



isel-CNC-Betriebssystem 5.x

Programmieranleitung

Die in dieser Druckschrift enthaltenen Informationen, technischen Daten und Maßangaben entsprechen dem neuesten technischen Stand zum Zeitpunkt der Veröffentlichung. Etwa dennoch vorhandene Druckfehler und Irrtümer können jedoch nicht ausgeschlossen werden. Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir dankbar.

Es wird darauf hingewiesen, dass die in unseren Druckschriften verwendeten Soft- und Hardwarebezeichnungen der jeweiligen Firmen im allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Rechte vorbehalten. Kein Teil unserer Druckschriften darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der **isel Germany AG** reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Hersteller: **isel Germany AG**
Bürgermeister-Ebert-Straße 40
D-36124 Eichenzell

Tel.: (06659) 981-0
Fax: (06659) 981-776
Email: automation@isel.com
<http://www.isel.com>

Stand: 09/2008

Inhaltsverzeichnis

1	Einleitung	1
1.1	Symbolerklärungen und Sicherheitshinweise	2
2	Befehlsaufbau der DNC-Befehle	3
2.1	Grundbefehlssatz ab Prozessorkarte 4.0	4
2.1.1	Befehl: Achsenanzahl setzen	4
2.1.2	Befehl: Referenzfahrt	5
2.1.3	Befehl: Referenzgeschwindigkeit setzen	7
2.1.4	Befehl: Bewegung relativ	8
2.1.5	Befehl: Bewegung zur Position (move to)	10
2.1.6	Befehl: Positionsabfrage	12
2.1.7	Befehl: Nullpunktverschiebung	13
2.1.8	Befehl: Ebenenwahl	14
2.1.9	Befehl: Peek (Lesen von Speicheradressen)	16
2.1.10	Befehl: Poke (Beschreiben von Speicheradressen)	17
2.1.11	Befehl: Batterie-RAM löschen	18
2.1.12	Befehl: CR/LF setzen	19
2.1.13	Befehl: Gerätenummer setzen	20
2.1.14	Befehl: TRACE (Einzelschrittausführung)	21
2.1.15	Befehl: Selbsttest	22
2.2	Ergänzender Befehlsumfang der Interface-Karte 5.0	23
2.2.1	Befehl: 3D-Linearinterpolation	23
2.2.2	Befehl: Zirkularinterpolation	25
2.3	Ergänzender Befehlsumfang bei Interface-Karten mit E/A Erweiterung	31
2.3.1	Befehl: Externe Speicherung	31
2.3.2	Befehl: Ausgangsport setzen	32
2.3.3	Befehl: Eingangsport setzen	32
2.4	Ergänzender Befehlsumfang EP 1090	33
2.4.1	Befehl: Output-Modul	33
2.5	Ergänzender Befehlsumfang ab Interface-Karte Version AZ 1350/5	33
2.5.1	Befehl Magnetbremse	33
2.6	Kontroll- und Steuercodes	34
2.6.1	Befehl: Selbsttest	34
2.6.2	Befehl: STOP	35
2.6.3	Befehl: µP-Reset	36
2.6.4	Befehl: Break	36
3	Befehlsaufbau CNC-Befehle	37
3.1	Grundbefehlssatz ab Prozessorkarte 4.0	38
3.1.1	Befehl: INPUT	38
3.1.2	Befehl: Referenzfahrt	39
3.1.3	Befehl: Bewegung relativ	40
3.1.4	Befehl: Bewegung zur Position (move to)	41

3.1.5	Befehl: Nullpunktverschiebung	42
3.1.6	Befehl: Ebenenwahl	43
3.1.7	Befehl: Synchronisationszeichen senden	44
3.1.8	Befehl: Warte auf Synchronisationszeichen	46
3.1.9	Befehl: Schleife/Verzweigung.....	47
3.1.10	Befehl: Impulssteuerung.....	49
3.1.11	Befehl: Zeitverzögerung	50
3.1.12	Befehl: Bewegung bis Impuls	51
3.1.13	Befehl: Starte angeschlossene Interface-Karte.....	52
3.2	Ergänzender Befehlsumfang Interface-Karte 5.0	53
3.2.1	Befehl: 3D-Linearinterpolation	53
3.2.2	Befehl: Zirkularinterpolation.....	54
3.3	Ergänzender Befehlsumfang bei Interface-Karten mit E/A-Erweiterung	56
3.3.1	Befehl: Ausgangsport setzen.....	56
3.3.2	Befehl: Eingangsport setzen.....	58
3.4	Ergänzender Befehl in Verbindung mit einer Programmwahleinheit.....	59
3.4.1	Befehl: Tastaturabfrage.....	59
4	Fehlermeldungen	61
4.1	Fehlermeldungen <i>isel</i> -Prozesskarten.....	61
4.2	Fehlermeldungen PAL-PC	65

Abbildungsverzeichnis

Bild 1:	Verlauf einer Referenzfahrt (zum Maschinennullpunkt)	6
Bild 2:	Umschaltbare Interpolationsebenen	14

1 Einleitung

Die Beschreibung des CNC-Betriebssystems 5.x ist eine umfassende Dokumentation aller Befehle der **isel**-Prozessorkarten. Die beschriebenen Befehle gelten für folgende **isel**-Steuerungssysteme:

- **isel**-Interface-Karte (bis Softwarestand 5.x)
- **isel**-CNC-Controller C 116, C 142/1, C 116-4, C 142-4
- **isel**-CNC-Steuerung C 10C, C 10C-E/A
- **isel**-Integrierte Technologien IT 108, IT 116
- **isel**-Bearbeitungszentrum EP 1090
- **isel**-Bearbeitungszentrum EP 1090/4

Das CNC-Betriebssystem unterstützt die Positionierung von maximal drei Schrittmotor-Antriebsachsen. Zusätzlich zu den Positionierparametern verarbeitet das Betriebssystem unterschiedliche Steuer- und Kontrollfunktionen.

Durch die Zusammenfassung aller Steuerungssysteme (hier Prozessorkarte genannt) innerhalb eines Betriebssystems sind evtl. Einschränkungen in Bezug auf die Programmierung der einzelnen Geräte zu berücksichtigen. Diese sind in den jeweiligen Hardware-Beschreibungen vermerkt.

Die in der Beschreibung verwendeten Programmbeispiele sind auf das Maximal-System ausgelegt. Somit müssen Sie ggf. Anpassungen im Bereich der Positionierbefehle vornehmen.

Der Begriff PAL-PC wird sowohl im Zusammenhang mit der Programmiersprache PAL-PC als auch mit dem Software-Ankopplungsmodul PAL-EP verwendet.

Zur direkten Programmierung der Prozessorkarten steht Ihnen ein festgelegtes Übertragungsformat zur Verfügung. Als Beispiel ist in dieser Dokumentation eine Programmierung in der Programmiersprache BASIC aufgelistet.

1.1 Symbolerklärungen und Sicherheitshinweise



Achtung

Dieses Symbol weist Sie darauf hin, dass Gefahr für Leben und Gesundheit für Personen besteht.



Gefahr

Dieses Symbol weist Sie darauf hin, dass Gefahr für Material, Maschine und Umwelt besteht.



Information

Dieses Symbol kennzeichnet wichtige Informationen.

2 Befehlsaufbau der DNC-Befehle

Im DNC-Modus werden die von einem Steuerrechner übergebenen Datensätze bzw. Befehle direkt ausgewertet und ausgeführt. Dafür ist zu Beginn der Datenkommunikation eine sogenannte Initialisierung notwendig. Sie besteht aus dem Dateneröffnungszeichen @, der Gerätenummer (Standard = 0) und der Anzahl der zu verfahrenen Achsen.

Anschließend werden der Prozessorkarte die Programmschritte einzeln übergeben und von ihr direkt ausgeführt.

Zur Überprüfung der Datenübertragung bzw. Meldung von aufgetretenen Fehlern werden über die Schnittstelle entsprechende ASCII-Zeichen an den Steuerrechner zurückgesendet. Dieses sogenannte Software-Handshake-Verfahren kann zu zwei unterschiedlichen Zeitpunkten realisiert werden:

1. Die Prozessorkarte setzt direkt nach Empfang des abzuarbeitenden Datensatzes das Quittierungs-/Fehlerzeichen ab.
2. Die Prozessorkarte arbeitet den übersendeten Befehlssatz ab, und meldet anschließend das Quittierungs-/Fehlerzeichen zurück.

Der gewünschten Modus wird durch Groß-/Kleinschreibung des Befehlszeichens unterschieden. Großbuchstaben führen zur Rückmeldung nach Abarbeitung der Befehle, Kleinbuchstaben zur direkten Rückmeldung.

Im folgenden wird der Befehlsumfang der Interface-Karte 4.0 beschrieben. Ergänzungen durch erweiterte Hardware (z. B. Interface-Karte 5.0) befinden sich am Ende des Kapitels.

Der in den Beispiel-Programmen benannte Terminalmode ist eine Funktion der **isel**-Software PAL-PC. Er wird im PAL-PC durch die Funktionstaste F2 eingeschaltet und stellt eine direkte Verbindung zwischen Bildschirm und Interface-Karte her.



Weitere Informationen hierzu finden Sie in der Anleitung zum PAL-PC, Kapitel X1 Kommunikationsfenster.

2.1 Grundbefehlssatz ab Prozessorkarte 4.0

2.1.1 Befehl: Achsenanzahl setzen

Zweck Durch Übergeben der Achsenanzahl wird die Prozessorkarte neu initialisiert. Dabei wird der Datenspeicher gelöscht und zur Speicheroptimierung entsprechend der Anzahl der Achsen neu eingeteilt.

Aufbau <GN><Achsen>
<GN> = Gerätenummer, Standard = 0
<Achsen> = Achsenangabe, s. u.

Anwendung @07

Erläuterung Die Karte wird durch @0 adressiert; der anschließende Zahlenwert beinhaltet die Achsenkonfiguration.

Achsenangabe	Wert
x	1
xy	3
xz	5
xyz	7

Beschränkung Nicht zulässig sind die Kombinationen @00, @02, @04, @06, sowie @08 und @09.



Programmierbeispiel

PAL-PC

GW-BASIC

#axis xyz;

```
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@07":gosub 1000
120 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print Karte meldet Fehler : ;a$
1030 stop
```



Der Befehl *Achsenanzahl setzen* löscht alle im RAM vorhandenen Daten, auch wenn durch die integrierte Option *Memory Back-up* die Daten nach Wegfall der Versorgungsspannung im RAM-Speicher der Prozessorkarte gespeichert waren.

2.1.2 Befehl: Referenzfahrt

Zweck	Die Prozessorkarte verfährt alle angegebenen Achsen an ihre Nullpunkte (Referenzpunkte). Die Referenzpunkte der Achsen liegen bei <i>isel</i> - Systemen an der dem Motor zugewandten Seite; positive Bewegungen finden in der Richtung vom Motor weg statt.								
Aufbau	@<GN>R<Achsen> <GN> = Gerätenummer, Standard = 0 <Achsen> = ganzzahliger Wert zwischen 1 und 7								
Anwendung	@0R7 oder @0r7								
Erläuterung	<p>Die Karte wird durch @0 adressiert. R gibt an, dass eine Referenzfahrt ausgeführt werden soll. Der anschließende Zahlenwert definiert die Achsen, die eine Referenzfahrt durchführen sollen:</p> <table><tr><td>x= 1</td><td>xy= 3</td></tr><tr><td>y= 2</td><td>yz= 6</td></tr><tr><td>z= 4</td><td>xyz= 7</td></tr><tr><td>xz=5</td><td></td></tr></table> <p>Die Reihenfolge der Ausführung ist dabei wie folgt festgelegt:</p> <p>—> Z-Achse —> Y-Achse —> X-Achse</p> <p>Dies gilt auch, wenn mit dem Ebenenbefehl eine andere Achse als Werkzeugachse bestimmt wurde. Hier kann ggf. die getrennte Referenzfahrt der Achsen Kollisionen mit dem Werkstück verhindern.</p> <p>Nach erfolgter Referenzfahrt sendet die Prozessorkarte ihr Quittierungszeichen und wartet auf die nächsten Befehle. Besteht die Notwendigkeit einer sofortigen Rückmeldung, müssen Sie statt R ein r verwenden. Die Prozessorkarte kann jedoch in jedem Fall erst wieder Befehle verarbeiten, nachdem die Referenzfahrt durch die Mechanik ausgeführt worden ist.</p>	x= 1	xy= 3	y= 2	yz= 6	z= 4	xyz= 7	xz=5	
x= 1	xy= 3								
y= 2	yz= 6								
z= 4	xyz= 7								
xz=5									
Beschränkung	<p>Den Befehl können Sie verwenden, nachdem eine Initialisierung der Prozessorkarte durch den Befehl <i>Achsenanzahl setzen</i> erfolgt ist und beschränkt sich auf die dort vorgegebene Achsenkonfiguration.</p> <p>Bei falscher Achsangabe erfolgt die Fehlerrückmeldung 3.</p> <p>Befindet sich die Karte im 3D-Modus, schaltet der Befehl wieder in die 2,5D-Ausführung zurück.</p>								



