor	Endstufensignale	Puls Richtung
Logikstandard	Schalter-Eingänge	24V
	Schalt-Ausgänge	24V / 1 .. 2A ( einstellbar )
	Motor Puls / Richtung	TTL
Kopplung	Potentialtrennung	alle 4 Kanäle gemeinsam
LED-Anzeigen	Bewegung	-Run +Run
	Referenz-Signal	XLS
	Endschalter	LS1 LS2
	Luftkissen-Ausg.	AC
	Luftkissen-Eing.	ACI
	Ausg.	OUT
	Eing.	IN
	Motorfunktion	Disable (Stop) Pwr Off (Phasenstrom Aus)
	Schnellstop	Emergency

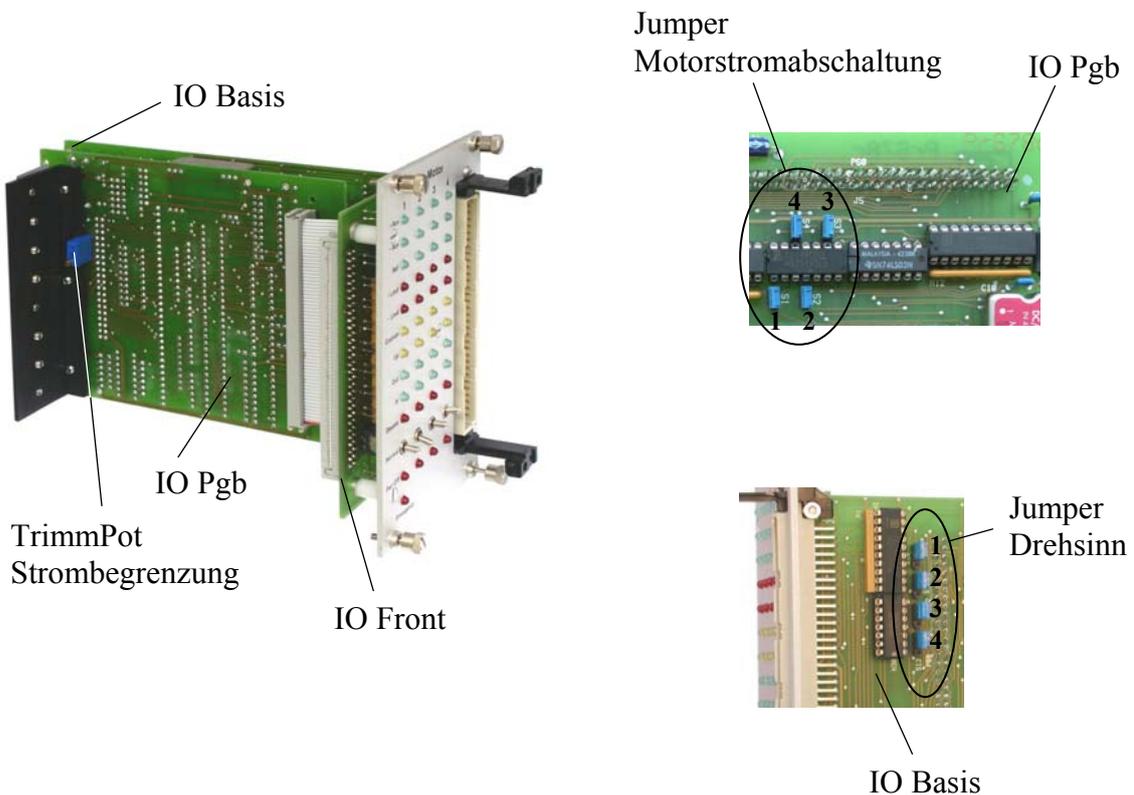
Schalter                      Motorfunktion                      Disable | Normal | Pwr Off

Schutz der Schalt-Ausgänge

Überspannung                      Zenerdioden  
 Kurzschluss                      einstellbare Strombegrenzung  
 Überlastung                      Übertemperatur-Abschaltung

Konfiguration

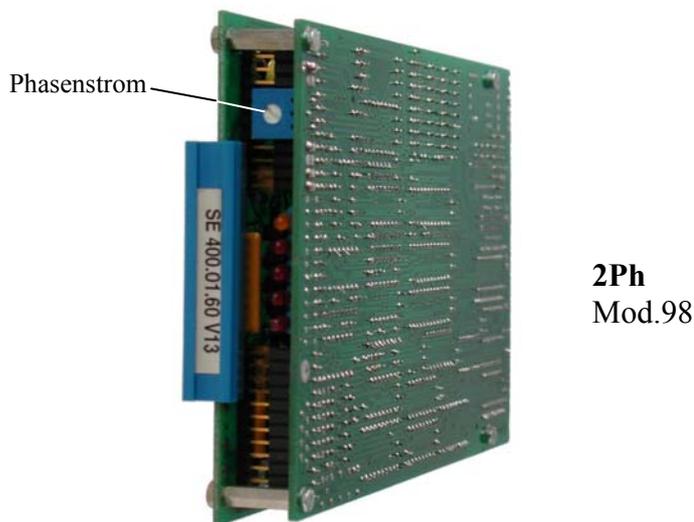
- Motor-Drehsinn  
 ( wird meistens am Endstufenmodul oder beim Anschluss des Motors eingestellt )
- Motorstromabschaltung  
 bei gesetztem Jumper lässt sich der Phasenstrom mit dem Signal  $OUT_m$  ( Befehl  $SO_m 1 | 0$  ) ein | aus schalten  
 die Stromabschaltung kann zudem mit dem Befehl  $ACS_m 1$  automatisiert werden  
 Anmerkung : bei gesetztem Jumper  $m$  kann das Signal  $OUT_m$  nicht für andere Aufgaben verwendet werden
- Strombegrenzung Schalt-Ausgänge  
 mit dem Trimpotentiometer kann die Strombegrenzung für die 4 AC- und die 4 OUT-Ausgänge des IO-Moduls gemeinsam eingestellt werden ( 1 .. 2 A )



**2Ph SE 400.01.60 Omni Ray**

Schrittmotor-Steuerkarte für 2 Ph-Motoren bis 1A

Anmerkung es sind verschiedene Versionen mit unterschiedlichem Print-Layout in Gebrauch



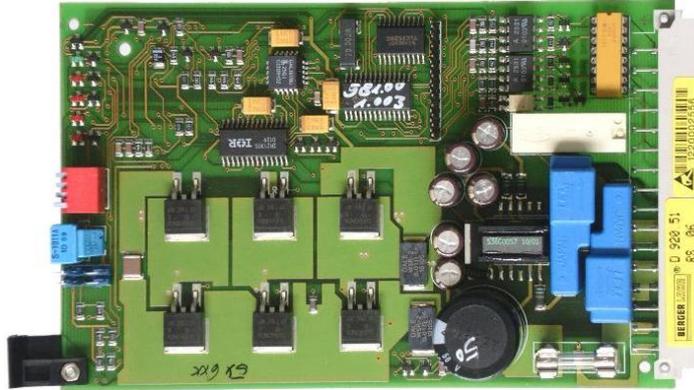
Grösse	3HE 5TE
Speisung	42 .. 60V
Stepzahl	400 MS / U
Phasenstrom	0..1A
Interface	5 .. 30V, keine Potentialtrennung
Schrittauflösung	400MS / U

Konfiguration	Lötbrücken
Eingangsschaltung	Low aktiv
Schrittauflösung	400MS / U
Stromabsenkung	ON

**Doku** ext. [Technische Dokumentation](#)  
 Schrittmotor-Leistungsverstärkerkarte  
 Serie SE ... V11 und SE ... V13  
 Zebotronics  
 Wild & Grabmaier Feinmechanik und Elektronik KG  
 München

**3Ph**     **D920.51**     Berger Lahr

Schrittmotor-Steuerkarte für 3 Ph-Motoren mit LH-Wicklung



**3Ph**

Art.Nr.	62 010 920.513
Grösse	3HE 6TE
Speisung	18..40V
Interface	5V, Optokoppler
Schrittauflösung	1000MS / U
Konfiguration	Werkseinstellung ( ausser Stromeinstellung )



- OK
- Kurzschluss
- Übertemperatur
- Überspannung
- Unterspannung
  
- Schrittzahl **Dip 1**
- Schrittzahl **Dip 2**
- Stromabsenkung**
- Tor/Enable**

Phasenstrom

beide Haken **offen**

**Doku** ext. [Technische Dokumentation](#)  
 Leistungssteuerkarte f. 3Ph-Schrittmotoren D920  
 DS-Nr. 220.920/Doku 8.97  
 Berger Lahr

**5Ph**    **D 550.01**    Berger Lahr  
 Schrittmotor-Endstufe für 5 Ph-Motoren mit LH-Wicklung



**5Ph**

Art.Nr.	620 105 500 10
Grösse	3HE 14TE
Speisung	35..70V
Interface	5V, Optokoppler
Schrittauflösung	1000MS / U
Schrittauflösung	1000MS / U
Konfiguration	Werkseinstellung ( ausser Drehrichtung und Strom-Einstellung )

- OK
- Kurzschluss
- Übertemperatur
- Unt/Üb-Spannung
- Phasenunterbruch
- keine Funktion



- PWM-Stromsteuerung
- Stromabsenkung aktiv
- Drehrichtung rechts
- Halbschritt

Phasenstrom

**Doku** ext. [Technische Dokumentation](#)  
 Leistungssteuerekarte f. 5Ph-Schrittmotoren  
 D55.00/01  
 DS-Nr. 220.550/DD 12.89  
 Berger Lahr

## TWINLINE

Grössere 3Ph Schrittmotoren von Berger Lahr ( VRDM 31117 mit 12Nm oder VRDM 31122 mit 16Nm ) werden mit der Leistungselektronik Typ TLD012 betrieben.

Die Leistungselektronik wird über eine TWINLINE Adapterkarte an die Schrittmotorsteuerung angeschlossen.

Der Adapter stellt die Signale Puls, Richtung und Stromabschaltung am 15P-DSub-Stecker RS422-kompatibel, und an der LEMO-Buchse die 24V-Speisung für den Steuerteil der TWINLINE-Einheit zur Verfügung.

### Adapterkarte

Die Adapterkarte wird im Endstufenchassis der Schrittmotorsteuerung anstelle einer 3Ph-Leistungskarte Typ D920 gesteckt.

Im Endstufenchassis müssen die gr und vi Adern der farbigen Flachband-Steuerkabel ( 0V der Endstufen-Steuersignale ) mit 16a/c der Endstufen-Printstecker verbunden werden ( 0V der Motorspeisung ).