Programmierbeispiel

PAL-PC

#axis xyz;
reference xyz;

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"ast#1
110 print#1,"@07":gosub 1000
120 print#1,"@0R7":gosub 1000
130 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler:";a$
1030 stop
```



Bei nicht angeschlossenem Referenzschalter gibt die Prozessorkarte permanent Impulse aus. Durch zweimaliges Betätigen des Stopp-Tasters besteht jedoch die Möglichkeit, die Referenzfahrt der entsprechenden Achse abubrechen.

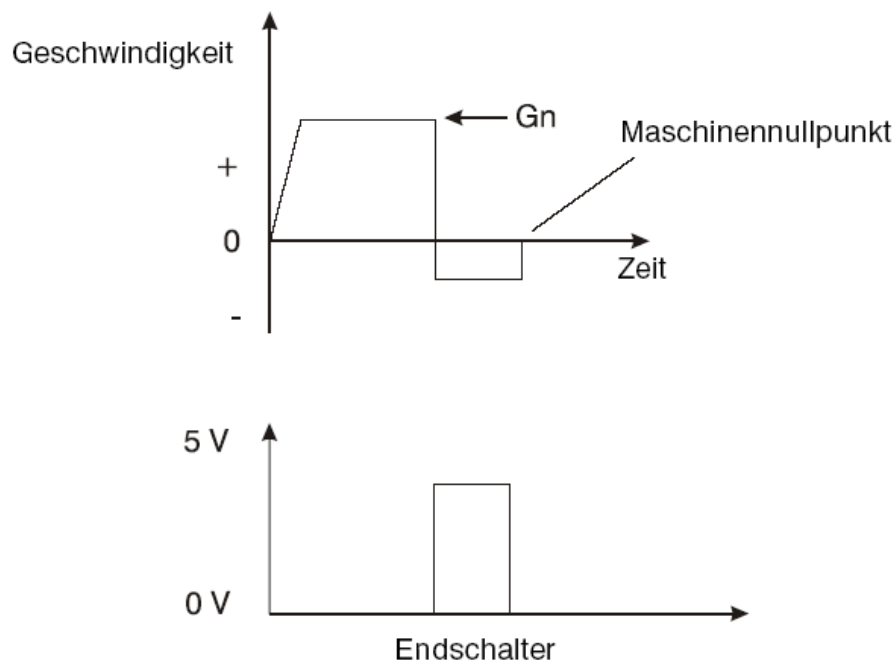


Bild 1: Verlauf einer Referenzfahrt (zum Maschinennullpunkt)

2.1.3 Befehl: Referenzgeschwindigkeit setzen

Zweck	Der Befehl definiert - getrennt für jede Achse - die Geschwindigkeit, mit der eine Referenzfahrt ausgeführt wird. Dabei handelt es sich nur um die Geschwindigkeit, mit der die Achse in negativer Richtung auf den Motor zuläuft; die Geschwindigkeit aus dem Schalter heraus lässt sich nicht beeinflussen (vgl. Referenzfahrt).	
Aufbau	@<GN>d<Gx> @<GN>d<Gx>, <Gy> @<GN>d<Gx>, <Gz> @<GN>d<Gx>, <Gy>, <Gz> <GN> <Gx> <Gy> <Gz> <Gx>, <Gy>, <Gz>	(x) (X-y) (X-z) (X-Y-z) = Gerätenummer, Standard = 0 = Referenzgeschwindigkeit x = Referenzgeschwindigkeit y = Referenzgeschwindigkeit z = ganze Zahl zwischen 30 und 10 000 Hz
Anwendung	@0d2500 @0d2400,3000 @0d1000,3000,9000	(1 Achse) (2 Achsen) (3 Achsen)
Erläuterung	Werden der Prozessorkarte keine Informationen zur Referenzgeschwindigkeit übergeben, erfolgt die Ausführung mit einem Default-Wert von 2 000 Schritten/s. Ein geänderter Wert bleibt beim Ausschalten erhalten, falls die Option <i>Memory-Back-up</i> eingebaut ist.	
Beschränkung	-	



Programmierbeispiel

PAL-PC

#axis xy;
#ref_speed 3000,5000;

GW-BASIC

```
100open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@03":gosub 1000
120 print#1,"@0d3000,5000":gosub 1000
130 print#1,"@0R3":gosub 1000
140 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler: ";a$
1030 stop
```



Eine zu hoch gewählte Referenzgeschwindigkeit kann in Verbindung mit einer großen Spindelsteigung durch die vorhandene Massenträgheit zu einer Beschädigung der Referenzschalter führen.

Die Prozessorkarte benötigt eine Schalthysterese des angeschlossenen Nullagenschalters (bei Anschluss von elektronischen Nullsensoren zu beachten).

2.1.4 Befehl: Bewegung relativ

Zweck	Die Prozessorkarte gibt entsprechend der übergebenen Schrittzahl und Schrittgeschwindigkeit für jede Leistungsstufe eine Impulskette aus. Die Verfahrbewegung wird sofort ausgeführt oder gespeichert.	
Aufbau	<p>@<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2></p> <p><GN> = Gerätenummer, Standard = 0 <Sx> = Schrittzahl x, Wert zwischen 0 und +/- 8 388 607 <Gx> = Geschwindigkeit x, Wert zwischen 30 und 10 000 • • <Gz2> = Geschwindigkeit Z-Achse (2. Bewegung)</p>	
Anwendung	<p>@0A 5000,900 (nur X-Achse) @0A 50,900,20,9000 (X- und Y-Achse) @0A 50,900,20,900,-20,900 (X- und Z-Achse) @0A 30,800,10,900,4,90,-4,30 (X-, Y- und Z-Achse)</p>	
Erläuterung	<p>Die Prozessorkarte wird durch @0 adressiert; A gibt an, dass eine Bewegung erfolgen soll. Die Prozessorkarte erwartet nun für jede Achse ein Zahlenpaar bestehend aus Schrittzahl und Geschwindigkeit.</p> <p>Die Bewegung erfolgt im Relativmaß, d. h. bezogen auf die letzte Position. Die Anzahl der Angaben muss mit der Achsenzahl übereinstimmen, d. h. ein Parameterpaar bei X-Betrieb, zwei Parameterpaare bei XY-Betrieb, drei Parameterpaare bei XZ-Betrieb und vier Parameterpaare für XYZ-Betrieb. Die einzelnen Zahlen müssen durch Kommata getrennt sein. Für die Z-Achse werden zwei Zahlenpaare erwartet, da für Bearbeitungsanwendungen sehr häufig die Situation "Fahren, Werkzeug absenken und anschließend anheben" vorkommt. Im 2,5D-Interpolationsbetrieb werden zuerst die Bewegungen der X und Y-Achse verfahren (linear interpoliert), anschließend wird die Z-Achse zuerst um die in z1 angegebenen und dann um die in z2 angegebenen Werte verfahren. Diese Interpolationszuordnung lässt sich bei 2D-Betrieb mit dem Ebenenbefehl ändern.</p> <p>Besteht die Notwendigkeit, nur eine Achse zu bewegen, sind dennoch für alle initialisierten Achsen Werte zu übertragen. Dabei ist für die Schrittzahlen der nicht bewegten Achsen 0 und für die Geschwindigkeit ein Wert zwischen 30 und 10 000 anzugeben. Die Prozessorkarte meldet sich nach erfolgter Ausführung mit dem Handshake-Character (0). Durch den unterschiedlichen Befehlscode a und A können Sie zwischen einer Quittierungsmeldung direkt nach der Übertragung und einer Rückmeldung nach Ausführungsende wählen. Die Prozessorkarte kann jedoch in jedem Fall erst wieder Befehle verarbeiten, nachdem die Ausführung beendet ist.</p>	

Beschränkung

Den Befehl können Sie erst verwenden, nachdem die Achsenanzahl gesetzt worden ist. Die Prozessorkarte prüft nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verlässt.



Programmierbeispiel

PAL-PC

```
#axis xy;  
move50(500),300(900);
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as#1  
110 print#1,"@03":gosub 1000  
120 print#1,"@0A50,500,300,900":gosub 1000  
130 print#1,"@0A20,200,-30,900":gosub 1000  
140 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



Im 2,5D-Interpolationsbetrieb wird die Geschwindigkeitsangabe der Achse mit dem längsten Weg als Bahngeschwindigkeit übernommen, die Geschwindigkeit der anderen Achse entsprechend dem Wegeverhältnis angepasst.
Im Gegensatz dazu wird im 3D-Interpolationsbetrieb die Geschwindigkeitsangabe der X-Achse als Vorgabewert für die Bahngeschwindigkeit herangezogen.

2.1.5 Befehl: Bewegung zur Position (move to)

Zweck	Die Prozessorkarte fährt mit den angegebenen Geschwindigkeiten an die angegebene Position. Die Verfahrbewegung wird sofort ausgeführt.	
Aufbau	@<GN>M<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2>	
	<GN>	= Gerätenummer, Standard = 0
	<Sx>	= Weginformation X-Achse
	<Gx>	= Geschwindigkeit X-Achse
	•	
	•	
	<Sz2>	= bei Absolut-Positionierung = 0
	<Gz2>	= Geschwindigkeit Z-Achse (2. Bewegung)
Anwendung	@0M 5000,900	(X-Achse)
	@0M 50,900,20,9000	(X- und Y-Achse)
	@0M 50,900,20,900,0,21	(X- und Z-Achse)
	@0M 30,800,10,900,4,90,0,21	(X-, Y- und Z-Achse)
Erläuterung	Die Prozessorkarte wird durch @0 adressiert. M gibt an, dass eine Absolut-Position folgt. Aus Kompatibilitätsgründen zum relativen Positionierbefehl werden auch hier für die Z-Achse zwei Zahlenpaare erwartet. Die zweite Positionsangabe der Z-Position muss jedoch Null sein und wird ignoriert. Die Prozessorkarte meldet sich nach erfolgter Ausführung mit dem Handshake-Character. Soll die Rückmeldung sofort erfolgen, müssen Sie m statt M verwenden. Die Prozessorkarte kann jedoch in jedem Fall erst wieder Befehle empfangen, nachdem die Ausführung beendet ist.	
Beschränkung	Den Befehl können Sie erst verwenden, nachdem die Achsenanzahl gesetzt worden ist. Der Befehl kann nicht während der Bearbeitung gespeicherter Befehle übersandt werden. Die Prozessorkarte prüft nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verlässt. Wollen Sie den Befehl speichern, müssen Sie die Prozessorkarte vorher in den Input-Modus setzen (vgl. Input) und den Befehlscode m benutzen.	



Programmierbeispiel

PAL-PC

```
#axis xy;  
reference xy;  
moveto 50(500),300(900);  
moveto 20(200),30(900);  
moveto 0(21),00(2000);  
stop.
```

GW-BASIC

```
100 open"com1.9600,N,8,1,DS,CD"as #1  
110 print#1,"@03":gosub 1000  
120 print#1,"@0M50,500,300,900"  
125 gosub 1000  
130 print#1,"@0M20,200,30,900"  
135 gosub 1000  
140 print#1,"@0M 0,21,700,2000"  
145 gosub 1000  
150 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

2.1.6 Befehl: Positionsabfrage

Zweck Die Prozessorkarte gibt die momentane Sollposition aller Achsen an den übergeordneten Rechner zurück.

Aufbau @<GN>P
<GN> = Gerätenummer, Standard = 0

Anwendung @0P

Erläuterung Die Prozessorkarte wird durch @0 adressiert. P gibt an, dass eine Positionsanfrage erfolgt. Die Prozessorkarte bestätigt dies mit dem Handshake-Character und gibt anschließend im hexadezimalen Format die Positionswerte aller Achsen aus (insgesamt 19 Byte = 18 hexadezimale Ziffern + 1 x Handshake)
Der Aufbau der rückgemeldeten Position ist folgendermaßen:

0	<u>000010</u>	<u>002000</u>	<u>FFFFFFE</u>
	A	B	C

A	Position x, hexadezimal im 2er-Komplement, im Beispiel der Wert 16 dezimal.
B	Position y, hexadezimal im 2er-Komplement, im Beispiel der Wert 8096 dezimal.
C	Position z, hexadezimal im 2er-Komplement, im Beispiel der Wert - 2 dezimal.

Beschränkung Der Befehl kann nur verwendet werden, wenn keine Verfahrbewegung stattfindet (wenn sich die Anlage im Stopp-Zustand befindet).
Der Befehl kann nicht während der Bearbeitung gespeicherter Befehle übersandt werden.
Die Prozessorkarte kann nicht prüfen, ob die Sollposition der aktuellen Position der Mechanik entspricht, da kein Regelkreis vorhanden ist.



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

@0P

-



Es werden immer die Positionen aller drei Achsen durch die Funktion zurückgegeben, unabhängig von der Anzahl der definierten Achsen. Die Interface-Karte sendet die entsprechenden ASCII-Zeichen mit der eingestellten Übertragungsgeschwindigkeit ohne, mittels Hardware-Handshake, eine Bestätigung des empfangenden Rechners zu erwarten.

2.1.7 Befehl: Nullpunktverschiebung

Zweck	Die Prozessorkarte speichert die momentane Position als virtuellen Nullpunkt für die angegebene(n) Achse(n). Die nächsten <i>Verfahre absolut</i> -Anweisungen berücksichtigen diesen virtuellen Nullpunkt als neuen Bezugspunkt																						
Aufbau	<div>@<GN>n<Achsen></div> <div><GN> = Gerätenummer, Standard = 0</div> <div><Achsen> = ganzzahliger Wert zwischen 1 und 7</div>																						
Anwendung	@0n7 • •	@0n1 • •																					
Erläuterung	<p>Die Karte wird durch @0 adressiert. n gibt an, dass eine Nullpunktverschiebung vorgenommen werden soll. Nach dem Befehl werden dem Rechner die Achsen mitgeteilt, für die eine Nullpunktverschiebung durchgeführt werden soll. Hierbei gilt die Zuordnung x = 1, y = 2, z = 4.</p> <p>Soll für mehrere Achsen eine Nullpunktverschiebung durchgeführt werden, müssen obige Werte addiert werden:</p> <table><thead><tr><th>Achsen</th><th>Wert</th><th>Achsen</th><th>Wert</th></tr></thead><tbody><tr><td>x</td><td>1</td><td>xy</td><td>5</td></tr><tr><td>y</td><td>2</td><td>yz</td><td>6</td></tr><tr><td>z</td><td>4</td><td>xyz</td><td>7</td></tr><tr><td>xy</td><td></td><td>3</td><td></td></tr></tbody></table> <p>Der Rechner meldet sich nach erfolgter Ausführung mit der Rückmeldung (vgl. Software-Handshake).</p>			Achsen	Wert	Achsen	Wert	x	1	xy	5	y	2	yz	6	z	4	xyz	7	xy		3	
Achsen	Wert	Achsen	Wert																				
x	1	xy	5																				
y	2	yz	6																				
z	4	xyz	7																				
xy		3																					
Beschränkung	Der virtuelle Nullpunkt hat nur für den Befehl <i>Verfahre absolut</i> eine Bedeutung. Relativpositionierung wird vom virtuellen Nullpunkt nicht beeinflusst, da hier ein Fahrvektor angegeben wird.																						



Programmierbeispiel

PAL-PC

```
#axis xy;
#elev 4,4;
moveto 80(900),8(900);
null xy;
moveto 2(900),4(990);
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@03":gosub 1000
120 print#1,"@0r3":gosub 1000
130 print#1,"@A1000,2000,2000,2000":gosub 1000
140 print#1,"@0n3":gosub 1000
150 print#1,"@M100,2000,100,2000":gosub 1000
160 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler : ";a$
1030 stop
```



Der virtuelle Nullpunkt wird durch Referenzfahrt wieder in den Anlagennullpunkt gelegt.

2.1.8 Befehl: **Ebenenwahl**

Zweck

2,5D interpolierende Prozessorkarten (z. B. Interface-Karte 4.0) können nur zwei von drei Achsen interpolieren. Im Einschaltzustand sind dies die X- und Y-Achse. Durch den *Ebenenwahl*-Befehl besteht jedoch die Möglichkeit, jede andere Ebenenkonfiguration als Hauptebene zu definieren. Die übrigbleibende dritte Achse wird als Werkzeugachse aufgefasst und anschließend, also nach der Positionierung der Hauptachsen verfahren.

Aufbau

@<GN>e<Ebene>
 <GN> = Gerätenummer, Standard = 0
 <Ebene> = Zahl zwischen 0 und 2
 0 = xy
 1 = xz
 2 = yz

Anwendung

@0e1 Schalte auf xz-Interpolation
 @0e0 Schalte auf xy-Interpolation

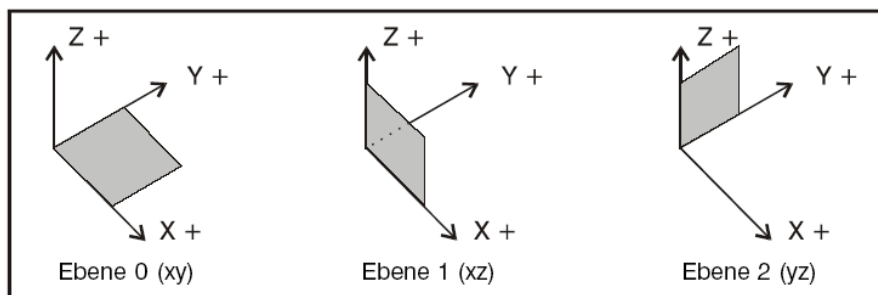


Bild 2: Umschaltbare Interpolationsebenen

Erläuterung

Um hohe Geschwindigkeiten zu erzielen (eine maximale Achsengeschwindigkeit von 10 kHz entspricht bei einer 5 mm-Spindelsteigung und einem 45° Grad Vektor einer Vektorgeschwindigkeit von 175 mm/s im Halbschrittbetrieb), kann die Prozessorkarte innerhalb dieser Zeit nur die Geschwindigkeitsverhältnisse von zwei Achsen gegeneinander berechnen. Der Ebenenbefehl erlaubt eine Umschaltung zwischen den Interpolationsebenen ohne Geschwindigkeitsverlust.

Beschränkung

Falls eine andere Interpolationsebene als XY gewählt wird, sollte für die zweite Bewegung der Bearbeitungsachse (die nicht interpolierte Achse) Null als Schrittzahl übergeben werden.



Programmierbeispiel

PAL-PC

```
#axis xyz;  
line yz;  
move 20(1000),30(1000),  
      33(1000),0(30);  
stop.
```

GW-BASIC

```
100 open"com1.9600,N,8,1,DS,CD"as #1  
110 print#1,"@07":gosub 1000  
120 print#1,"@0e2"  
125 gosub 1000  
130 print#1,"@0M20,200,30,900,33,900, 0,21"  
135 gosub 1000  
140 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



Im obigen Beispiel werden die Y- und Z-Achsen interpoliert (geradlinig zum Zielpunkt verfahren), die X-Achse nachgeführt.

Die Reihenfolge der Referenzfahrt wird durch den Ebenenwahlbefehl nicht beeinflusst.

Falls eine geänderte Referenzfahrtreihenfolge benötigt wird, dürfen keine Referenzbefehle mit Achsenkombinationen übergeben werden.

2.1.9 Befehl: Peek (Lesen von Speicheradressen)

Zweck Der *Peek*-Befehl ermöglicht, den Inhalt einer Speicherzelle der Prozessorkarte sowohl im Datenspeicher als auch im Festwertspeicher über die serielle Schnittstelle abzufragen.

Aufbau

@<GN>c<Addr>	(Festwertspeicher)
@<GN>b<Addr>	(Schreib-/Lesespeicher)
<GN>	= Gerätenummer, Standard = 0
<Addr>	= Adresse zwischen 0 und 65 536

Anwendung

@0c 2048
@0b 4711

Erläuterung Die Karte wird durch @0 adressiert. c gibt an, dass ein Wert aus dem Festwertspeicher gelesen werden soll. 2048 gibt die Adresse des zu lesenden Wertes an. Der Rechner antwortet mit dem Software-Handshake gefolgt von zwei Zeichen, die einen Hexadezimalwert angeben, der dem Inhalt der Speicherzelle entspricht. Soll ein Wert aus dem Datenspeicher gelesen werden, ist statt c der Befehlscode b zu verwenden.

Beschränkung -



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

@0b 65531

-



Dieser Befehl wird in erweiterter Form in Verbindung mit einer E/A-Erweiterungseinheit eingesetzt (vgl. Kapitel 2.3.3, Speicherbare Befehle: *Eingangsport lesen*).

2.1.10 Befehl: **Poke (Beschreiben von Speicheradressen)**

Zweck Der *Poke*-Befehl erlaubt es, den Speicherinhalt des Datenspeichers einer Prozessorkarte zu ändern.

Aufbau @<GN>B<Addr>,<Data>

<GN> = Gerätenummer, Standard = 0
<Addr> = Adresse zwischen 0 und 65 535
<Data> = Wert zwischen 0 und 255

Anwendung @0B 33211,128

Erläuterung Die Karte wird durch @0 adressiert. B gibt an, dass ein Wert in den Speicher geschrieben werden soll. 33211 gibt die Adresse des zu schreibenden Wertes an. 128 ist der neue Wert dieser Speicherzelle.
Der Rechner bestätigt die Ausführung mit dem Software-Handshake.

Beschränkung Der Befehl prüft nicht, ob ein am Datenbus angeschlossenes Gerät die Daten korrekt übernommen hat.



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

@0B33211,128

-



Der Befehl sollte nicht verwendet werden, um interne Kartenparameter zu ändern, da sich deren Adresse ohne Mitteilung ändern kann. Der Befehl sollte nicht mit Adressen kleiner 32767 benutzt werden, da diese Adressen als Datenspeicher von der Prozessorkarte genutzt werden.

2.1.11 Befehl: Batterie-RAM löschen

Zweck Der Befehl löscht alle Daten des RAM-Speichers auch wenn sie durch die Option *Memory Back-up* quasi-permanent gespeichert sind. Damit werden auch alle eventuell vorhandenen Angaben über Referenzgeschwindigkeiten, Achsen usw. zurückgesetzt.

Aufbau @<GN>k (direkt)
<GN> = Gerätenummer, Standard = 0

Anwendung @Ok

Erläuterung Die Karte wird durch @0 adressiert. k gibt an, dass das Batterie-RAM gelöscht werden soll. Der Rechner meldet sich nach erfolgter Ausführung mit der Rückmeldung (vgl. Software-Handshake).



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

@0k
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@Ok":gosub 1000
120 stop
1000 if loc(1)<1 then goto 1000
1010 a\$=input\$(1,1)
1015 if a\$="0" then return
1020 print "Karte meldet Fehler : ";a\$
1030 stop



Sollte sich das RAM auf diese Weise nicht mehr löschen lassen, ist es kurzzeitig aus seiner Fassung zu entnehmen und anschließend wieder einzusetzen.

2.1.12 Befehl: CR/LF setzen

Zweck Einige Computer müssen beim Datenempfang an der seriellen Schnittstelle ein Delimiter (Begrenzungs)-Zeichen erhalten. Dies ist eine betriebssystem-spezifische Eigenart einiger Prozessrechner (z. B. DEC VAX, HP-Prozessrechner). Das Delimiter-Zeichen wird vom Prozessrechner benötigt, um am Ende der Datenübermittlung einen Interrupt für den empfangenen Prozess auslösen zu können. Falls Sie einen MS-DOS-Rechner (IBM-PC, XT, AT o. ä.) verwenden, sollte dieser Befehl nicht verwendet werden.

Aufbau @<GN>C<STATUS> (direkt)
<GN> = Gerätenummer, Standard = 0
<STATUS> = 0 = AUS (Standard), 1 = EIN

Anwendung @0C1

Erläuterung Die Karte wird durch @0 adressiert. C gibt an, dass der Software-Handshake geändert werden soll. Die Karte meldet sich bei allen Befehlen dann mit der Folge :

Fehler CHR(13) CHR(10)

Beschränkung Die Karte lässt sich anschließend nur noch mit diesem neuen Protokoll ansprechen. Um in das alte Protokoll zurückzuschalten, muss das neue Protokoll zur Übergabe verwendet werden.

Ein Programmbeispiel ist nicht verfügbar, da weder PAL-PC noch GW-BASIC auf diesen Prozessrechnern verfügbar sind.

2.1.13 Befehl: **Gerätenummer setzen**

Zweck Die Gerätenummer der Prozessorkarte (<GN>) wird geändert. Zulässig sind Nummern zwischen 0 und 9 (chr(48) ... chr(58)). Die neue Gerätenummer bleibt bis zum Ausschalten des Gerätes aktiv.

Aufbau @<GN>G<GNneu> (direkt)

 <GN> = Gerätenummer, Standard = 0
 <GNneu> = Zeichen zwischen 0 und 9

Anwendung @0G1

Erläuterung Die Karte wird durch @0 adressiert. G gibt an, dass eine neue Gerätenummer zur Adressierung der Karte verwendet werden soll. Nach G erwartet die Karte die neue Gerätenummer. Der Rechner meldet sich nach erfolgter Speicherung mit der Rückmeldung (vgl. Software-Handshake).

Beschränkung Die Karte lässt sich anschließend nur noch mit dieser Gerätenummer ansprechen. Die Karte prüft nicht, ob eine zulässige Gerätenummer übergeben wird.