Die grüne LED zeigt die Betriebsbereitschaft der TWINLINE-Einheit an ( Motorstrom On ).

### Installation der TWINLINE-Einheit

Die TWINLINE-Einheit wird separat im Steuerschrank montiert.

Sie hat einen eigenen Netzanschluss (230V).

Mit dem TWINLINE-Adapter wird sie über zwei Kabel ( Signal und 24V Speisung ) verbunden.

Am TWINLINE-Modul wird Dip-Schalter 5 in Stellung Off gesetzt ( Enable-Funktion ), alle andern in Stellung ON.

### Betriebsanzeigen der TWINLINE-Einheit

detaillierte Beschreibung in der Technischen Dokumentation TLD01x von Berger Lahr

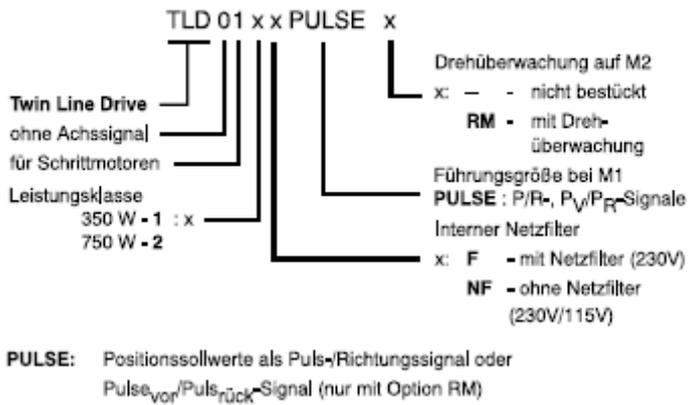
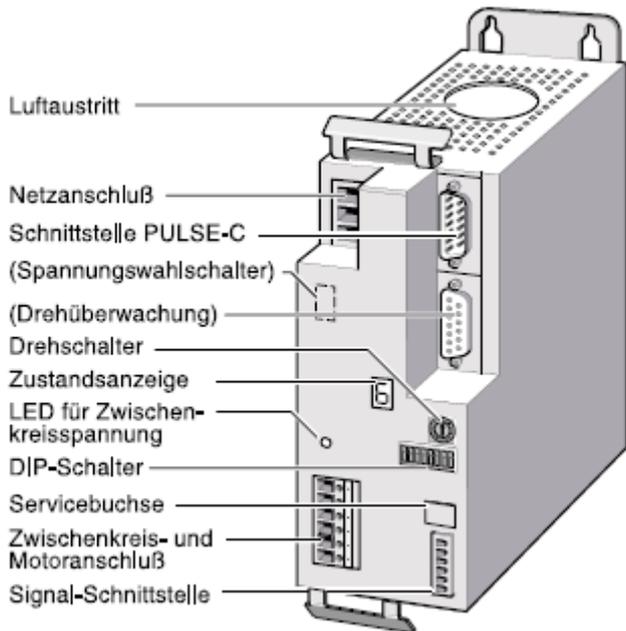
#### Kap. 7 Diagnose und Fehlerbehebung

Die wichtigsten Zustände ( ander Zustandsanzeige der TWINLINE-Einheit angezeigt ) sind

6	Betriebsbereit
3	Motorstrom aus ( Schalter am IO-Interface in Stellung Power Off )
1 blinkend	keine Netzspannung an der Leistungselektronik

Fehlerzustände der Leistungselektronik ( blinkende Zustandsanzeige ) müssen durch kurze Betätigung des entsprechenden Schalters am IO-Interface quittiert werden ( Power Off und zurück ).

**TWINLINE Drive TLD01x**



**Doku** ext. [TechnischeDokumentation](#)

Twin Line Drive 01x

Leistungselektronik für Schrittmotoren

**TLD01x**

Betriebssystem: 1.1xx

Bestellnr.: 9844 1113 101

Ausgabe: d107, 09.02

Berger Lahr

## **Verbindungskabel**

Die Leistungseinheiten werden über je ein Verbindungskabel mit dem zugehörigen Motorcontroller-Modul in der VME-Steuereinheit verbunden. Die Länge der Kabel muss genau angepasst werden.

## **Netzanschluss**

Werden in einer Anlage zwei oder drei Leistungseinheiten eingesetzt, so sollen diese zur besseren Lastverteilung aus verschiedenen Phasen gespeist werden.

## **Motoranschluss**

Die Schrittmotoren werden über abgeschirmte Kabel mit paarweise verdrehten Adern angeschlossen. Die Kabelabschirmung wird mit dem Steckergehäuse verbunden.

## **Anschluss Steuerung Ausgänge**

Zum Anschluss der Steuerausgänge dient ein im Rack montierter Verteiler-Klemmenblock, welcher über Flachbandkabel mit der Leistungseinheit verbunden ist. Die Abschirmungen der externen Signalkabel werden über die Masseklemmen verbunden.

## **Anschluss Steuerung Eingänge**

Zum Anschluss der Steuereingänge dient ein im Rack montierter Verteiler-Klemmenblock, welcher über Flachbandkabel mit der Leistungseinheit verbunden ist. Die Abschirmungen der externen Signalkabel werden über die Masseklemmen verbunden.

## **24V**

An der 24V-Steckdose steht die unregelte Steuer-Speisepannung zur Verfügung. Die Belastung darf zusammen mit den Ausgangsströmen der Steuerung 4A nicht überschreiten.

## Software

Aktuelle Softwareversion **V3.2**

Softwarevariante für  $\mu$ E1 **V3.2 $\mu$**  ( die Module **McMain** und **EncoderA** sind durch Spezial-Versionen ersetzt )

Programmiersprache für den Sourcecode	MODULA-2
Entwicklungs-Software	MODES
wird nicht mehr unterstützt	Modula-2 Cross-Development System for MOTOROLA MC68000 on IBM-PC Version 3.0      Jan. 1991

Die Steuerungs-Software ist in zwei im CPU-Modul gesteckten mit **H** und **L** gekennzeichneten EPROMs gespeichert.

Der Austausch der EPROMs ist unter **VME-Module CPU** beschrieben.

neu in der Version V3.2 sind

- Interface-Driver für SSI4 Encoder-Interface mit SSI-Schnittstellen
- kein System-Reset über die seriellen Schnittstellen mit %
- für den Betrieb mit Magnetbremse kann die Prüfung des Rückmelde-Eingangs ACI im OPT-Register abgeschaltet werden
- direkte Steuerung eines Motors über 2 SPS-Signale ( 24V )
- verbesserte HELP-Funktion ( seitenweise Anzeige der Listen )

## Funktion der Software

Die Software steuert im System die internen Abläufe und den Datenverkehr, sowie die Kommunikation über die seriellen Schnittstellen.

Interface-Drivermodule für folgende Hardware sind enthalten

- 3 **MOT** Stepmotor-Interfaces
- 3 **DSP** Displays
- 3 **INC** Inkrementalgeber-Interfaces
- 3 **SSI4** Encoder-Interfaces mit SSI-Schnittstelle
- 6 **AEC** Absolutgeber-Interfaces
- 6 **SSI** Encoder-Interfaces mit SSI-Schnittstelle ( DMS )
- 2 RS232 serielle Schnittstellen ( im CPU-Modul )
- Tastensteuerung ( im DSP-Modul )

## Systemstart

Beim Systemstart wird die Hardware-Konfiguration des Systems durch die autom. Modulerkennung überprüft.

Wenn zuvor Interface-Module entfernt oder neu eingesetzt, oder wenn Moduladressen geändert wurden, wird ein Kaltstart ausgeführt.

alle Positionswerte werden gelöscht und die aktuellen Parameter durch ihre Defaultwerte oder, wenn ein AUX-Modul eingesetzt ist, durch die im EEPROM-Modul gespeicherten Werte ersetzt.

Ein Kaltstart kann mit den Display-Bedienungstasten auch manuell ausgelöst werden.

( am Displaymodul der Achsen 1 – 4 die grüne quadratische Systemtaste und die unterste grüne runde Taste ( Zeile 4 ) gedrückt halten und den CPU-Schalter nach oben kippen )

Wenn die Modulerkennung keine Änderung der Konfiguration findet, wird ein Warmstart durchgeführt, d.h. die aktuellen Parameter und Positionswerte werden aus dem batteriegepufferten RAM unverändert übernommen.

**Ausnahmen** : **DSP** und **AUX**-Module werden nicht überprüft.

Für Inbetriebnahme- und Wartungsarbeiten können DSP-Module bestückt

od. entfernt werden ( bei ausgeschalteter Stromversorgung ! ) ohne Verlust der Konfigurationsdaten beim nachfolgenden Neustart der Steuerung

Nach der Initialisierung aller VME-Module und der Ausgabe von Einschaltmeldungen auf die beiden seriellen Schnittstellen und das erste Displaymodul ( M1 .. M4 ) wird der Monitorloop ausgeführt.

## Monitorloop

Im Monitorloop werden periodisch die Prozeduren aufgerufen für

- Schnittstellenkommunikation und Befehlsinterpretation
- Starten von Funktionen
- Tasteneingaben
- Steuern und Überwachen der Achsenaktivitäten
- Ablesen und Anzeige der Achspositionen

Die Zykluszeit liegt, abhängig von der Belastung durch die Aktivitäten, im Bereich von 10 bis 100 ms

## Befehlssatz

<b>A</b>	Nachkommastellen	<b>OPT</b>	Optionen-Register
<b>AC</b>	Luftkissen	<b>P</b>	positionieren
<b>ACS</b>	aut. Motorstromabschaltung	<b>PD</b>	positionieren hex
<b>AM</b>	aktive Motoren	<b>PR</b>	positionieren relativ
<b>CFG</b>	Systemkonfiguration	<b>Q</b>	Referenzparameter
<b>CLS</b>	alle Ereigniszähler löschen	<b>R</b>	Referenzlauf
<b>D</b>	Schleppfehlertoleranz	<b>RF</b>	Referenz-Testlauf
<b>DE</b>	Default	<b>RI</b>	Read Input
<b>E</b>	Beschleunigung	<b>RMT</b>	Remote
<b>EC</b>	Encoderzuordnung	<b>S</b>	Stop
<b>ECHO</b>	Echomode	<b>SA</b>	Luftkissenfehler
<b>EP</b>	Encoder-Programmiercode	<b>SN</b>	System-Resets
<b>ERR</b>	Text zu Fehler- od. End-Meldung	<b>SP</b>	Positionierungen
<b>F</b>	closed Loop	<b>SR</b>	Pos.-Failures
<b>FB</b>	fast backward	<b>ST</b>	Position-Faults
<b>FF</b>	fast forward	<b>SW</b>	Watchdog-Resets
<b>FD</b>	Encoderfaktor	<b>SB</b>	Slow backward
<b>FM</b>	Motorfaktor	<b>SF</b>	Slow forward
<b>G</b>	Start/Stop-Frequenz	<b>SO</b>	Output setzen
<b>H</b>	Softwarelimiten	<b>SS</b>	show Status
<b>ID</b>	Software-Version	<b>SYR</b>	System-Register
<b>J</b>	Maximal-Frequenz	<b>T</b>	einseitig anfahren
<b>K</b>	Referenzlauf-Modus	<b>U</b>	Position
<b>L</b>	Spielausgleich	<b>UD</b>	Position hex
<b>M</b>	Positionier-Toleranz	<b>UU</b>	Position H, V fix
<b>MEM</b>	Memo-Text	<b>V</b>	Encoder-Nullpunkt
<b>MN</b>	Motorname	<b>W</b>	Luftkissenlogik
<b>MSR</b>	Motor-Statusregister	<b>Z</b>	Encoderumfang
<b>N</b>	Endschalter freifahren	<b>? ?C ?M</b>	Helpfunktionen

## Befehlssyntax

Die Befehle können beliebig mit Gross- oder Kleinbuchstaben geschrieben werden.