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

@0G1

```
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1, "@03":gosub 1000
120 print#1, "@0G1":gosub 1000
130 print#1, "@1i":gosub 1000
140 print#1, "m 8000,900,800,900":gosub 1000
150 print#1, "n 3":gosub 1000
160 print#1, "m 200,900,400,990":gosub 1000
170 print#1, "9":gosub 1000
180 print#1, "@1S":gosub 1000
190 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler : ";a$
1030 stop
```



Um PAL-PC eine Prozessorkarte mit geänderter Gerätenummer anzusprechen, wird der Befehl #GN verwendet. Die Prozessorkarte kann anschließend nur noch mit dieser Gerätenummer angesprochen werden.

2.1.14 Befehl: TRACE (Einzelschrittausführung)

Zweck	Die Prozessorkarte führt die gespeicherten Befehle einzeln aus. Nach jedem Befehl wird auf ein Zeichen an der seriellen Schnittstelle gewartet und der Befehlszählerstand mit allen zugehörigen relevanten Parametern ausgegeben.
Aufbau	@<GN>t <GN> = Gerätenummer, Standard = 0
Anwendung	@0t
Erläuterung	Die Prozessorkarte führt die Befehle wie bisher aus; vor jedem Befehl wird jedoch der Befehlszählerstand als Integer-Zahl ausgegeben. Nach dem Befehlszählerstand wird die Befehlsnummer und die Operationskonstante mit zugehörigen Daten ausgegeben. Die Zeile wird mit CR abgeschlossen. Der Rechner wartet dann auf ein Zeichen an der Schnittstelle. Anschließend wird der Befehl ausgeführt. Die Funktion verhält sich bei jedem auszuführenden Befehl wie folgt: (A): Der Trace-String wird ausgegeben (s. u.). Es wird auf ein Zeichen gewartet. Falls Zeichen = 127 dann wird µP-Reset ausgelöst. Befehl wird ausgeführt. Falls Befehl = Datenfeldende, dann Schluss. Sonst nächster Befehl, nach (A).

Der bei jedem Befehl übergebene Trace-String hat folgenden Aufbau

Aufbau	<u>01234</u>	<u>00001</u>	<u>30</u>	<u>000001</u>	<u>FE87...</u>	<u>FFFF01</u>	<u>FE01</u>
	A	B	C	D	E	F	G

- A Speicherzeiger - Er gibt an, wo der Befehl im Speicher der Prozessorkarte abgelegt ist.
- B Befehlszähler - Er gibt die Nummer des aktuellen NC-Befehles an.
- C NC-Befehlscode - Er gibt den auszuführenden Befehl an.
Die Angabe ist in Hexadezimal und bezieht sich auf den ASCII-Wert des Befehlscodes. Im Beispiel oben ist der Befehl 0 = Bewegung relativ gespeichert.
- D Befehlsparameter der X-Achse - Im obigen Beispiel ist die 24-Bit-Hexadezimal-Darstellung des Fahrweges in der 2er-Komplement-Darstellung.
- E Geschwindigkeitswert der X-Achse - Um die Geschwindigkeit rückzuwandeln, kann der Bruch $921600/(HI\text{-}Byte \cdot (256 - LO\text{-}Byte))$ gebildet werden.
- F Befehlsparameter wie D, jedoch für die z2-Angabe.
- G Geschwindigkeitsangabe für die z2-Angabe.

Bei anderen Befehlen werden die Parameter in der übergebenen Reihenfolge entweder als Zeichen oder als 2er-Komplement abgelegt.



Bei den Synchronisations-Befehlen ist zuerst das Zeichen für die Trace-Funktion und anschließend das Synchronisations-Zeichen zu übergeben. Die Zeichen <Leertaste>, <TAB> und <Linefeed> können nicht verwendet werden, um die Einzelschrittausführung fortzuschalten.

Um den Trace-Modus zu beenden, müssen Sie die Prozessorkarte ausschalten oder bei der Anfrage auf ein Zeichen DEL (char(127)) übergeben.

Die Parameter-Reihenfolge und -Speicherung kann sich im Rahmen der technischen Weiterentwicklung ohne Mitteilung bei Software-Versionen ändern.

Die Anzahl der rückgesendeten Befehlsparameter entspricht der gewählten Achsenzahl.

2.1.15 Befehl: **Selbsttest**

Zweck Diese Anweisung löst einen Selbsttest der Prozessorkarte aus. Im Gegensatz zum Selbsttest der durch Betätigen des Starttasters eingeleitet wird, bewirkt der Befehl nur die Ausführung des ersten Teiles der Testroutine, nicht den Verfah- und Schnittstellentest.

Aufbau @<GN>? (direkt)
<GN> = Gerätenummer, Standard = 0

Anwendung @0?

Erläuterung Die Karte wird durch @0 adressiert. ? gibt an, dass ein Selbsttest durchgeführt werden soll. Die Karte testet den Speicherbereich, Prozessor und Prozessorregister sowie interne Speicherbereiche. Anschließend werden einige Prozessorkartenbezogene Variablen und eine Prüfsumme ausgegeben.
Um den erweiterten Selbsttest auszuführen, müssen Sie den Starttaster festhalten und das Gerät einschalten.

Um den Selbsttest zu beenden, müssen Sie das Gerät ausschalten, vorher können keine anderen Befehle übergeben werden!



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

@0?

-



Zur weiteren Befehlsübergabe an die Prozessorkarte muss der Selbsttest abgeschlossen sein. Anderenfalls wird ein Listen-Zeichen (Fehler @) als Fehlermeldung gesendet. PALPC wird in diesem Fall Fehler 164 rückgemeldet.

2.2 Ergänzender Befehlsumfang der Interface-Karte 5.0

2.2.1 Befehl: 3D-Linearinterpolation

Zweck	Die Interface-Karte 5.0 erweitert die 2,5D-Interpolation des Standard-Betriebssystems auf eine 3D-Interpolation (räumlich). Mit diesem Befehl können Sie die Interpolation gezielt ein- und ausschalten.
Aufbau	<p>@<GN>z<STATUS></p> <p><GN> = Gerätenummer, Standard = 0 <STATUS> = 0 - 3D-Interpolation aus = 1 - 3D-Interpolation ein</p>
Anwendung	@0z1
Erläuterung	<p>Die Prozessorkarte wird mit dem Dateneröffnungsteil @0 auf einen neuen Befehl vorbereitet. z1 ändert die Interpolation von 2-Achs- auf 3-Achs-Betrieb.</p> <p>Die Anweisung wirkt modal, d. h. alle move- und moveto-Anweisungen werden dreidimensional ausgeführt. Die Angabe von z2-Parametern in diesen Verfahrbewegungen wird ignoriert. Die Geschwindigkeitsangabe der Interpolation muss bei der X-Angabe erfolgen.</p>



Programmierbeispiel

PAL-PC

```
#axis xyz;  
reference xyz;  
set3don;  
move 10(700),15(800),3(400),  
0(30);  
set3doff;
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@07":gosub 1000  
120 print#1,"@0r7":gosub1000  
130 print#1,"@0z1":gosub 1000  
140 print#1,"@0A100,700,150,800,30,400,0,30"  
145 gosub 1000  
150 print#@1,"@0z0":gosub1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

info

Die Einleitung einer Referenzfahrt schaltet automatisch auf 2,5D-Interpolation zurück.

Die korrekte Bearbeitung einer 3D-Interpolation setzt als Bezugsebene eine XY-Ebene voraus (vgl. Ebenenwahl).

Die maximale Geschwindigkeit für eine 3D-Interpolation beträgt 10 000 Schritte/s.

Die auf der Mechanik erreichbare Geschwindigkeit hängt von den angeschlossenen Motoren und Leistungsteilen ab.

Für Eilgang-Bewegungen sollten Sie kurzzeitig auf eine 2D-Interpolation umschalten und bei angehobener Z-Achse die Positionierung ausführen, da während einer 3D-Interpolation keine Kollisionsprüfung stattfindet.

Bei einer 3D-Interpolation ist nach einem Stopp-Befehl die aktuelle Position nicht korrekt.

Die Positionsabfrage liefert erst nach Ende der Ausführung die korrekten Werte zurück.

2.2.2 Befehl: Zirkularinterpolation

Zweck Bearbeiten von Kreisen und Kreisbögen mit konstanter Bahngeschwindigkeit. Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle ausgelöst. Der erste Befehl legt die Kreisrichtung fest, im zweiten werden die Interpolationsparameter übergeben.

Aufbau

Kreisrichtung: @<GN>f-1 Gegen den Uhrzeigersinn
@<GN>f0 Uhrzeigersinn

Kreisbogen: @<GN>y B,V,D,Xs,Ys,Rx,Ry

B	= Bogenlänge in Schritten
V	= Geschwindigkeit (30 bis 10 000)
D	= Interpolationsparameter
Xs	= Startpunkt x
Ys	= Startpunkt y
Rx	= Richtung x
Ry	= Richtung y

Berechnung der Parameter

Bogenlänge B Die Bogenlänge gibt die Länge des Bogens zwischen dem Start- und Endpunkt der Kreisinterpolation in Schritten an. Zur Berechnung des Parameters können Sie auf nachstehende Programmteile zurückgreifen. Hierbei gilt:

A - Anfangswinkel des Bogen- oder Kreissegmentes
 $A = \pi \cdot \text{Anfangswinkel} / 180$

E - Endpunkt der Bewegung
 $E = \pi \cdot \text{Endwinkel} / 180$

B - die resultierende Bogenlänge



Zur Berechnung der Bogenlänge dürfen nur Winkel im Bogenmaß verwendet werden.

1. Approximierende Formel (nur bei Viertel-, Halb- und Vollkreisen)

$$B = 4 \cdot \text{Radius} \cdot (E - A) / \pi$$

2. Berechnung der Bogenlänge mittels Software-Routine

```
if (Kreisrichtung=Gegenuhrzeigersinn) then
  begin
    while(A<0) do A:=A+2.0*pi;
    while(E<0) do E:=E+2.0*pi; {Winkel in den positiven Bereich normieren}
    while (A>=pi/2.0) do
      begin
        A:=A-pi/2;
        E:=E-pi/2;
      end;
    B:=0.0;
    while (E-A>=pi/2.0) do
      begin
        E:=E-pi/2.0;
        B:=B+2.0*Radius;
      end;
    B:=B+Radius*(cos (A) -cos (E) +sin (E) -sin (A));
  end;
else {Kreisrichtung = Uhrzeigersinn}
  begin
    while (A>0) do A:=A-2.0*pi;
    while (E>0) do E:=E-2.0*pi; {Winkel in den negativen Bereich normieren}
    while (A<=-pi/2.0) do
      begin
        A:=A+pi/2;
        E:=E+pi/2;
      end;
    B:=0.0;
    while (A-E>=pi/2.0) do
      begin
        E:=E+pi/2.0;
        B:=B+2.0*Radius;
      end;
    B:=B+Radius*(cos (A) -cos (E) +sin (A) -sin (E));
  end;
if (B<0) then B:= -B;
```

info

Die berechnete Bogenlänge muss auf den nächsten, ganzzahligen Wert gerundet übergeben werden. Hierbei sind Werte im Bereich von 3 bis 8 000 000 Schritten zulässig.

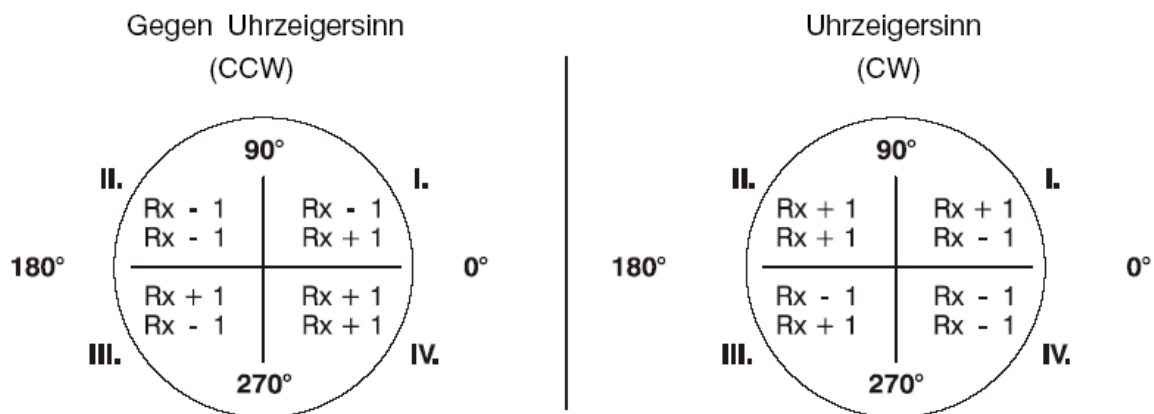
Geschwindigkeit V

Als Geschwindigkeitsangaben sind alle ganzzahligen Werte von 30 bis 10 000 Schritte/s zulässig.

Ob und mit welchen Geschwindigkeiten die Interpolation ausgeführt werden kann, hängt von den verwendeten Leistungsteilen und der angeschlossenen Mechanik ab.

Richtung Rx und Ry

Die Parameter Rx und Ry geben der Prozessorkarte an, in welchem Quadranten des Kreises die Interpolation startet.



Startpunkt Xs und Ys

Diese Parameter geben die Startpunkte Xs und Ys relativ auf den Kreismittelpunkt an.

Zur Berechnung gelten folgende Formeln:

$$Xs = \text{Radius} * \cos(A)$$

$$Ys = \text{Radius} * \sin(A)$$

Interpolationsparameter D

D muss übergeben werden, da die Prozessorkarte aufgrund ihrer Speicherkapazität nicht in der Lage ist, diesen Parameter zu berechnen.
Zur Berechnung können Sie folgenden Programmausschnitt heranziehen.