Befehlselemente ( Befehl, Achsnummer, Parameter ) werden durch Leerzeichen separiert.

Die Befehls-Sequenz wird abgeschlossen und aktiviert mit der **Enter**- od. **Return**-Taste bzw. dem ASCII-Zeichen **Return**.

In den Tabellen sind die Achsnummern *m* kursiv geschrieben.

Die eckigen Klammern [ ] kennzeichnen optionale Parameter

Die Klammern sind nicht Bestandteil des Befehls und werden nicht eingegeben

Wird der optionale Parameter weggelassen, so wird, mit Ausnahmen bei **CFG**, **RI**, **SO** und **U**, der aktuelle Parameterwert ausgegeben

Auswahlwerte sind in den Tabellen durch | getrennt. z.B. **0|1** ( Wert kann 0 oder 1 sein )

Ein in einer Befehlssequenz eingegebener Parameterwert ersetzt den zuvor im CMOS-Memory gespeicherten Wert.

## Datentypen

**cardinal** ganzzahliger positiver Wert ( Achsnummern )

**integer** +/- ganzzahliger Wert ( Steps )

**real** +/- Wert mit 0 .. 3 Nachkommastellen ( Positionswerte )

## Bezeichnungen

**MS** Motor-Steps

**DS** Encoder-Steps

## Befehle im Detail

### A Nachkommastellen

Syntax **A m [n]** card

Anzahl Nachkommastellen von Positionswerten, die für die Achse  $m$  gerundet ausgegeben werden ( Display und Schnittstellen )

**n = 0, 1, 2, 3** Anzahl Nachkommastellen  
**n = 9** Display-Anzeige - - - für nicht benützten Kanal  
**ohne n** aktuelle Nachkomma-Stellenzahl für die Achse  $m$  anzeigen

Default **n = 3** 3 Nachkommastellen

### AC Luftkissen

Syntax **AC m [0|1]**

direkte Steuerung des Luftkissenschalters  $AC m$  bzw. Abfrage des Luftkissen-Rückmeldeeingangs  $ACI m$

die aktuellen Schaltzustände der Luftkissenschalterschalter können nicht abgefragt werden

**0** Schalter  $AC m \rightarrow$  **Off** bzw. Eingang  $ACI m = L$  0V  
**1** Schalter  $AC m \rightarrow$  **On** bzw. Eingang  $AICm = H$  +24V  
**ohne** aktuellen Schaltzustand von  $ACI m$  anzeigen

Default und nach Systemstart sind alle Luftkissenschalter = **Off**

## **ACS aut. Motorstromabschaltung**

Syntax **ACS *m* [0|1]**

Aktivieren der aut. Motorstrom-Abschaltfunktion für Achse *m*

zusätzlich ist im IO-Interface der entsprechende Jumper zu setzen

diese Funktion benützt den Schaltausgang OUT *m* , welcher dadurch nicht für andere Aufgaben verwendet werden kann

**0** keine Motorstromabschaltung für Achse *m*

**1** Motorstromabschaltung für Achse *m* aktiv, wenn zusätzlich der entsprechende Jumper im IO-Interface gesetzt ist

**ohne** aktuelle Einstellung für die Achse *m* anzeigen

Default = **0** keine Motorstromabschaltung für Achse *m*

## **AM aktive Motoren**

Syntax **AM** hex

zeigt die momentan aktiven Achsen als Bitset in 3-stelligem hex-Format links M12 ... M1 rechts

## **CFG Systemkonfiguration**

Syntax **CFG [*m*]** card

!!! nur im Terminalmode möglich ( ECHO = 1 )

**ohne** Achsnummer *m*

- Anzeige der im System installierten Interfaces als Bitsets  
Format hex 3-stellig ohne Vornullen
- Anzeige der Encoderzuordnungen

**mit** Achsnummer *m*

Anzeige der wichtigsten Parameter für die Achse *m*

## **CLS**     **alle Ereigniszähler löschen**

Syntax     **CLS ALL**

alle Ereigniszähler für alle Achsen M1 .. M12 → **0** setzen

- SA     Luftkissenfehler
- SN     Systemstarts
- SP     Positionierungen
- SR     Position failures
- STPosition faults
- SW     Watchdog-Resets

der Parameter "**ALL**" dient als Sicherheit gegen unbeabsichtigtes Löschen der Zähler

Default                     die Ereigniszähler bleiben unverändert

## **D**     **Schleppfehlertoleranz**

Syntax     **D m [r]**                     real

Testintervall in sec. für die periodische Laufüberwachung im ClosedLoop-Betrieb

r = **0.1 .. 9.9**                     sec

**ohne** r                     aktuellen Wert von r für die Achse m anzeigen

Default     r = **0.1**                     Testintervall = 0.1sec

## **DE**     **Default**

Syntax     **DE m**

alle Parameterwerte der Achse m auf Defaultwerte setzen  
die Text-Register **MN** und **MEM** der Achse m werden gelöscht  
die Ereigniszähler, werden nicht beeinflusst

## **E Beschleunigung**

Syntax **E m [r]** card

Beschleunigungswert in kHz/s für Achse *m*

**r = 1 .. 1000** kHz/s

**ohne r** aktuellen Wert von *r* für die Achse *m* anzeigen

Default **r = 20** Beschleunigung = 20 kHz/s

## **EC Encoderzuordnung**

Syntax **EC m [t n]** card

der Encoder Nr. **n** am Interface vom Typ **t** wird der Achse *m* zugeordnet

**t = 0, 1, 2, 3** Encoder-Interfacetyp

**n = 1 .. 12** sofern vorhanden und nicht belegt

ein der Achse *m* zugeordneter Encoder wird frei gegeben mit

**EC m 0 0**

**ohne t n** aktuelle Werte von *t* und *n* für die Achse *m* anzeigen

Default **= 0 0** keine Encoderzuordnung für die Achse *m*

Interfacetypen

- **0** ohne Encoder, Motorschrittzähler wird verwendet
- **1** Incrementalgeber
- **2** Absolutgeber mit Parallel-Interface ( Litton 13 Bit und 17 Bit )
- **3** Absolutgeber mit SSI-Schnittstelle an **DMS SSI-Interface**
- **4** Absolutgeber mit SSI-Schnittstelle an **SSI4-Interface**

die Encodernummer ist durch den Anschluss des Encoders am Interface definiert

**ECHO Echomode**

Syntax **ECHO [0|1]**

Betriebsart der aktiven RS232 Schnittstelle

- 0** Computermode ohne Echo und ohne Zeilenvorschub
- 1** Terminalmode mit Echo und Zeilenvorschub, sowie zusätzlichem Klartext in den Meldungen
- ohne** Betriebsart der aktiven Schnittstelle anzeigen

nach Systemstart = **1** Terminalmode für beide Schnittstellen

**EP Encoderprogrammiercode** f. SSI4- oder SSI-Interface

Syntax **EP m [l B|G]** card char

- l = 1 .. 26** Wortlänge
- B** Binärcode
- G** Graycode

**ohne** aktuellen Programmiercode für die Achse *m* anzeigen

Default **24 G** f. ROC424 24 Bit Gray-Codierung

alternativ nur für Eingabe in SSI-Interface ( DMS )

Syntax **EP m c** card

Konfigurations-Code SSI-Encoderinterface zur Definition von Wortlänge, Binär- od. Gray-Code und Parität des Encoders für nicht definierte Code-Werte wird automat. der Default-Wert eingesetzt

**c = 0 .. 25** Tabelle in Technische Unterlagen SM-ABSxxM  
4.3 Programmier-Register  
DMS Dorsch Micosystems GMBH

gebräuchliche Werte für *c*

- 1** ROC424, ROQ425 24 Bit Gray Heidenhain
- 13** ROC417 17 Bit Gray Heidenhain

**ERR** Text zu Fehler- od. End-Meldungen anzeigen

Syntax **ERR** *Meldung* Systemmeldung z.B. \*MP, ?CMD ...

!!! nur im Terminalmode möglich ( ECHO = 1 )

**F** closed Loop

Syntax **F** *m* [0|1]

Betrieb ohne od. mit Encoder

automatisch → **1** , wenn der Achse *m* ein Encoder zugeordnet wird

**0** OpenLoop kein Encoder zugeordnet oder ausgeschaltet

**1** ClosedLoop Betrieb mit Encoder

**ohne** aktuellen Wert für die Achse *m* anzeigen

nach Systemstart = **1** wenn der Achse *m* ein Encoder zugeordnet ist

Default = **0** OpenLoop, kein Encoder ist der Achse *m* zugeordnet

**FB** fast backward

Syntax **FB** *m* [*f*] Hz card

**FF** fast forward

Syntax **FF** *m* [*f*] Hz card

Achse *m* frei bewegen rückwärts od. vorwärts mit der aktuellen Maximal-Frequenz bzw. der Frequenz **f**  
Start und Stop der Bewegung mit der eingestellten Beschleunigung

Der Bewegungsbereich ist begrenzt durch die Endschalter und die eingestellten Software-Limiten der Achse *m*

**f** = **33 .. 10000** bewegen mit der Stepfrequenz **f**

**ohne f** bewegen mit der für die Achse *m* gesetzten Stepfrequenz

**FD Encoderfaktor**

Syntax **FD *m* [c n]** int

**FM Motorfaktor**

card

Syntax **FM *m* [c n]**

Umrechnungsfaktoren von Encoder- bzw. Motor-Steps zu grd. oder mm

die Werte werden gebildet aus Getriebe-Übersetzung und Encoder- bzw. Motor-Stepauflösung

c, n bedeuten Zähler, Nenner für Anzahl MS bzw. DS pro grd. od. mm

**FD** der Drehsinn des zugeordneten Encoders kann durch die Wahl des Vorzeichens des Encoderfaktors geändert werden

**FM** der Drehsinn des Motors kann nicht durch das Vorzeichen des Motorfaktors **nicht** geändert werden

**0.002 < c / n < 50000** +- für FD, für FM nur pos.

**ohne** c n aktuelle Werte c und n für die Achse *m* anzeigen

Default c n = **1000 1** 1000 Steps pro Einheit

Anwendungsbeispiele im **Anhang Getriebefaktoren**

**G Start/Stop-Frequenz**

Syntax **G *m* [f]** Hz card

Motor-Stepfrequenz der Achse *m* in Hz im Start/Stop-Betrieb und beim Freifahren der Endschalter

f = **33 .. 10000** Hz

**ohne** f aktuelle Start/Stop-Frequenz für die Achse *m* anzeigen

Default f = **600** 600 Hz

## H Softwarelimiten

Syntax **H m [l h]** real

frei definierbare Wegbegrenzungen für die Achse *m*

l untere Grenze

h obere Grenze

**h > l**

der maximal nutzbare Wertebereich ist abhängig von den eingestellten Faktoren FM und FD, sowie vom verwendeten Encoder

**ohne l h** aktuelle Softwarelimiten für *m* anzeigen

Default l = **-180.000** Arbeitsbereich der Achse *m*  
 h = **360.000** von -180 bis +360 grd. | mm, sofern nicht  
 durch Endschalter enger eingeschränkt

## ID Software-Version

Syntax **ID**

Anzeige der Version der eingesetzten EPROM-Software und der im System existierenden Motorachsen ( hex-Format ohne Vornullen )

## J Maximal-Frequenz ( Positionierfrequenz )

Syntax **J m [f]** Hz card

Motor-Stepfrequenz der Achse *m* im Positionierbetrieb in Hz

f = **33 .. 20000** Hz

**ohne f** aktuelle Stepfrequenz für die Achse *m* anzeigen

Default f = **1000** 1000 Hz

## K Referenzlauf-Modus

Syntax **K m [r]**

Definition des Referenzlauf-Verfahrens beim Betrieb mit Incremental-Encoder

r =	<b>0</b>	kein Referenzlauf	
r =	<b>-1</b>	Rückwärts-Endschalter	
r =	<b>1</b>	Vorwärts-Endschalter	
r =	<b>2</b>	separater Referenz-Schalter	
r =	<b>-11</b>	Rückwärts-Endschalter	} + Indexsignal
r =	<b>11</b>	Vorwärts-Endschalter	
r =	<b>12</b>	separater Referenz-Schalter	
<b>ohne</b>	r	aktuellen Wert von r für die Achse m anzeigen	

Default r = **0**

## L Spielausgleich

Syntax **L m [r]** MS card

r definiert eine Anzahl Motorschritte zur Kompensation von Motorgetriebe-Spiel für die Achse m notwendig sind

wenn die Motorlaufrichtung gegenüber der vorhergehenden Bewegung ändert, macht der Motor zur Kompensation des Getriebespiels r zusätzliche Motorsteps

r =	<b>0 .. 50000</b>	MS Motorsteps
<b>ohne</b>	r	aktuellen Wert von r für die Achse m anzeigen

Default r = **0** kein Spielausgleich

**M Positionier-Toleranz**

Syntax **M m [r]** card

Toleranzwert für die aut. Positionsprüfung beim Positionieren im ClosedLoop-Betrieb für die Achse *m*

**r = 0 .. 50000** DS Encodersteps

**ohne r** aktuellen Toleranzwert für die Achse *m* anzeigen

Default **r = 1** 1 Encoderstep

**MEM Memo-Text**

Syntax **MEM m [t]**

**MN Motorname**

Syntax **MN m [t]**

Text oder Motorname für die Achse *m*

1 Wort ohne Leerzeichen, max. 15 Zeichen  
nur zur Information, keine weiteren Funktionen

**t = Text** max. 16 Zeichen keine Leerzeichen !

**ohne t** Memotext od. Motorname der Achse *m* anzeigen

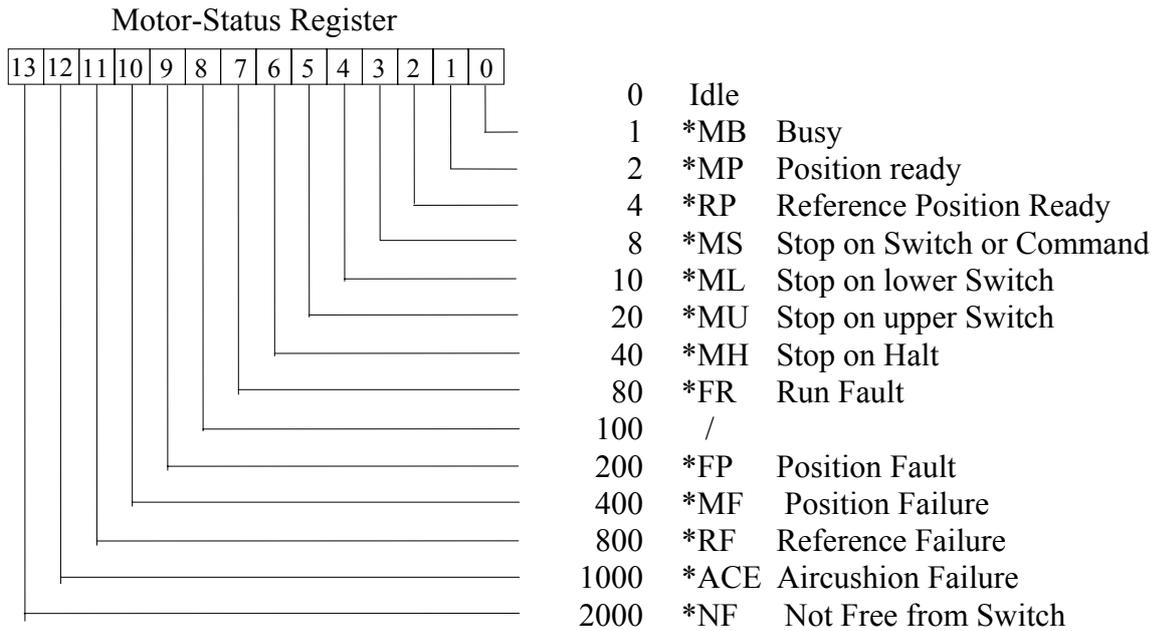
Default kein Memotext bzw. Motorname

**MSR Motor-Statusregister**

Syntax **MSR m** hex

Motor-Statusregister der Achse *m* anzeigen

Der Systemregisterwert wird als Summe der einzelnen zutreffenden Bitwerte ausgegeben



**N Endschalter freifahren**

Syntax **N m**

Achse *m* freifahren nachdem ein Endschalter angefahren wurde  
 ( normalerweise nicht notwendig da aut. freigefahren wird )

**OPT Optionen-Register**

Syntax **OPT m [o]** hex

Register zur Aktivierung spezieller Betriebsarten für die Achse *m*



**o** = Summe hex der einzelnen zu aktivierenden Bitwerte

**ohne o** Optionenregister der Achse *m* anzeigen

Default **o = 0** keine Optionen gesetzt

1	autofreerun	fährt den angefahrenen Endschalter sofort wieder frei	
2	Puls-Kontakt	Motorstart durch ACI-Puls	Vorwärts
		IN-Puls	Rückwärts
4	Dauer-Kontakt	Motorlauf durch ACI-Signal	Vorwärts
		IN-Signal	Rückwärts
8	kein ACI-Test	ACI wird nicht getestet ( z.B. f. Bremse )	

## **P** positionieren

Syntax **P m [r]** real

Achse *m* positionieren auf Position *r* , bzw anzeigen des gesetzten Zielparameterwertes der letzten Positionierung mit der Achse *m*

Die Zielposition muss innerhalb der Softwaregrenzen liegen

*r* = **Zielposition im Realformat** ( grd, mm ... )

**ohne r** für die Achse *m* Zielwert der letzten Positionierung anzeigen

## **PD** positionieren hex

Syntax **PD m [d]** DS hex

Achse *m* positionieren auf Position *d* , bzw anzeigen des gesetzten Zielparameterwertes der letzten Positionierung mit der Achse *m*

Die Zielposition muss innerhalb der Softwaregrenzen liegen

*d* = **Zielwert** des Encoders im **Hex-Format** ( Encodersteps DS )

Datenformat

24 Bit bin. , 2-er Komplement im **Hex-Format**

für positive Werte ist die Eingabe von Vornullen nicht notwendig

**ohne d** für die Achse *m* Zielwert der letzten Positionierung anzeigen



## RI Read Input

Syntax **RMT [m]**

- **ohne** Achsnummer *m* Schaltzustände aller Steuer-Eingänge  
IN 1 .. IN 12 anzeigen
- **mit** Achsnummer *m* Schaltzustand des Eingangs  
IN *m* anzeigen

**0** Eingang IN *m* = **L** 0V

**1** Eingang IN *m* = **H** +24V

## RMT Remote

Syntax **RMT [s]** card

Status der RS232-Schnittstellen

Eingabe

**s = 0** Offline

**s = 1** Online für Hostschnittstelle ( Rückseite )

**s = 2** Online für Service-Schnittstelle ( Front )

keine Schnittstelle hat Priorität

**ohne s** Remote-Status der Host-Schnittstelle anzeigen

**s = 0** Host-Schnittstelle **nicht** Online

**s = 1** die Host-Schnittstelle ist Online

der Status der Service-Schnittstelle kann nicht abgefragt werden

nach Systemstart **s = 0** Offline

**S Stop**

Syntax **S [m]** card

**mit** Achsnummer *m* Achse *m* stoppen

**ohne** Achsnummer *m* alle Achsen stoppen

Luftkissen- und Stromabschalt-Funktionen werden ebenfalls bedient

**SA Luftkissenfehler** card

Syntax **SA m [CLR]**

**SN System-Reset** card

Syntax **SN [CLR]**

**SP Positionierungen** card

Syntax **SP m [CLR]**

**SR Pos.-failures** card

Syntax **SR m [CLR]**

**ST Pos.-faults** card

Syntax **ST m [CLR]**

**SW Watchdog-Resets** card

Syntax **SW [CLR]**

**CLR** den entsprechenden Ereigniszähler der Achse *m* löschen

**ohne** aktuellen Zählerstand für die Achse *m* anzeigen

Default die Zählerstände bleiben erhalten

**SB Slow backward**

Syntax **SB *m* [*f*]** Hz card

**SF Slow forward**

Syntax **SF *m* [*f*]** Hz card

Achse *m* frei rück- bzw. vorwärts bewegen mit der aktuellen Start/Stop-Frequenz oder, wenn angegeben, der Frequenz *f* die Stepfrequenz wird direkt ein- und ausgeschaltet ( ohne Beschleunigungsrampe )