```
function Summe(xx:real):real;
begin
  if(xx>0) then
    Summe:=xx*(xx+1)
  else
    Summe:= -xx*(xx-1)
  end;

function formel:real;
begin
  if (Kreisrichtung=Gegenuhrzeigersinn) then
    Formel:= (Rx*Ry*Radius+Rx*Ry*Summe(Radius-1.0)
              -Rx*Summe(Xs+(Rx-Ry)/2.0)+Ry*Summe(Ys+(Rx+Ry)/2.0))/2;
  else
    {Richtung = Uhrzeigersinn}
    Formel:= (-Rx*Ry*Radius-Rx*Ry*Summe(Radius-1.0)
              - Rx*Summe(Xs+(Rx+Ry)/2.0) + Ry*Summe(Ys+(Ry-Rx)/2))/2;
  end;
D:=Formel;
```



Der berechnete Parameter muss als gerundet und als ganzzahliger Wert übergeben werden.



Programmierbeispiel zur Parameter-Berechnung

Es soll ein Viertelkreis (90°) entgegen dem Uhrzeigersinn mit einem Radius von 200 Schritten und einer Verfahrensgeschwindigkeit von 1 500 Schritten/s ausgeführt werden.
Als Startwinkel ist 135° festgelegt.

1. Winkel im Bogenmaß:
A $= \pi * 135 / 180$
 $= 3 * \pi / 4$
B $= \pi * (135 + 90) / 180$
 $= 5 * \pi / 4$
2. Bogenlänge: (nach Näherungsformel)
B $= 4 * \text{Radius} * (E - A) / 180$
 $= 2 * 200 * (5 * \pi / 4 - 3 * \pi / 4) / \pi$
 $= 400$
3. Richtung Rx, Ry:
Anfangswinkel $= 135^\circ$, Gegenuhrzeigersinn
Rx $= -1$
Ry $= -1$
4. Startpunkt Xs, Ys:
Xs $= \text{Radius} * \cos(\text{Anfangswinkel})$
 $= 200 * \cos(135)$
 $= -141$
Ys $= \text{Radius} * \sin(\text{Anfangswinkel})$
 $= 200 * \sin(135)$
 $= 141$

5. Interpolationsparameter D: (Gegenuhrzeigersinn)

$$D = \frac{(R_x * R_y * \text{Radius} + R_x * R_y * \text{Summe}(\text{Radius} - 1.0) - R_x * \text{Summe}(X_s + (R_x - R_y) / 2.0) + R_y * \text{Summe}(Y_s + (R_x + R_y) / 2.0))}{2}$$

$$\begin{aligned} \text{Summe}(\text{Radius}-1) &= \text{Summe}(200-1) \\ &= \text{Summe}(199) \\ &= 199 * (199+1) \\ &= 39\,800 \end{aligned}$$

$$\begin{aligned} \text{Summe}(X_s + (R_x - R_y) / 2.0) &= \text{Summe}(-141 + (-1 - (-1)) / 2.0) \\ &= \text{Summe}(-141) \\ &= -(-141) * ((-141) - 1) \\ &= -20\,022 \end{aligned}$$

$$\begin{aligned} \text{Summe}(Y_s + (R_x + R_y) / 2.0) &= \text{Summe}(141 + (-1 + (-1)) / 2.0) \\ &= \text{Summe}(141-1) \\ &= 140 * (140+1) \\ &= 19\,740 \end{aligned}$$

$$\begin{aligned} D &= \frac{(R_x * R_y * \text{Radius} + R_x * R_y * 39800 - R_x * (-20022) + R_y * 19740)}{2} \\ &= \frac{((-1) * (-1) * 200 + (-1) * (-1) * 39800 - (-1) * (-20022) + (-1) * 19740)}{2} \\ &= \frac{(200 + 39800 - 20022 - 19740)}{2} \\ &= 119 \end{aligned}$$

Der Programmabschnitt muss also folgendermaßen aussehen:

```
...
...
@0f-1
@0y400,1500,119,-141,141,-1,-1
...
...
```

bzw. im Direktformat:

```
...
...
f-1
y400,1500,119,-141,141,-1,-1
...
...
9
@0s
```

2.3 Ergänzender Befehlsumfang bei Interface-Karten mit E/A Erweiterung

2.3.1 Befehl: Externe Speicherung

Zweck	Speichern eines CNC-Programmes auf ein externes Speichermedium.
Aufbau	@<GN>u <GN> = Gerätenummer, Standard = 0
Erläuterung	Dieser Befehl ermöglicht, ein im Datenspeicher der Prozessorkarte befindliches CNC-Programm auf einem Datenspeicher im Scheckkartenformat (Memory-Card) abzuspeichern oder von einer Speicherkarte zurück in die Prozessorkarte zu lesen. Dabei ist folgender Ablauf einzuhalten:

1. Übertragen von Prozessorkarte zur Memory-Card
 - a) Programm wie gewohnt in Prozessorkarte übertragen
 - b) Memory-Card einstecken
 - c) Befehl @0u übergeben
 - d) Memory-Card entnehmen
2. Übertragen von Memory-Card nach Prozessorkarte
 - a) Controller einschalten
 - b) Memory-Card einstecken
 - c) µP-Reset drücken
 - d) Memory-Card entnehmen



Die Memory-Card sollte beim Einschalten der Steuerung nicht gesteckt sein. Bei den Memory-Card-Typen 8 k x 8 und 16 k x 8 wird nicht die vorhandene Speicherkapazität geprüft, d. h. dass bei komplexen Programmen die Speichergrenze ohne Fehlermeldung überschritten wird.

2.3.2 Befehl: Ausgangsport setzen

Zweck Die Prozessorkarte setzt am definierten Ausgangsport der E/A-Erweiterungseinheit ein gewünschtes Ausgangsmuster.

Aufbau @<GN>B<ADRESSE>, <WERT>

<GN> = Gerätenummer, Standard = 0
<ADRESSE> = Ausgangsport 1 ———> 65 529
= Ausgangsport 2 ———> 65 530
<WERT> = 0 ... 255

Erläuterung Dieser Befehl entspricht im wesentlichen dem Poke-Befehl des Standard-Betriebssystems 5.x. Im CNC-Betrieb (Speicher-Modus) wird zusätzlich zur byteweisen Bearbeitung eine bitweise Bearbeitung der Ausgangsports möglich. So können Sie gezielt einzelne Bit setzen oder löschen.

2.3.3 Befehl: Eingangsport setzen

Zweck Die Prozessorkarte liest das anstehende Bitmuster am Eingangsport der E/A-Erweiterung ein.

Aufbau @<GN>b<ADRESSE>

<GN> = Gerätenummer, Standard = 0
<ADRESSE> = Eingangsport 65 531

Erläuterung Der Befehlsaufbau ist identisch mit dem *Peek*-Befehl des Standard-Betriebssystems 5.x. Im Speicher-Modus (CNC-Betrieb) besteht darüber hinaus die Möglichkeit, in Abhängigkeit eines Bitmusters Verzweigungen nach Vorwärts und Rückwärts zu realisieren.

2.4 Ergänzender Befehlsumfang EP 1090

2.4.1 Befehl: Output-Modul

Zweck Hauptspindel zu- oder abschalten. Die Interface-Karte schaltet die Hauptspindel ein bzw. aus.

Aufbau @<GN>h<Parameter> (DNC)
h<Parameter> (speicherbar)
<GN> = Gerätenummer, Standard =0
<Parameter> = 1 Hauptspindel einschalten
= 0 Hauptspindel abschalten

Anwendung @0h1

Erläuterung Die Karte wird durch @0 adressiert. h gibt an, dass das Output-Modul geschaltet werden soll. Der Parameter 1 bewirkt das Einschalten des integrierten Relais. Sollte sich die Spindel bereits im Ein-Zustand befinden, hat der Befehl keine Auswirkungen. Der Zustand der Hauptspindel wird durch den Dezimalpunkt der 7-Segment-Anzeige angezeigt. Der Rechner meldet sich nach erfolgter Ausführung mit der Rückmeldung 0.

2.5 Ergänzender Befehlsumfang ab Interface-Karte Version AZ 1350/5

2.5.1 Befehl Magnetbremse

Zweck Der Befehl ermöglicht eine softwaremäßige Aktivierung bzw. Deaktivierung eines speziellen Ausgangs, der über Relais verstärkt, Magnetbremsen in Antriebseinheiten steuert.

Aufbau @<GN> g <Status>

<GN> = Gerätenummer
<Status> = 1 Bremse magnetisiert
= 0 Bremse inaktiv

Anwendung @0g1

Erläuterung Die Karte wird durch @0 adressiert, g gibt an, dass das Brems-Relais geschaltet werden soll. Status 1 bewirkt das Einschalten des Relais. Dadurch wird eine angeschlossene Magnet-Bremse aktiviert und gibt die Antriebsachse des Motors frei. Bei Status 0 wird die Bremse nicht bestromt, d. h. dass die Antriebsachse des Motors gebremst wird. Der Rechner meldet sich nach erfolgter Ausführung mit der Rückmeldung 0.

2.6 Kontroll- und Steuercodes

Kontroll- und Steuercodes ermöglichen über die serielle Schnittstelle den direkten Eingriff in den Funktionsablauf der Interface-Karte. Dabei werden die jeweils gesendeten Kommandos ohne Verzögerung ausgeführt.

2.6.1 Befehl: Selbsttest

Zweck Die Prozessorkarte überprüft die Funktionsbereitschaft ihrer Funktionsbaugruppen.

Aufbau chr(252)

Erläuterung Die Interface-Karte überprüft die Kapazität ihres Datenspeichers, die Checksumme ihres Betriebs-Eproms sowie die Schalterstellung des DIP-Schalters. Anschließend werden zum Test der angeschlossenen Schrittmotoren einige Taktimpulse an die Leistungselektronik der X und Y-Achse ausgegeben. Abgeschlossen wird die Test-Routine durch die permanente Ausgabe eines ASCII-Zeichensatz über die serielle Schnittstelle.

Beschränkung -



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

chr(252)

-



Den Selbsttest können Sie nur beenden, wenn Sie die Versorgungsspannung ausschalten oder einen µP-Reset durchführen.

2.6.2 Befehl: **STOP**

Zweck Die Prozessorkarte bricht eine momentane Verfahrbewegung ab.

Aufbau chr(253)

Erläuterung Eine Positionierbewegung im DNC-Modus (relativ oder absolut) kann durch einen STOPP-Befehl unterbrochen werden, ohne dass Schrittverluste auftreten. Ein danach ausgeführter START-Impuls beendet den unterbrochenen Funktionsablauf. Außerdem können Sie nach einem STOPP-Befehl mit Hilfe des Befehles *Positionsabfrage* die aktuelle erreichte Position rücklesen.

Beschränkung Der Befehl können Sie nur verwenden, wenn eine Positionierbewegung stattfindet.



Programmierbeispiel

PAL-PC

GW-BASIC

(Terminalmode)

chr(253) -



Die Prozessorkarte meldet als Quittierung den Stopp-Fehler zurück.

Da der Befehl ohne Adressierung arbeitet, werden die Verfahrbewegungen aller angeschlossenen Prozessorkarten unterbrochen.

Der übergeordnete Rechner muss die letzte anzufahrende Position im DNC-Modus erneut senden.

2.6.3 Befehl: **µP-Reset**

Zweck	Die Prozessorkarte unterbricht sofort alle Aktivitäten und geht in den Reset-Zustand
Aufbau	chr(254)
Erläuterung	Durch einen µP-Reset wird der integrierte Mikro-Controller abrupt in seinen Einschaltzustand zurückgeschaltet. Die Ausgänge liegen während des Reset-Zustandes auf Vcc-Potential und schalten nach Aufheben auf GND-Potential.
Beschränkung	-



Programmierbeispiel

PAL-PC GW-BASIC

(Terminalmode)

chr(254) -

2.6.4 Befehl: **Break**

Zweck	Der Break-Befehl dient zum Abbruch des Aktuellen Positioniervorganges.
Aufbau	chr(255)
Erläuterung	Durch Senden eines Break-Befehles an der seriellen Schnittstelle wird der aktuelle Positioniervorgang der Interface-Karte abgebrochen, ohne dass eine Stopp-Rampe eingeleitet wurde. Dadurch auftretende Schritt-fehler werden ignoriert. Im Gegensatz zum ähnlich wirkenden µP-Reset, kann nach dem <i>Break</i> -Befehl normal weitergearbeitet werden ohne eine Neuinitialisierung der Prozessorkarte vorzunehmen.

Beschränkung -



Programmierbeispiel

PAL-PC GW-BASIC

(Terminalmode)

chr(255) -

3 Befehlsaufbau CNC-Befehle

Im CNC-Modus speichert die Prozessorkarte alle übersendeten Befehle im internen Datenspeicher. Zur Aktivierung ist nach der Standard-Initialisierung

@<Gerätenummer> <Achszahl> der Befehl @<Gerätenummer>i zu übertragen.

Anschließend wird das Datenfeld übergeben und mit dem Befehl 9 als

Datenfeldende-Zeichen abgeschlossen.

Jetzt können Sie das Programm erneut durch einen externen Start-Befehl aktivieren.

Als Start können Sie sowohl eine Start-Taste (z. B. in der Frontplatte der Prozessorkarte) als auch den Befehl @<Geräte- nummer>S verwenden.

Durch den physikalisch begrenzten RAM-Speicher der Prozessorkarten ist die Anzahl der speicherbaren Befehle auf ca. 1 200 im 3-Achs-Betrieb, ca. 1 800 im 2-Achs-Betrieb und ca. 2 400 im 1-Achs-Betrieb eingeschränkt.

Um bei Ausfall der Versorgungsspannung (z. B. Abschalten der Versorgungsspannung) keinen Datenverlust des RAM-Speichers zu erleiden, können Sie durch einen optionalen Akku bzw. einer Primärzelle ein sogenanntes Memory-back-up erzeugen.

Im folgenden werden die speicherbaren Befehle der Prozessorkarte aufgelistet und kurz erläutert. Eine Detailerklärung kann unter dem entsprechendem Befehl des Direkt-Modus nachgeschlagen werden.

3.1 Grundbefehlssatz ab Prozessorkarte 4.0

3.1.1 Befehl: INPUT

Zweck Die Prozessorkarte wird in den Speicher-Modus gesetzt. Alle nachfolgenden Befehle werden im internen Datenspeicher abgelegt. Die Ausführung der gespeicherten Befehle wird mit dem Befehl Start oder der Start-Taste veranlasst.

Aufbau @<GN>i
<GN> = Gerätenummer, Standard = 0

Anwendung @0i

Erläuterung Die Karte wird durch @0 initialisiert. i gibt an, dass zu speichernde Befehle folgen. Nach Empfang des Befehles erwartet die Prozessorkarte ein komplettes NC-Programm - bestehend aus speicherbaren Befehlen. Dieses muss mit dem Datenfeld-Ende (9) abgeschlossen sein. Das Datenfeld kann alle speicherbaren Befehle enthalten.

Beschränkung Den Befehl können Sie erst verwenden, nachdem die Achsenanzahl gesetzt, d. h. die Prozessorkarte initialisiert ist.



Programmierbeispiel

PAL-PC