Der Fahrbereich ist begrenzt durch die Endschalter und die eingestellten Software-Limiten der Achse *m*

**f = 33 .. 10000** bewegen mit der Stepfrequenz *f*

**ohne f** Achse *m* mit der Stepfrequenz *f* bewegen

**SO Output schalten**

Syntax **SO *m* [*0|1*]**

direkte Steuerung des Outputs Out *m*

**0** Out *m* → **Off**

**1** Out *m* → **On**

**ohne** Schalt-Zustand von Out *m* anzeigen

nach Systemstart sind alle Outputs **Off**

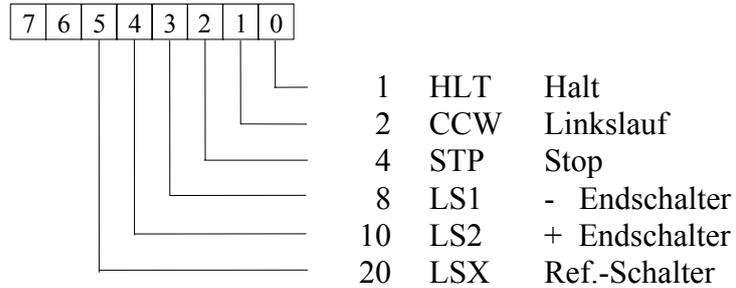
Default Out *m* = **0** Off

**SS show Status**

Syntax **SS** *m* hex

für die Achse *m* Statusregister des Motor-Controllers anzeigen

Motorcontroller-Statusregister

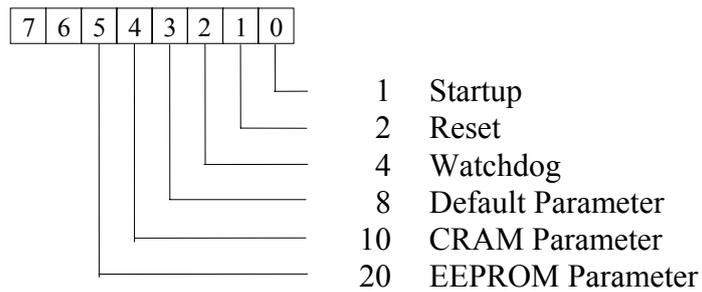


**SYR System-Register**

Syntax **SYR** hex

Systemregister der Steuerung anzeigen hex

System-Status Register

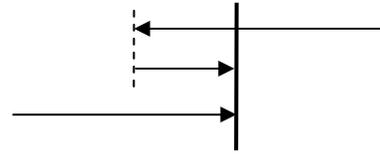


das Systemregister zeigt die Ursache des letzten Systemstarts und die Quelle des dabei geladenen Parametersatzes

**T**      **einseitig anfahren**      Nur im Closed Loop Betrieb !

Syntax      **T m [r]**      MS    int

die Zielposition der Achse *m* immer aus der durch **r** definierten Richtung anfahren  
 wenn die direkte Anfahrriechtung falsch ist, wird zuerst das Ziel zuerst überfahren und anschliessend in der verlangten Richtung angefahren



das Vorzeichen von **r** bestimmt die finale Anfahrriechtung, der Betrag von **r** den Overshoot in MS ( Motorsteps )

**r = +- 0 .. 10000**      MS    Motorsteps

**ohne r**      aktuellen Wert von **r** für die Achse *m* anzeigen

Default      **r = 0**      Anfahrriechtung nicht definiert

**U**      **Position**

Syntax      **U [m [r]]**      real

Istposition

- **mit m** und **r**      Definition eines neuen Positionswertes an der aktuellen Position für die Achse *m*
- **mit** Achsparameter **m**      Ist-Position der Achse *m* anzeigen
- **ohne** Parameterangabe      alle 12 Achspositionen *m* anzeigen

Default      **U = 0.000**      Positionswert der Achse *m* = 0

**UD**      **Position hex**

Syntax      **UD m**

Position der Achse *m* in Encodersteps **DS** als hex-Zahl anzeigen ( 6-stellige hex-Darstellung von 24 Bit bin., 2-er-Komplement pos. Werte ohne Vornullen )

## UU Position ohne Veränderung der H- und V-Parameterwerte

Syntax **UU** *m* [r] real

nach Eingabe eines neuen Positionswertes *r* behalten die Softwarelimiten und der Encodernullpunktwert der Achse *m* ihre Lage in Bezug zur Mechanik

*r* neue Ist-Position  
**ohne** *r* Position der Achse *m* anzeigen

## V Encoder-Nullpunkt

Syntax **V** *m* [r] DS int

Positionswert des Encodernullpunkts in Encodersteps

*r* DS Nullposition des Encoders in Encoder-Steps  
**ohne** *r* Nullposition des Encoders von Achse *m* anzeigen

Default *r* = **0** Istposition = 0.000 am Encodernullpunkt

## W Luftkissenlogik

Syntax **W** *m* [r] card

Definition der Luftkissensteuerung für die Achse *m* und der Verknüpfung mit einer weiteren Achse

*r* = **0** keine Luftkissensteuerung für die Achse *m*

*r* = *m* Luftkissensteuerung für die Achse *m*

*r* = *n* *n* = Achsnummer des in der Logikkette folgenden Motors  
 Luftkissensteuerung für die Achsen *m* und *n*

**ohne** *r* aktuellen Wert von *r* für die Achse *m* anzeigen

Default *r* = **0** keine Luftkissensteuerung für die Achse *m*

Die Luftkissensteuerung kann auch zur aut. Steuerung von Magnetbremsen eingesetzt werden

## **Z Encoderumfang**

Syntax **Z m [r]** DS card

Stepzahl pro  $360^0$  des der Achse *m* zugeordneten Encoders

erlaubt das Über- und Unterfahren des Nullpunkts eines Singleturn-Encoders  
( Pseudo-Multiturn )

r DS Anzahl Encodersteps pro Umdrehung des Encoders

**ohne r** aktuellen Wert von r für die Achse *m* anzeigen

Default r = 0 Funktion ausgeschaltet für Achse *m*

## **? ?C ?M Helpfunktionen**

Syntax ?  
?C  
?M

!!! nur im Terminalmode möglich ( ECHO = 1 )

? zeigt die beiden Detail-Befehle ?C und ?M

?C Befehlsliste anzeigen

?M Liste der System-Meldungen anzeigen

## Fehler- und Status-Meldungen

Meldungen, die im Terminalmodus ( **Echo 1** ) unmittelbar nach Achsbewegungen oder fehlerhaften Eingaben vom System auf der aktiven Schnittstelle ausgegeben werden.

Im Computermode ( **Echo 0** ) werden keine Statusmeldungen spontan ausgegeben und Fehler Meldungen Fehlermeldungen ohne Text.

Statusregister-Werte abfragen mit **MSR m**

Meldung		Fehler
?ADR		Address Error
?BSY		Motor Busy
?CMD		Command Error
?LOC		Local Mode
?PAR		Parameter Error
?RNG		Range Error

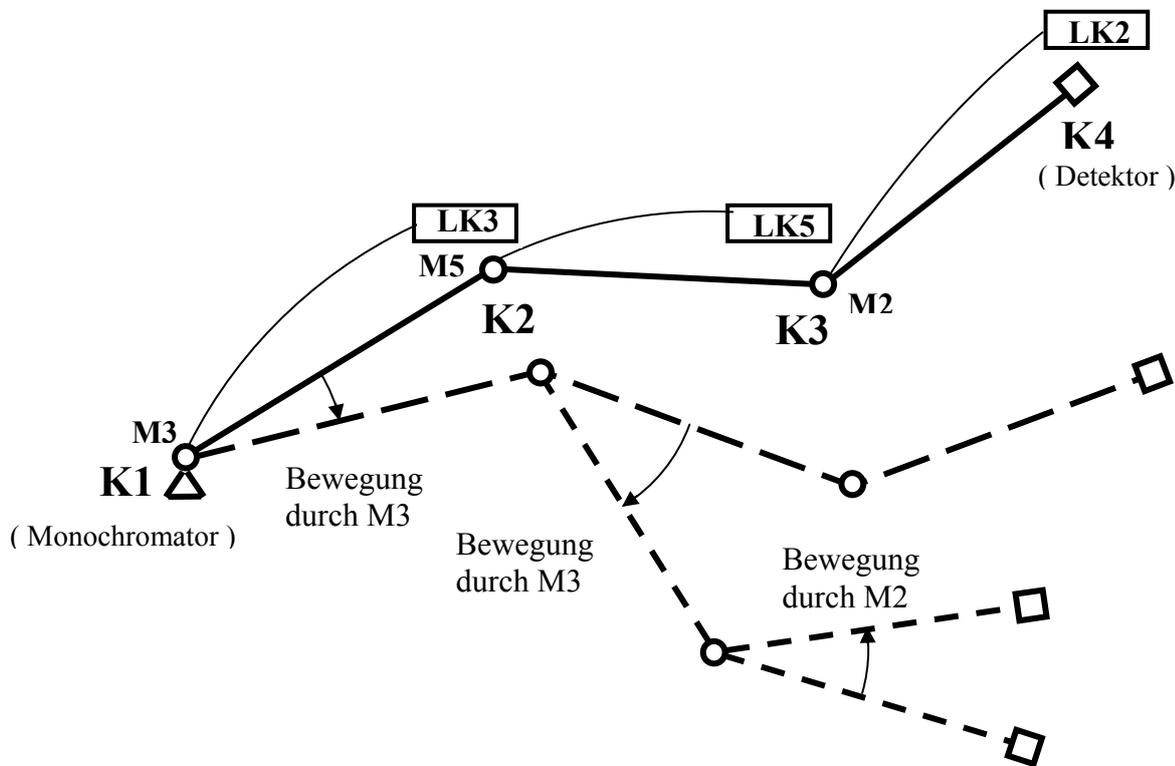
Meldung	Wert hex	Status
*MB	1	Motor Busy
*MP	2	Position Ready
*RP	4	ReferencePosition Ready
*MS	8	Stop on Switch or Command
*ML	10	Stop on lower Switch
*MU	20	Stop on upper Switch
*MH	40	Stop on HALT
*FR	80	Run Fault
*MR	100	Run Failure
*FP	200	Position Fault
*MF	400	Position Failure
*RF	800	Reference Failure
*ACE	1000	Aircushion Error
*NF	2000	Not free of Switch

# Anhang

## Luftkissensteuerung

**Beispiel** 3 Achsen-System mit dem ortsfesten Knoten K1 und den auf Luftkissen LK3, LK5 und LK2 gelagerten Gelenk-Knoten K2, K3, K4

Knoten	K1	K2	K3	K4
Motoren	M3	M5	M2	
Luftkissen		LK3	LK5	LK2
gesteuert durch Ausgänge		AC3	AC5	AC2
Rückmelde-Eingänge		ACI3	ACI5	ACI2



### Logikfunktion

Damit eine Achse sich bewegen kann, müssen alle in der Kette folgenden Knoten durch ihre Luftkissen angehoben werden ( z.B. müssen LK5 und LK2 heben, wenn M5 drehen soll )

Mit den Parametern **W m n** wird eine Logik konfiguriert, welche für jede Achse die betreffenden Luftkissen-Schaltausgänge aktiviert und die zugehörigen Rückmeldungen überwacht.

( *m* aktuelle, *n* nächstfolgende Motorachse in der Kette )

### Konfiguration

Knoten	K1 mit M3	<b>W 3 5</b>	
	K2 mit M5	<b>W 5 2</b>	
	K3 mit M2	<b>W 2 2</b>	Abschluss der Kette

## Magnetbremsensteuerung

Zur aut. Steuerung von Magnetbremsen wird die Luftkissenfunktion eingesetzt.

### Konfiguration

Für Achse  $m$      **W  $m$  m**

Wenn keine Rückmeldung überwacht werden muss, kann die Prüfung des Rückmedesignals abgeschaltet werden durch Setzen von Bit 3 ( Wert 8 ) im Optionregister der Achse  $m$ .

Für Achse  $m$      **Opt  $m$  8**

## Getriebefaktoren bestimmen

Die Drehbewegung eines Schrittmotor wird meistens über ein Getriebe in die Bewegung des Antriebs umgesetzt.

Als Getriebe werden kompakte Zahnrad-, oder Planetengetriebe, einzelne Zahnräder, Schneckengetriebe, Zahnriemengetriebe und lineare Antriebe mit Zahnriemen oder Spindeln verwendet ( auch in Kombination ).

Die Schrittauflösung des Antriebs wird dabei mit dem Übersetzungsverhältnis des Getriebes transformiert.

Auch der Messgeber wird meist über Getriebe ( Das Motorgetriebe selber, ein spezielles Messgetriebe oder beides kombiniert ) mit dem Antrieb gekoppelt.

Der Parameter **FM** für den Motor gibt die Anzahl Motorschritte an, welche notwendig ist um den Antrieb um ein Winkelgrad bzw. einen Millimeter zu bewegen.

Der Parameter **FD** für den Messgeber (Winkel- od. Linear-Geber) gibt die Anzahl der gemessenen Geberschritte pro Winkelgrad bzw. Millimeter Bewegung des Antriebs an.

### Syntax

**FM  $m$  [c] [n]**

$m = 1 ..12$      Motor-Achsnummer  
 c [Card]     Anzahl Motorschritte pro  
 n [Card]     Anzahl Winkelgrad | mm

Eine Änderung des Motor-Drehsinns muss an der Hardware ( Motor-Enstufe, durch Phasenleitertausch oder durch Jumper im IO-Interface vorgenommen werden ).

**FD  $m$  [c] [n]**

$m = 1 ..12$      Motor-Achsnummer  
 c [int]     Anzahl Encoderschritte pro  
 n [card]     Anzahl Winkelgrad | mm

Der Drehsinn des Gebers kann durch Ändern des Vorzeichens von c gewechselt werden

**Motor-Schrittauflösung  $k_m$**

2 Ph Halbschritt	<b>400 MS / 360<sup>0</sup></b>	<b>400 MS / U</b>
3 Ph einstellbar *	200 MS / 360 <sup>0</sup>	200 MS / U
	400 MS / 360 <sup>0</sup>	400 MS / U
	500 MS / 360 <sup>0</sup>	500 MS / U
	<b>1000 MS / 360<sup>0</sup></b>	<b>1000 MS / U</b>
5 Ph Vollschritt	500 MS / 360 <sup>0</sup>	500 MS / U
	<b>1000 MS / 360<sup>0</sup></b>	<b>1000 MS / U</b>

\* Bei 3 Ph Antrieben kann die eingestellte Auflösung durch die Betriebsart **Mikroschritt** verzehnfacht werden.

**Encoder-Schrittauflösung des  $k_a$**

Incrementalgeber	<b>100 DS / mm</b>	
	<b>1000 DS / mm</b>	
Absolutgeber Litton		
11 Bit und 17 Bit	<b>256 DS / 360<sup>0</sup></b>	<b>256 DS / U</b>
SSI Absolutgeber Heidenhain		
ROQ 425	<b>8192 DS / 360<sup>0</sup></b>	<b>8192 DS / U</b>
ROC 424	<b>4096 DS / 360<sup>0</sup></b>	<b>4096 DS / U</b>
ROQ 417	<b>131072 DS / 360<sup>0</sup></b>	<b>131072 DS / U</b>

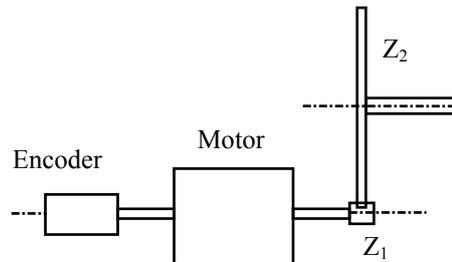
## Getriebe-Übersetzung

$$\text{Getriebeübersetzungsfaktor } i = \frac{\text{Anzahl Wellenumdrehungen Motor-/Encoderseite}}{\text{Anzahl Wellenumdrehungen Antriebseite}}$$

## Zahnrad-Übersetzung

$$\text{Getriebeübersetzungsfaktor } i = \frac{\text{Zähnezahl Antriebseite}}{\text{Zähnezahl Motor-/Encoderseite}} = \frac{Z_1}{Z_2}$$

Zähler und Nenner ganzzahlig



$$\text{Schrittauflösung Antrieb } k = k_m * i_1 * i_2 * \dots$$

$$\text{Schrittauflösung Messung } k = k_d * i_1 * i_2 * \dots$$

$k_m$	Schrittauflösung Motor	[ MS / 360 <sup>0</sup> ]
$k_d$	Schrittauflösung Encoder	[ DS / 360 <sup>0</sup> ]
$i_1, i_2 \dots$	Übersetzungen einzelner Getriebestufen	

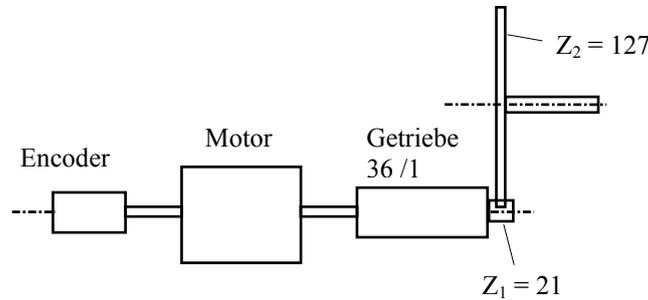
Die Übersetzung von Linearantrieben wird in Anzahl Umdrehungen der Antriebswelle pro Millimeter Translation angegeben. ( immer ganze Zahlen für Zähler und Nenner ! )

Die Übersetzung von Spindelantrieben ( Linearantriebe ) kann, wenn nicht bekannt, meist an der Spindelsteigung gemessen werden.

**Achtung** Spindelantriebe sind ev. mehrgängig !

**Beispiel**

Drehantrieb



Antrieb 2 Ph-Motor	$k_m = 400 \text{ MS} / 360^\circ$
Absolutgeber direkt an Motorwelle	$k_d = 256 \text{ DS} / 360^\circ$
Getriebe 36 / 1	$i_1 = 36 / 1$
2 Zahnräder 21z, 121z	$i_2 = 127 / 21$

Da der Encoder auf der Motorwelle sitzt, ist das Messgetriebe mit dem Getriebe des Antriebspfads identisch

$$\text{Auflösung Antrieb } k = \frac{400 \text{ MS} * 36 * 127}{3608 * 1 * 21} = \frac{40 \text{ MS} * 127}{218} = \frac{5080 \text{ MS}}{218}$$

Daraus folgt der Parameterwert für den Motor

**FM m 5080 21**

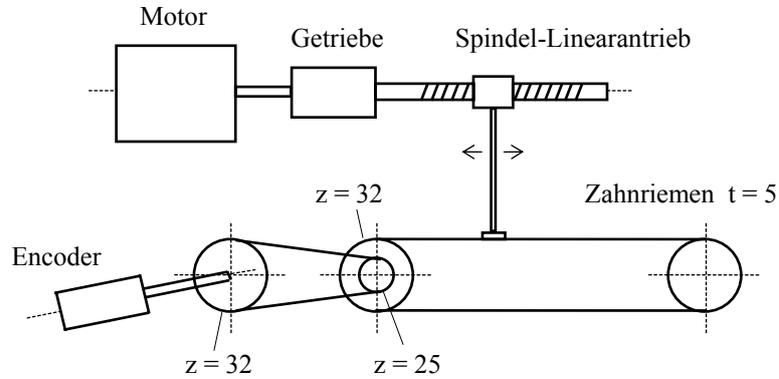
$$\text{Auflösung Messung } k = \frac{256 \text{ DS} * 36 * 127}{3608 * 1 * 21} = \frac{256 \text{ DS} * 127}{10 * 218} = \frac{32512 \text{ DS}}{2108}$$

Daraus folgt der Parameterwert für den Encoder

**FD m 32512 210 od. 16256 105**

**Beispiel**

Linearantrieb



Motor 5Ph Halbschritt	1000 MS / U
Getriebe	3 / 1
Spindel-Linearantrieb	10 mm / U
Messgeber	1024 DS / U
Messgetriebe Zahnriemen t = 5 mm	
Zahnscheibe 32 z	1U / 32 * 5 mm
Zahnriemengetriebe 32z, 25z	

$$\text{Auflösung Antrieb } k = \frac{1000 \text{ MS} * 3 * 1 \text{ U}}{1 \text{ U} * 1 * 10 \text{ mm}} = \frac{300 \text{ MS}}{1 \text{ mm}}$$

Daraus folgt der Parameterwert für den Motor

**FM m 300 1**

$$\text{Auflösung Messung } k = \frac{1024 \text{ DS} * 25 * 1}{1 \text{ U} * 32 * 32 * 5 \text{ mm}} = \frac{5 \text{ DS}}{1 \text{ mm}}$$

Daraus folgt der Parameterwert für den Encoder

**FD m 5 1**

## Referenzverfahren

Positioniersysteme, welche nicht mit Absolutmessgebern ausgerüstet sind, d. h. Systeme ohne Messgeber oder mit inkrementalen Dreh- oder Lineargebern, müssen ihr relatives Messsystem an einer Fixposition referenzieren. Es stehen dazu 6 verschiedene Verfahren zu Verfügung.