```
#axis xy;  
move 50(100),40(100);  
stop.  
#start
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@03":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"0 50,100,40,100":gosub 1000  
140 print#1,"9":gosub 1000  
150 print#1,"@0S":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler: ";a$  
1030 stop
```



Der *Input*-Befehl löscht die bisher gespeicherten Anweisungen. Anschließend erwartet die Prozessorkarte ein komplettes Datenfeld - abgeschlossen mit dem Datenfeld-Ende-Kennzeichen (9).

Tritt während der Übertragung der zu speichernden Befehle ein Fehler auf, verlässt die Prozessorkarte den Input-Modus, das bis dahin übergebene NC-Programm geht verloren.

3.1.2 Befehl: Referenzfahrt

Zweck Die Prozessorkarte verfährt alle angegebenen Achsen an ihre Nullpunkte (Referenzpunkte)

Aufbau 7<Achsen>
<Achsen> = ganzzahliger Wert zwischen 1 und 7

Erläuterung 7 gibt an, dass eine Referenzfahrt ausgeführt werden soll. Der anschließende Zahlenwert definiert die Achsen, die eine Referenzfahrt durchführen sollen.

x	= 1	xz	= 5
y	= 2	yz	= 6
xy	= 3	xyz	= 7
z	= 4		

Die Reihenfolge der Ausführung ist dabei wie folgt festgelegt:

—> Z-Achse —> Y-Achse —> X-Achse

Dies gilt auch, wenn mit dem Ebenenbefehl eine andere Achse als Werkzeugachse bestimmt worden ist. Hier kann gegebenenfalls die getrennte Referenzfahrt der Achsen Kollisionen mit dem Werkstück verhindern.



Programmierbeispiel

PAL-PC

#axis xyz;
reference xyz;

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@07":gosub 1000stop.
120 print#1,"@0i":gosub 1000
130 print#1,"77":gosub 1000
140 print#1,"9":gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler:";a$
1030 stop
```



(siehe Befehl *Referenzfahrt* Kapitel 2.1.2)

3.1.3 Befehl: **Bewegung relativ**

Zweck

Die Prozessorkarte gibt entsprechend der übergebenen Schrittzahl und Schrittgeschwindigkeit für jede Leistungsstufe eine Impulskette aus.

Die Verfahrenbewegung wird sofort ausgeführt oder gespeichert.

Aufbau

0 <Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2>

0 = Bewegung relativ
<Sx> = Schrittzahl x, Wert zwischen 0 und $\pm 8\,388\,607$
<Gx> = Geschwindigkeit x, Wert zwischen 30 und 10 000
•
•
<Gz2> = Geschwindigkeit Z-Achse (2. Bewegung)

Erläuterung

0 gibt an, dass eine relative Bewegung erfolgen soll. Die Prozessorkarte erwartet nun für jede Achse ein Zahlenpaar bestehend aus Schrittzahl und Geschwindigkeit.



Programmierbeispiel

PAL-PC

```
#axis xyz;  
move 50(500),300(900),  
      20(200),-20(900);  
move 20(300),300(3000),  
      0(21),0(21);  
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@03":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"0 35,800,250,2000":gosub 1000  
140 print#1,"0 20,2000,-25,1000":gosub 1000  
150 print#1,"9":gosub 1000  
160 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



(siehe Befehl *Bewegung relativ* Kapitel 2.1.4)

3.1.4 Befehl: Bewegung zur Position (move to)

Zweck	Die Prozessorkarte fährt mit den angegebenen Geschwindigkeiten an die angegebene Position. Die Verfahrbewegung wird sofort ausgeführt.
Aufbau	m <Sx>, <Gx>, <Sy>, <Gy>, <Sz1>, <Gz1>, <Sz2>, <Gz2>
Erläuterung	m gibt an, dass eine Absolut-Position folgt. Aus Kompatibilitätsgründen zum relativen Positionierbefehl werden auch hier für die Z-Achse zwei Zahlenpaare erwartet. Die zweite Positionsangabe der Z-Position muss Null sein. Diese Zahl wird ignoriert, muss aber vorhanden sein



Programmierbeispiel

PAL-PC

```
#axis xy;  
moveto 50(500),300(900);  
moveto 50(500),300(900);  
moveto 20(200),30(900);  
stop.
```

GW-BASIC

```
100 open"com1.9600,N,8,1,DS,CD"as #1  
110 print#1,"@03":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"m 500,800,200,31":gosub 1000  
140 print#1,"m31,500,40,500":gosub 1000  
150 print#1,"9":gosub 1000  
160 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



(siehe Befehl *Bewegung zur Position* Kapitel 2.1.5)

3.1.5 Befehl: Nullpunktverschiebung

Zweck	Die Prozessorkarte speichert die momentane Position als virtuellen Nullpunktes für die angegebene(n) Achse(n). Die nächsten <i>Verfahre absolut</i> -Anweisungen berücksichtigen diesen virtuellen Nullpunkt als neuen Bezugspunkt.
Aufbau	n<Achsen> <Achsen> = ganzzahliger Wert zwischen 1 und 7
Erläuterung	n gibt an, dass eine Nullpunktverschiebung vorgenommen werden soll. Nach dem Befehl werden müssen Sie dem Rechner die Achsen mitteilen, für die eine Nullpunktverschiebung durchgeführt werden soll. Hierbei gilt die Zuordnung x = 1, y = 2, z = 4. Soll für mehrere Achsen eine Nullpunktverschiebung durchgeführt werden, müssen obige Werte addiert werden.



Programmierbeispiel

PAL-PC

```
#axis xy:
move 350(800),200(800);
null xy;
move 20(500),30(300);
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@03":gosub 1000
120 print#1,"@0i":gosub 1000
130 print#1,"0 8000,900,800,900":gosub 1000
140 print#1,"n 3":gosub 1000
150 print#1,"m 200,900,400,990":gosub 1000
160 print#1,"9":gosub 1000
170 print#1,"@0S":gosub 1000
180 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler : ";a$
1030 stop
```



(siehe Befehl *Nullpunktverschiebung* Kapitel 2.1.7)

3.1.6 Befehl: Ebenenwahl

Zweck 2,5-D interpolierende Prozessorkarten (z. B. Interface-Karte 4.0) können nur zwei von drei Achsen interpolieren. Im Einschaltzustand sind dies die X- und Y-Achse. Durch den *Ebenenwahl*-Befehl haben Sie die Möglichkeit, jede andere Ebenenkonfiguration als Hauptebene zu definieren. Die übrigbleibende dritte Achse wird als Werkzeugachse aufgefasst und anschließend an die Positionierung der Hauptachsen verfahren.

Aufbau e<Ebene>

<Ebene>	= Zahl zwischen 0 und 2
0	= xy
1	= xz
2	= yz

Erläuterung (siehe Kapitel 2.1.8)



Programmierbeispiel

PAL-PC

```
#axis xyz;  
line yz;  
move 20(200),33(500),  
40(1000),0(21);  
stop.
```

GW-BASIC

```
100 open"com1.9600,N,8,1,DS,CD"as #1  
110 print#1,"@07":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"e2":gosub 1000  
140 print#1,"m20,200,30,900,33,900, 0,21"  
150 gosub 1000  
160 print#1,"9":gosub 1000  
170 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



(siehe Befehl *Ebenenwahl* Kapitel 2.1.8)

3.1.7 Befehl: Synchronisationszeichen senden

Zweck

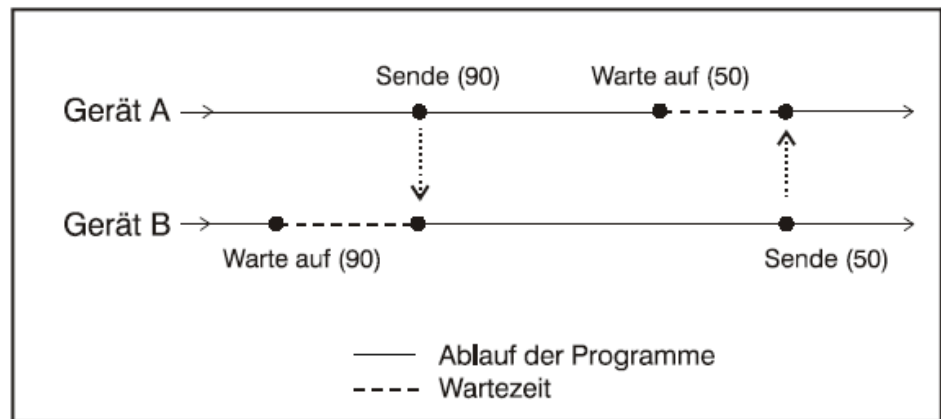
Die Prozessorkarte teilt einer zweiten Prozessorkarte oder einem übergeordneten Rechner mit, dass ein bestimmter Punkt im Ablaufschema (NC-Programm) erreicht ist. Der Befehl dient dazu, die Prozessorkarte mit einer externen Einheit zu synchronisieren bzw. eine externe Einheit aufzufordern, eine Tätigkeit vorzunehmen.

Aufbau

1 <SyncChar>
<SyncChar> = Synchronisationszeichen zwischen 33 und 125

Erläuterung

Die Prozessorkarte sendet an die serielle Schnittstelle ein definiertes ASCII Zeichen. Die Empfangsstation wartet - bedingt durch den Befehl *Warte auf Synchronisationszeichen* - auf das entsprechende Zeichen und fährt nach Erhalt mit dem programmierten CNC-Ablauf fort. Nachfolgendes Schema zeigt einen kurzen Überblick über den Funktionsablauf.



Beschrankung

Durch die Befehle *Synchronisationszeichen senden* und *Warte auf Synchronisationszeichen* konnen ohne ubergeordneten Rechner nur zwei Prozessorkarten synchronisiert werden. Die ubergebene Nummer des Synchronisationszeichens muss ein druckbares Zeichen im Bereich von 33 - 125 sein, da andere Zeichen von der Prozessorkarte gefiltert werden. Das Zeichen 64 sollte nicht verwendet werden, da dieses Zeichen den Datenverkehr wartender Prozessorkarten offnet. Die seriellen Schnittstellen der Gerate sind uber das Verbindungskabel Interface - Interface zu verbinden.



Programmierbeispiel

PAL-PC

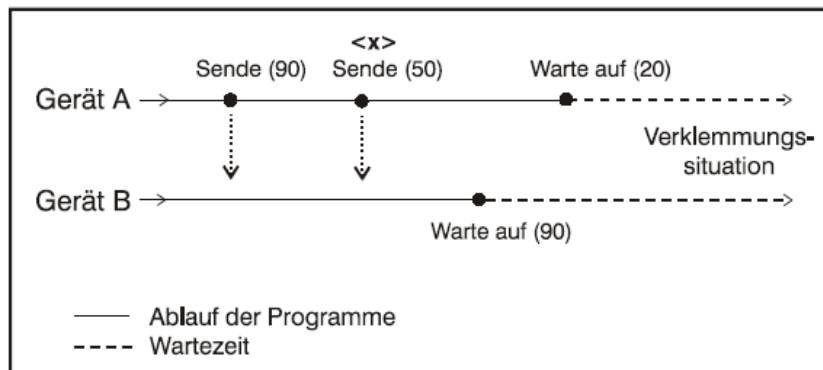
```
#axis xyz;
#input
send 90;
•
•
•
```

GW-BASIC

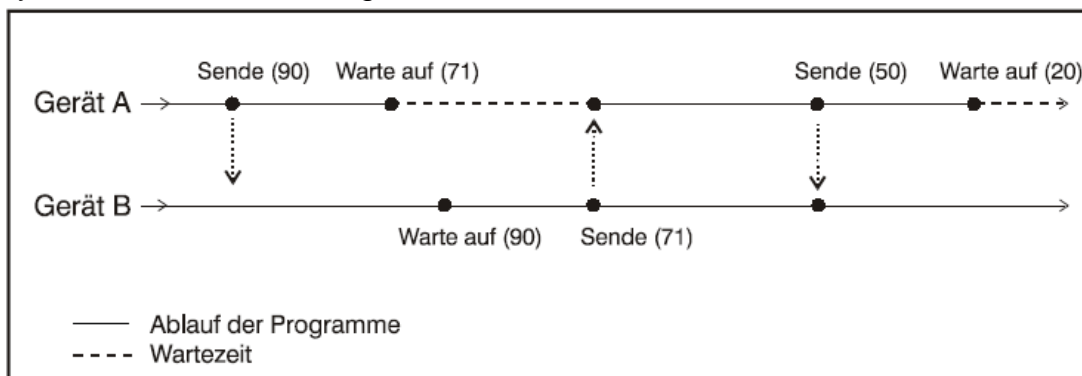
```
100 open"com1:9600,N,8,1,DS,CD"as #1
110 print#1,"@07":gosub 1000
120 print#1,"@0i":gosub 1000
130 print#1,"1 90":gosub 1000
140 print#1,"9":gosub 1000
150 print#1,"@0s":gosub 1000
160 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler : ";a$
1030 stop
```

Zur Prüfung des PAL-PC-Programmes können Sie die Funktion *Kommunikation* verwenden, zur Prüfung des BASIC-Programmes können Sie das bei dem Schnittstellentest angegebene Programm benutzen.

Die Interface-Karte kann während des Ablaufes einer gespeicherten Befehlsfolge nur ein Zeichen empfangen und zwischenspeichern. Folgende Situation führt daher zwangsweise zu einer Verklemmung des Gesamtsystems:



Bei <x> steht im Eingangspuffer des Gerätes B ein Zeichen char(50) (vorher gesendetes Zeichen wurde überschrieben). Der Prozess wird somit "ewig" auf das notwendige Zeichen char(90) warten. Aus diesem Grund sollte das sendende Gerät auf eine Bestätigung des empfangenden Gerätes warten, bevor ein erneutes Synchronisationszeichen gesendet wird.



3.1.8 Befehl: Warte auf Synchronisationszeichen

Zweck	Die Prozessorkarte wartet auf den Empfang des angegebenen Zeichens an der seriellen Schnittstelle. In Verbindung mit einem übergeordneten Rechner kann der Befehl für Verzweigungen im gespeicherten Ablauf genutzt werden.	
Aufbau	2 <SyncChar>, <SyncChar> <Offset>	<Offset> = Synchronisationszeichen von 33 bis 125 = Verzweigung bei Empfang <SyncChar>+1 Zahl zwischen - 32 767 und + 32 767
Anwendung	2 50,0 502 55,7	Warte auf Synchronisationszeichen Warte auf 55, verzweige 7 Befehle vorwärts bei Empfang von 56
Erläuterung	Verwendung des Befehles vgl. "Sende Synchronisationszeichen". In Verbindung mit einem übergeordneten Rechner können Sie diesen Befehl auch für logische Entscheidungen im Prozessablauf nutzen:	

Programmschritt

- 1
- 2
- 3 wait 50,-1 <— ext. Rechner sendet 50 oder 51
- 4
- 5

Im Ablauf des Datenfeldes wird bei Befehl 3 gestoppt. Sendet der übergeordnete Rechner ein char(50), wird als nächster Befehl Nr. 4 ausgeführt; sendet der Rechner ein char(51), wird als nächster Befehl der Befehl Nr. 2 ausgeführt.

Allgemein: Wird das auf das zu wartende Zeichen folgende Zeichen von der Prozessorkarte empfangen, findet die angegebene Verzweigung statt, anderenfalls wird der dem Wartebefehl folgende Befehl ausgeführt.



PAL-PC

```
#axis x;  
input  
label: move 3(1000);  
wait 50,label;  
stop.  
#start
```

Programmierbeispiel

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1#  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"0 500,5000":gosub 1000  
140 print#1,"2 50,-1":gosub 1000  
150 print#1,"9":gosub 1000  
160 print#1,"@0s":gosub 1000  
170 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

Dies Programm wird nach der Übertragung zur Prozessorkarte zunächst mit @0s gestartet. Im Programmablauf folgt eine relative Bewegung der X-Achse. Anschließend wartet die Prozessorkarte an der seriellen Schnittstelle auf den Empfang von char(50) oder char(51). Bei einem gesendeten char(51), verzweigt die Karte zurück, es wird dann erneut die relative Bewegung ausgeführt. Bei Empfang des char(50) wird das Programm beendet.

Bitte beachten Sie, dass eine Verzweigung vor oder hinter das Ende des Datenfeldes zu nicht vorhersehbaren Ergebnissen führen kann.

3.1.9 Befehl: Schleife/Verzweigung

Zweck	Programmschleifen dienen dazu, gleichartige Bewegungsabläufe zusammenzufassen. Hierdurch wird der zur Verfügung stehende Speicherplatz der Prozessorkarte besser genutzt. Durch Verzweigungen kann nach einer logischen Entscheidung an einen bestimmten Punkt des Prozesses zurückverzweigt werden.		
Aufbau	3 <Anzahl>,<Offset> <Anzahl> <Offset>	= Schleifen: Verzweigung: =	0 < Zahl < 32 767 0 Anzahl der zu wiederholenden Befehle bzw. Verzweigungsziel relativ Schleifen: - 1 > Zahl > - 3 000 Verzweigung: - 3 000 < Zahl < 3 000
Anwendung	3 25,-1 3 0,-5 3 0,5 3 6,-5	Wiederhole letzten Befehl 25 mal Verzweige immer 5 Schritte zurück Überspringe die nächsten 4 Befehle Wiederhole die letzten 5 Befehle 6 mal	
Erläuterung	Trifft die Prozessorkarte innerhalb des CNC-Programmablaufes auf den Befehl 3, wird ein Schleifenzähler eingerichtet, vorbesetzt und der Befehlszähler um den angegebenen Offset korrigiert. Die Befehle bis zum Schleifenzähler werden so oft wiederholt, bis der Schleifenzähler Null erreicht. Anschließend wird mit der Ausführung des ersten Befehles nach der Schleife fortgefahren. Falls die Angabe der Schleifenanzahl 0 ist, wird eine erzwungene Verzweigung ausgelöst.		



Programmierbeispiel

PAL-PC

```
#axis x;  
#input  
repeat  
repeat  
move 2(1000);  
until 5;  
move -10(2000);  
until 10;  
stop.  
#start
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"0 200,2000":gosub 1000  
140 print#1,"3 5,-1":gosub 1000  
150 print#1,"0 -1000,1000":gosub 1000  
160 print#1,"3 10,-3":gosub 1000  
170 print#1,"9":gosub 1000  
180 print#1,"@0S":gosub 1000  
190 stop  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

info

Es darf nicht vor dem Anfang des Datenfeldes verzweigt werden.
Vorwärtsschleifen (3 10,10) sind unzulässig.
Eine Schleife wiederholt immer die letzten n-Befehle.
Es muss mindestens ein Befehl wiederholt werden; 3 10,0 ist unzulässig.
Schleifen dürfen geschachtelt sein, die maximale Schachtelungstiefe beträgt 15.
Eine Schleife darf nicht durch eine Verzweigung vorwärts verlassen werden.

3.1.10 Befehl: Impulssteuerung

Zweck Die Hardware-Option *Impulsausgabe* erweitert die Signalaus- und -eingänge an der Prozessorkarte um einen gesonderten Port.

Sie können ihn sowohl als Eingang als auch als Ausgang verwenden.

Aufbau 4 <Option>
<Option> = ganze Zahl zwischen 1 und 6.

- 1 = Ausgang auf EIN setzen
- 2 = Ausgang auf AUS setzen
- 3 = Impuls für 0,5 s
- 4 = Auf Impuls warten
- 5 = Impuls ausgeben und auf Quittierung warten
Wiederhole nach 0,5 s
- 6 = Auf Impuls warten und Quittierung ausgeben

Anwendung 4 1
4 5

Erläuterung Die Impulsausgabe dient dazu, externe Geräte mit der Prozessorkarte zu verbinden. Die einzelnen Optionen (1 ... 6) bewirken eine reibungslose Steuerung mit geringem externem Hardware-Aufwand. Die Impulsausgabe-Option stellt einen über ein Reed-Relais gepufferten potentialfreien Ausgang zur Verfügung.



PAL-PC

```
#axis x;  
#input  
pulse wait;  
move 2(9000)
```

-
-
-

Programmierbeispiel

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"4 4":gosub 1000  
140 print#1,"0 200,9000":gosub 1000  
150 print#1,"9":gosub 1000  
160 print#1,"@0S":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



Beachten Sie, dass der Impulsausgang vor dem Ende des Programmes zurückgesetzt werden muss, da sonst die Prozessorkarte einen Startbefehl am Startastereingang erkennt und das gespeicherte Programm sofort erneut ausführt. Sollte dies geschehen, muss die Prozessorkarte mit dem Not-Aus-Taster gestoppt oder ausgeschaltet werden. Der Stopp-Taster wird ignoriert solange der Impulsausgang gesetzt ist.

3.1.11 Befehl: Zeitverzögerung

Zweck	Die Prozessorkarte wartet die angegebene Zeit, bevor der nächste Programmschritt ausgeführt wird.	
Aufbau	5 <time> <time>	- Zahl zwischen 0 ... 32 767 (Angabe in 1/10 s)
Anwendung	5 40	(warte 4 s)
Erläuterung	-	



Programmierbeispiel

PAL-PC

```
#axis x;  
#input  
move 2(1000);  
wait 100;  
move -2(9000);  
stop.  
#start
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"0 200,1000":gosub 1000  
140 print#1,"5 100":gosub 1000  
150 print#1,"0 -200,9000":gosub 1000  
160 print#1,"9":gosub 1000  
170 print#1,"@0S":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



Eine Zeitverzögerung können Sie nicht durch Betätigen des Stopp-Tasters abbrechen.
Bei fehlerhafter Programmierung bricht die µP-Taste den Prozess ab.

3.1.12 Befehl: Bewegung bis Impuls

Zweck

Dieser Befehl ermöglicht, die relative Positionierung ohne eine exakte Weginformation vorzugeben (siehe 1. Hinweis unten). Dabei wird der Prozessorkarte ein maximaler Verfahrensweg vorgegeben, der durch einen externen Stopp-Impuls abgebrochen wird (z. B. Betätigen der Stopp-Taste). Anschließend wird der nächste Befehl des Datenfeldes abgearbeitet (siehe 2. Hinweis unten).

Aufbau

6 <Sx>, <Gx>, <Sy>, <Gy>, <Sz1>, <Gx1>, <Sz2>, <Gz2>
<Sx> = Schrittzahl X-Achse

• •
<Gz2> = Geschwindigkeit 2. Z-Achse

Erläuterung

Der Befehlsaufbau ist identisch mit dem Befehl *Bewegung relativ* (siehe 3.1.3 und 2.1.4)



Programmierbeispiel

PAL-PC

```
#axis x;  
#input  
movep 20(1000);  
movep -20(1000);  
stop.  
#start
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"6 200,1000":gosub 1000  
140 print#1,"6 -200,9000":gosub 1000  
150 print#1,"9":gosub 1000  
160 print#1,"@0S":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

info

Die Prozessorkarte prüft nicht, ob die Bewegung den zulässigen Verfahrbereich der angeschlossenen Mechanik verlässt (Endschalter könnten aktiviert werden).

info

Die Impulslänge des externen Tasters darf maximal 40 ms betragen. Falls diese Impulslänge nicht eingehalten werden kann, muss im Programmablauf als nachfolgender Befehl eine Zeitverzögerung programmiert werden, da sonst der anschließende Befehl übergangen wird.