### Achtung

Für die Referenzfahrt müssen die entsprechenden End- bzw. Referenz-Schalter aus der Istposition gefahrlos angefahren werden können. Softwarelimiten werden ignoriert.

Der Referenzlaufmodus wird für jede Motorachse durch ihren Parameter K bestimmt.

**K** *m r*

**r**

- 0** kein Referenzlauf
- 1** mit Endschalter **LS1**
- 1** mit Endschalter **LS2**
- 2** mit separatem Referenzschalter an Eingang **IN**
- 11** mit Endschalter **LS1** und Indexpuls
- 11** mit Endschalter **LS2** und Indexpuls
- 12** mit separatem Referenzschalter an Eingang **IN** und Indexpuls

Für die Modi -11, 11, 12 ist ein Inkrementalgeber mit Indexausgang notwendig.

Die Modi 2 und 12 setzen einen aus dem Arbeitsbereich jederzeit anfahrbaren Referenzschalter voraus, welcher durch eine genügend lange Schiene derart betätigt wird, dass aus seinem Schaltzustand eindeutig die Fahrtrichtung zum Erreichen der Schaltposition aus der Istposition bestimmt werden kann, d.h., ein Schienenende muss ausserhalb des normalen Fahrbereichs liegen.

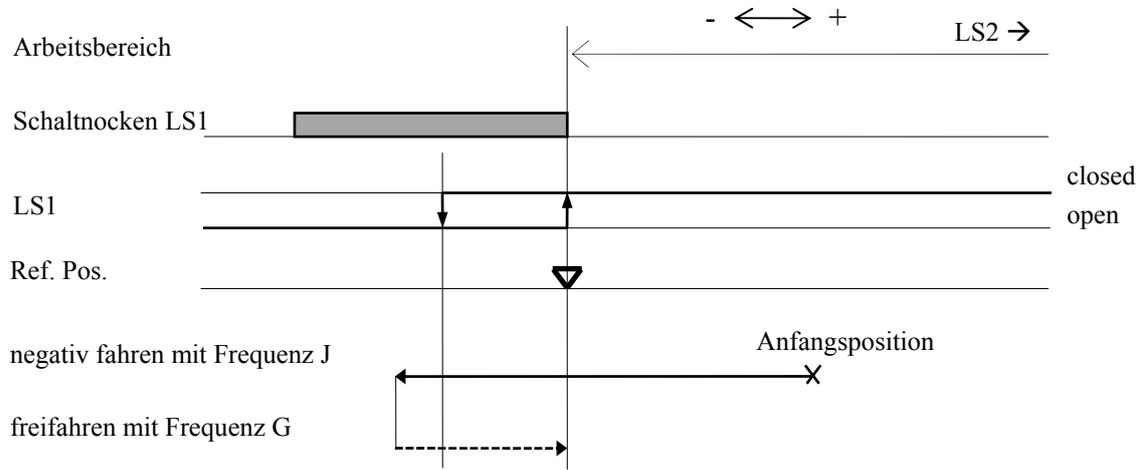
Die Fahrwege werden mit der Positionierfrequenz J zurückgelegt. Das Freifahren der Schalter und das Anfahren des Indexpunkts erfolgen mit Start-Stop-Frequenz G.

Für die Modi -11, 11, 12 kann zur Reduktion der Fahrzeit ein Referenzparameter Q bestimmt werden, damit ein Teil der Strecke vom End- bzw. Referenzschalter zum Indexpunkt mit Positionierfrequenz J zurückgelegt werden kann. Q soll so gewählt werden, dass die Reststrecke *r* bis zum Indexpunkt, die mit Start-Stop-Frequenz gefahren wird, in ca. 1s gefahren wird.

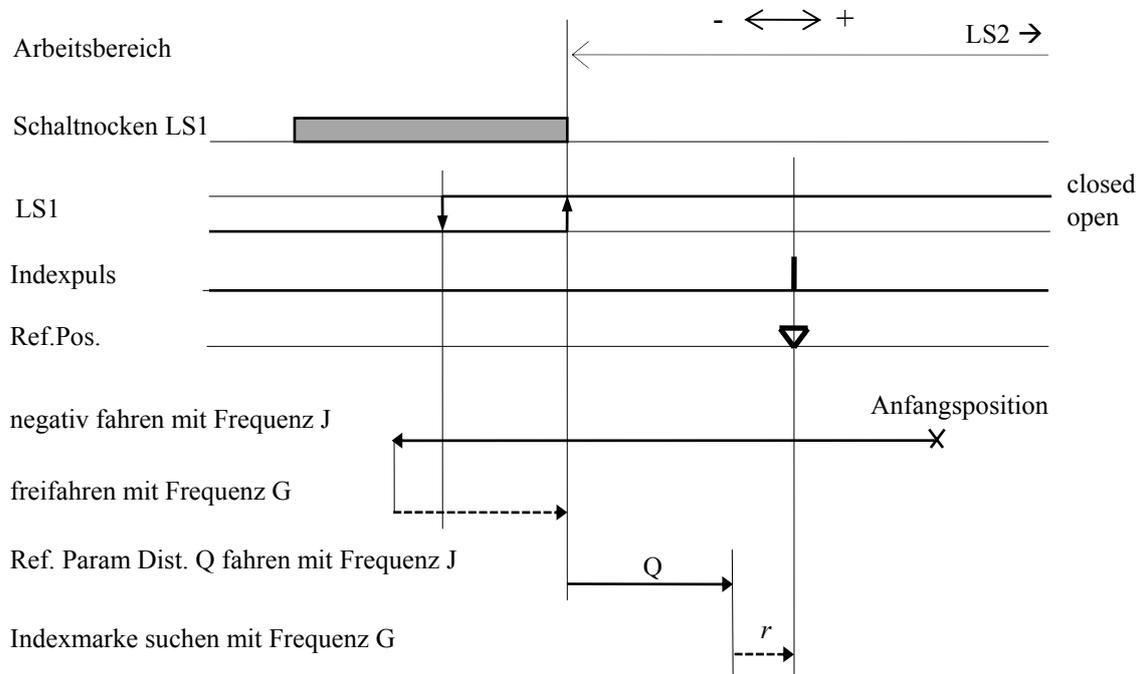
**Achtung !** Wenn die Index-Position infolge eines zu gross gewählten Q-Wertes überfahren wird, sucht das System nach der nächstfolgenden Index-Position ( Nächster Umgang des Encoders )

## Mode K

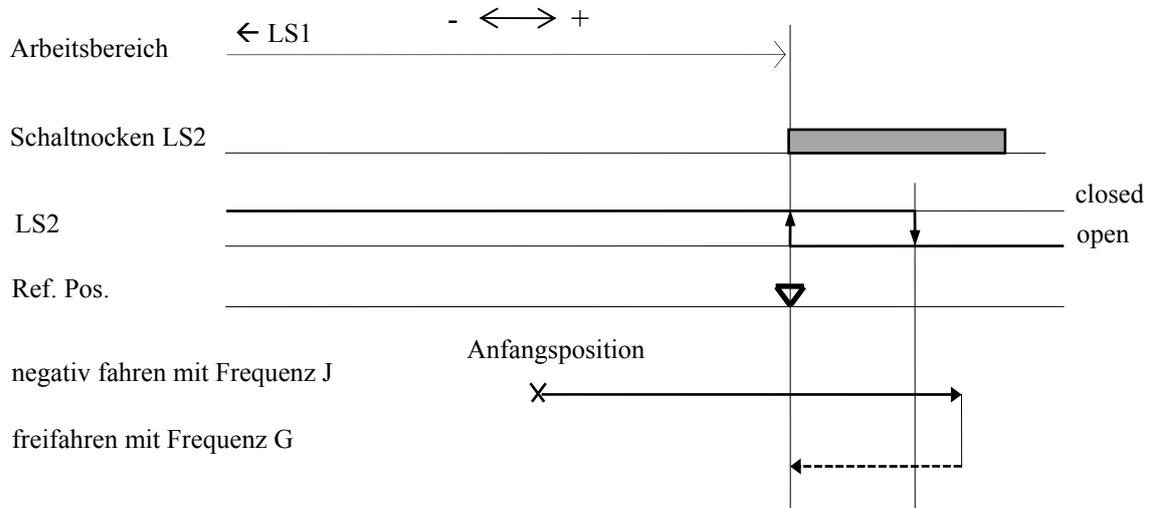
### -1 mit Endschalter LS1



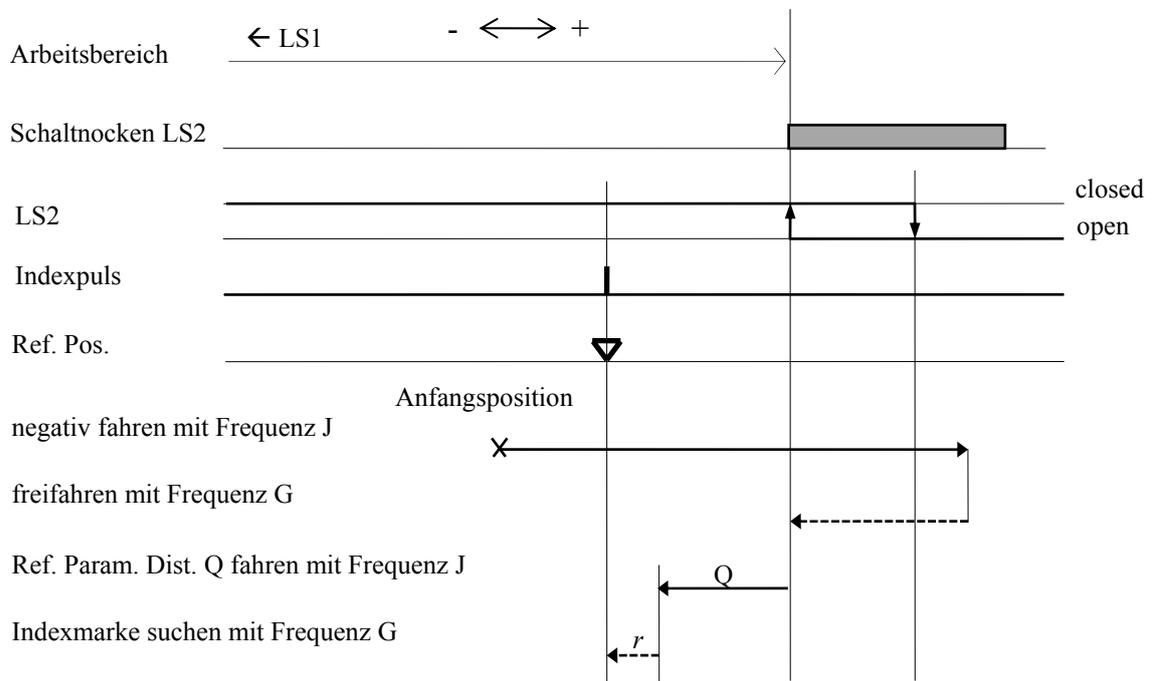
### -11 mit Endschalter LS1 und Indexpuls



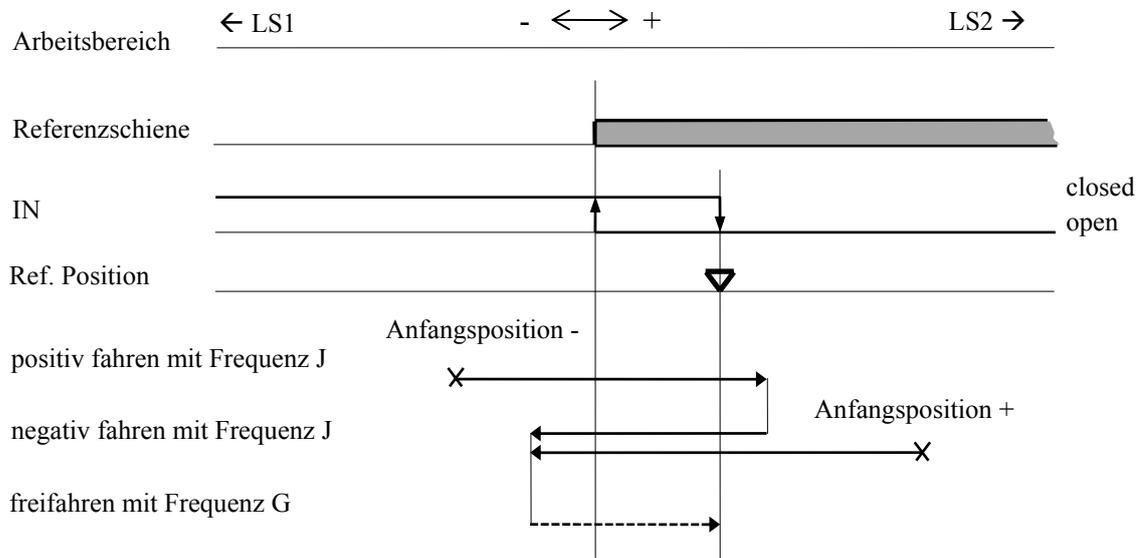
## 1 mit Endschalter LS2



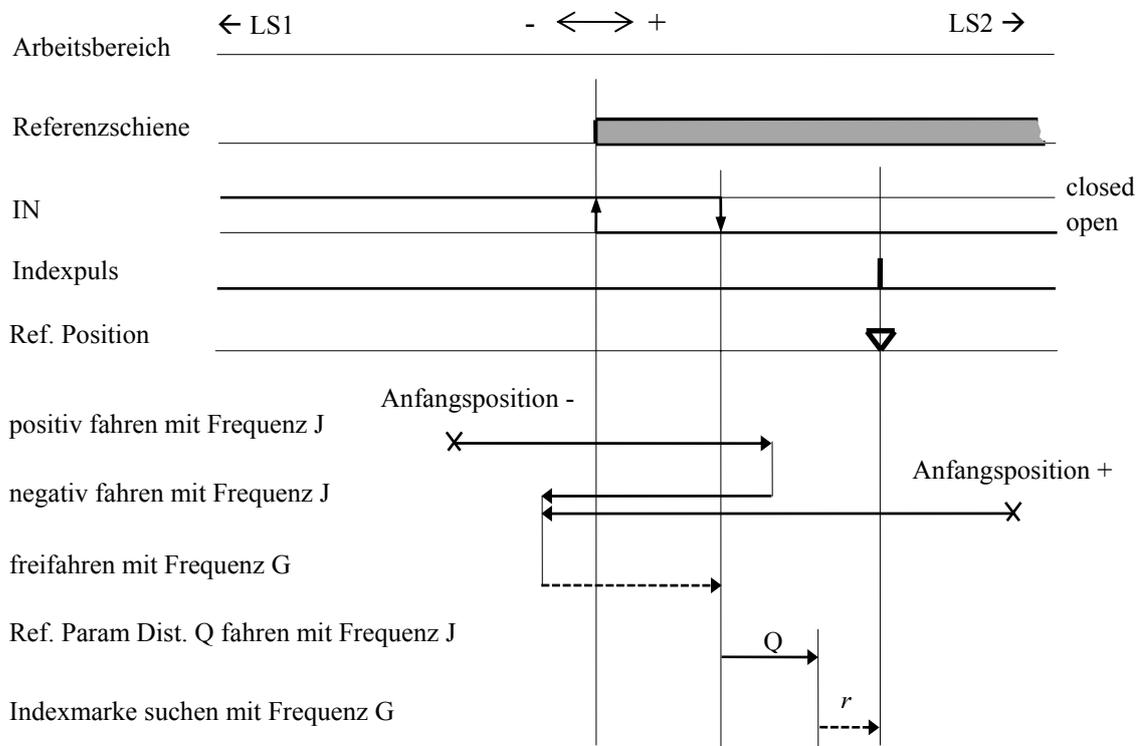
## 11 mit Endschalter LS2 und Indexpuls



## 2 mit Referenzschalter



## 12 mit Referenzschalter und Indexpuls



## Varianten und Spezialausführungen der Motorsteuerung

### μE1

Für μE1 wurde eine Variante der Standard-Kompaktsteuerung gebaut unter Einbezug von bereits vorhandenen Schrittmotoren, Endstufenkarten und Absolut-Winkelgebern ( 2 Stück )

- Die Verdrahtung der Messerleiste für die Endstufenkarte vom M1 ist an die RUBRAN-Karte angepasst
- Für den Anschluss der Endschalter ist ein zusätzlicher LEMO-Stecker vorhanden
- Die Encoder werden über ein spezielles Absolutgeber-Interface angeschlossen
- Das Encoder-Interface ist zusätzlich mit zwei Display-Anschlüssen ausgerüstet
- Für den Betrieb dieses speziellen Interfaces müssen EPROMs mit der Variantenversion **V3.2μ** eingesetzt werden

### Spezifikationen

#### Antrieb

Achse	Endstufenkarte	Motorstrom	Chassisverdrahtung
M1	RUB02.102 Rubran 2Ph	2A	speziell 2Ph
M2	D 920 Berger Laar 3 Ph standard	5.8A	standard 3Ph

#### Encodermodul

**AEC** Interface für Absolut-Encoder 76 GC-10-1-S-1 ( Litton ) und Externdisplay VMIO10-C

Piggyback	AEC Pgb μSR
Typencode	2
Kanalzahl	2
Grösse	3HE 4TE
Steckplatz	variabel

#### Encoder

Schnittstellen	10 Bit parallel, 5V-Logik
Codierung	Graycode
Encodertyp	Litton 76 GC-10-1-S-1
Kopplung	keine Potentialtrennung
Anschluss	2 x DSub 15p
Konfiguration	Moduladresse

Interface Nr	Moduladresse	Encoder
AEC 1	2400	1, 2

### Externdisplay

Schnittstellen 4 Dekaden parallel, 5V-Logik  
 Codierung BCD

### EPROM

Version **V3.2 $\mu$**

In dieser Softwarevariante wurden die Programm-Module **McMain** und **EncoderA** der Normalversion **V3.2** durch Spezialmodule mit gleichen Namen ersetzt

Beim Systemstart wird die Version **V3.2 $\mu$**  angezeigt.

Alle weiteren Funktionen sind mit der Normalversion **V3.2** identisch

### Anmerkung

Solange kein Absolutgeber-Interface eingesetzt wird, können beide Softwareversionen verwendet werden

## Externe Dokumente

Herstellerdokumente im externen Ordner **TM-04-05\_add**

**Anmerkung** Die mit # gekennzeichneten Positionen werden nicht mehr produziert

### VME Module

<a href="#"><u>VCPU02</u></a>	#	VME CPU-Modul
<a href="#"><u>VMIO10</u></a>	#	VME IO-Basismodul
<a href="#"><u>VMIO27/28</u></a>	#	INC Interface-Piggyback zu VMIO10
<a href="#"><u>SM_ABS</u></a>	#	SSI Interface-Piggyback zu SMOD_3
<a href="#"><u>Z8530</u></a>	#	SCC Serial Communication Adapter zu VCPU02
<a href="#"><u>Z8536</u></a>	#	CIA Complex Interface Adapter zu VCPU10 und VMIO10
<a href="#"><u>RTC2421</u></a>		RTC Realtime Clock zu VCPU02
<a href="#"><u>THCT12</u></a>		Incremental Interface THCT12024 zu VMIO27/28
<a href="#"><u>HSMC20CP</u></a>	#	Stepmotorcontroller DIL40 für Piggyback MOT alte Version
<a href="#"><u>HSMC20_8_P</u></a>		Stepmotorcontroller PLCC44 für Piggyback MOT

### Endstufen Module

<a href="#"><u>2Ph Karten</u></a>		2Ph Karte in 2 verschiedenen Bauarten
<a href="#"><u>3Ph D920</u></a>		3Ph Karte, moderneres 3Ph-System ersetzt 5Ph-System
<a href="#"><u>5Ph D550</u></a>	#	5Ph Modul, nicht für Neukonstruktionen
<a href="#"><u>TLD01x Twinline</u></a>		3Ph Modul für leistungsfähigere Antriebe, externe Montage