3.1.13 Befehl: **Starte angeschlossene Interface-Karte**

Zweck Dieser Befehl ergänzt die durch Prozesssynchronisation mögliche gegenseitige Aktivierung von zwei Prozessorkarten um vier Optionen.

Aufbau 8 <GN> <Optionen> [<Achsen>]

<GN>	=	Gerätenummer der anzusprechenden Karte
<Option>=	R	Referenzfahrt auslösen, Ende abwarten
	S	Starte zweite Karte, Ende abwarten
	r	Referenzfahrt auslösen, eigene Ausführung fortsetzen
	s	Starte zweite Karte, eigene Ausführung fortsetzen
<Achsen>	=	Achsangabe für Referenzfahrt

Anwendung 8 0S
8 0R1

Erläuterung -



Programmierbeispiel

PAL-PC

```
#axis xy;  
#input  
repeat  
move 20(100),20(100);  
tell 0 reference x;  
move 20(100),20(100);  
until 0;  
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@03":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"0 20,100,20,100":gosub 1000  
140 print#1,"8 0R1":gosub 1000  
150 print#1,"0 -20,100,-10,100":gosub 1000  
160 print#1,"3 0,-4":gosub 1000  
170 print#1"9":gosub 1000  
1000 ...
```



Bei Verwendung der Optionen r und s muss sichergestellt werden, dass ein erneuter Befehl erst nach Bearbeitungsende des momentanen Befehls in der angesprochenen Prozessorkarte, gesendet wird.

3.2 Ergänzender Befehlsumfang Interface-Karte 5.0

3.2.1 Befehl: 3D-Linearinterpolation

Zweck	Die Interface-Karte 5.0 erweitert die 2,5D-Interpolation des Standard-Betriebssystems auf eine 3D-Interpolation (räumlich). Mit dem Befehl können Sie diese Interpolation gezielt ein- und ausschalten.
Aufbau	$z<STATUS>$ $<STATUS>$ = 0 - 3D-Interpolation aus = 1 - 3D-Interpolation ein
Erläuterung	Die Anweisung wirkt modal, d. h. alle move- und moveto-Anweisungen werden dreidimensional ausgeführt. Die Angabe von z2-Parametern in diesen Verfahrbewegungen wird ignoriert. Als Geschwindigkeitsangabe der Interpolation wird der Wert der X-Achse herangezogen.



Programmierbeispiel

PAL-PC

```
#axis xyz;  
reference xyz;  
set3don;  
move 10(700),15(800),3(400),  
      0(30);  
set3doff;  
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@07":gosub 1000  
120 print#1,"@0i":gosub1000  
130 print#1,"z1":gosub 1000  
140 print#1,"0100,700,150,800,30,400,0,30"  
145 gosub 1000  
150 print#@1,"z0":gosub1000  
160 print#1,"9":gosub 1000  
1000 ...
```



(nähere Erläuterungen siehe Befehl *3D-Linearinterpolation* Kapitel 2.2.1).

3.2.2 Befehl: Zirkularinterpolation

Zweck	Bearbeiten von Kreisen und Kreisbögen mit konstanter Bahngeschwindigkeit. Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle ausgelöst. Der erste Befehl legt die Kreisrichtung fest, im zweiten werden die Interpolationsparameter übergeben.	
Aufbau	Kreisrichtung	f1 Gegenuhrzeigersinn f0 Uhrzeigersinn
	Kreisbogen	y B , V, D, Xs, Ys, Rx, Ry
	B	Bogenlänge - sie gibt die Länge des Bogens zwischen Start- und Endwinkel des Kreissegmentes in Schritten an.
	V	Geschwindigkeit - sie legt die Positioniergeschwindigkeit während der Bearbeitung fest ($30 < V < 10\,000$).
	Rx Richtung X -	die Parameter Rx u. Ry zeigen der Prozessorkarte an,
	Ry Richtung Y	in welchem Quadranten des Kreises die Interpolationen startet.
	Xs Startpunkt X -	Xs und Ys geben den Startpunkt der Interpolation
	Ys Startpunkt Y	bezogen auf den Kreismittelpunkt an.
	D	Interpolationsparameter - Aufgrund der geringen Speicherkapazität erwartet die Prozessorkarte die Angabe des Quadranten, in dem der Kreisbogen beginnt und die tangentielle Richtung des zu beschreibenden Kreises.

Erläuterung (siehe Befehl *Zirkularinterpolation* Kapitel 2.2.2)



Programmierbeispiel

Nach einer relativen Positionierung von 150 mm (600 Schritte) soll ein Viertelkreis entgegen dem Uhrzeigersinn ausgeführt werden. Der Radius des Kreises ist mit 50 mm (200 Schritte) vorgegeben, der Startwinkel beträgt 0° der Endwinkel 90° . Die Geschwindigkeit über den gesamten Verlauf soll 200 Schritte/s betragen.

PAL-PC

```
#axis xyz;  
reference xyz;  
move 150(200),150(200),  
      15(800),0(21);  
circle_ccw50(300),0,90;  
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@07":gosub 1000  
120 print#1,"@0r7":gosub 1000  
130 print#1,"@0i":gosub 1000  
140 print#1,"0 600,200,600,200,15,800,0,21"  
145 gosub 1000  
150 print#1,"f1":gosub 1000  
160 print#1,"y6400,300,-400,800,-1,1,"":gosub1000  
170 print#1,"9":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```



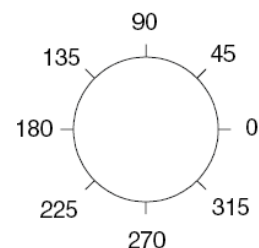
Bei Verwendung von PAL-PC wird die Berechnung der Parameter von der PC-Software übernommen. Somit beschränkt sich die Programmierung auf die Angabe von Radius, Verfahrensgeschwindigkeit des Kreissegm. sowie den Start- und Endwinkel der Kreisbahn.

Die Unterscheidung der Bewegungsrichtung geschieht durch den Befehl circle_cw → Bewegung im Uhrzeigersinn und circle_ccw → Bewegung entgegen dem Uhrzeigersinn (siehe auch Beschreibung PAL-PC).



Beispiel: Kreisinterpolation in PAL-PC

Es sei ein Kreis mit einem Radius von 20 mm gegeben, als Arbeitsgeschwindigkeit werden 5 000 Hz angenommen. Nachfolgende Befehlszeilen zeigen die Programmierung bei unterschiedlichen Start- und Stoppwinkeln in positiver Richtung (gegen den Uhrzeigersinn).



circle_ccw 20(5000),0,360;	Vollkreis	Beginn und Ende bei 0°
circle_ccw 20(5000),0,45;	Kreisabschnitt	Beginn bei 0° und Ende bei 45°
circle_ccw 20(5000),45,225;	Kreisabschnitt	Beginn bei 45° und Ende bei 225°
circle_ccw 20(5000),225,585;	Vollkreis	Beginn und Ende bei 225°

Bei Bewegung in negativer Richtung (im Uhrzeigersinn) achten Sie bitte darauf, dass der Startwinkel immer größer als der Stoppwinkel ist. Gegebenenfalls müssen Sie zum Startwinkel den Wert 360° (Vollkreis) addieren.

circle_cw 20(5000),360,0;	Vollkreis	Beginn und Ende bei 0°
circle_cw 20(5000),360,45;	Kreisabschnitt	Beginn bei 0° und Ende bei 45°
circle_cw 20(5000),405,225;	Kreisabschnitt	Beginn bei 45° und Ende bei 225°
circle_cw 20(5000),585,225;	Vollkreis	Beginn und Ende bei 225°

3.3 Ergänzender Befehlsumfang bei Interface-Karten mit E/A-Erweiterung

3.3.1 Befehl: Ausgangsport setzen

Zweck Die Prozessorkarte setzt am definierten Ausgangsport der E/A-Erweiterungseinheit ein gewünschtes Ausgangsmuster.

Aufbau p<Adresse>, <BITNR>, <Wert>

<Adresse> = Ausgangsport, 1 —> 65 529
= Ausgangsport, 2 —> 65 530
<BITNR> = bitweise setzen, 1 <[BITNR]> 8
= byteweise setzen, 128
<Wert> = 0 ... 255

Erläuterung Als <WERT> ist ein Zahlenwert einzugeben, der je nach <BITNR> die entsprechenden Ausgänge einzeln beschreibt oder bei der byteweisen Verarbeitung das Ausgangsmuster des kompletten Ports setzt.

1. bitweise setzen

Die Bitnummer bestimmt, welches Ausgangsbit bearbeitet wird; der Wert legt den Betriebszustand des Bits fest.

Befehl	Ausgangsport	Bit	Zustand
p65529,5,0	Port I	5	aus
p65529,4,1	Port I	4	ein
p65530,1,1	Port II	1	ein

2. byteweise setzen

Bei der byteweisen Bearbeitung des Ausgangsports legt der <WERT> das Bitmuster des kompletten Ausganges fest.

Befehl	Ausgangsport	Dualmuster
p65529,128,0	Port I	00000000
P65529,128,27	Port I	00011011
p65530,128,205	Port II	11001101
p65530,128,255	Port II	11111111



Programmierbeispiel

PAL-PC

```
#axis x;  
reference x;  
set_port 65529,5=0;  
set_port 65530,128=27;  
stop.  
#start  
•
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"p 65529,5,0":gosub 1000  
140 print#1,"p 65530,128,27":gosub 1000  
150 print#1,"9":gosub 1000  
160 print#1,"@0S":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

info

Die Verarbeitung der Signalausgänge wird innerhalb der Prozessorkarte ablaufgesteuert durchgeführt. Somit ist ein Setzen bzw. Löschen von Ausgängen während einer Befehlsbearbeitung z. B. einer Positionierbewegung nicht möglich.

Bei Ausfall der Versorgungsspannung der Prozessorkarte schalten alle Ausgänge auf inaktiv. In Verbindung mit den open-Kollektor-Ausgängen der E/A-Erweiterungseinheit ergeben sich somit folgende Signalzustände:

Ausgang 1

Alle Ausgänge sind aktiv low

Durch Opto-Isolierung der Ausgänge und Einsatz einer externen Spannungsversorgung sind die Endstufen-Transistoren bei fehlender Ansteuerung durchgeschaltet, das Spannungspotential am Kollektor-Ausgang beträgt 1,0 V (VCEsat). Eine zwischen einem Ausgang und +Vs ext. angeschlossene Lampe leuchtet.

Ausgang 2

Alle Ausgänge sind inaktiv

Bei fehlender Steuerspannung am Eingang der Endstufen-Transistoren sind die Ausgänge offen, d. h. eine zwischen einem Ausgang und +Vs angeschlossene Lampe leuchtet nicht.

3.3.2 Befehl: Eingangsport setzen

Zweck Die Prozessorkarte liest das anstehende Bitmuster am Eingangsport der E/A-Erweiterung ein.

Aufbau o<Adresse>,<BITNR>,<Wert>,<Offset>
 <GN> = Gerätenummer, Standard = 0
 <Adresse> = Eingangsport 65 531
 <BITNR> = bitweise lesen, 1 < [BITNR] > 8
 = byteweise lesen, 128
 <Wert> = zu vergleichender Wert
 <Offset> = ist die Anzahl der Programmschritte, um die eine Verzweigung vor- oder rückwärts erfolgen soll.
 Dabei wird der Eingangsport auf das im Parameter <WERT> festgelegte Bitmuster überprüft und bei Erfüllung der Bedingung die Verzweigung durchgeführt.

Erläuterung Durch Angabe des Parameters <BITNR> unterscheidet das Betriebssystem zwischen einer bit- oder byteweisen Verarbeitung des Eingangsports.

1. bitweise lesen

Die Bitnummer bestimmt, welches Eingangsbit abgefragt wird.

Befehl	Abfragekriterium	Verzweigung
o65531,2,0,3	Bit 2 = aus	3 Zeilen vorwärts
o65531,8,1,-2	Bit 8 = ein	2 Zeilen zurück

2. byteweise lesen

Bei der byteweisen Bearbeitung der Signaleingänge wird das Bitmuster des kompletten Ports abgefragt.

Befehl	Abfragekriterium	Verzweigung
o65531,128,10,3	Dual 00001010	3 Zeilen vorwärts
o65531,128,0,-2	Dual 00000000	2 Zeilen rückwärts



Programmierbeispiel

PAL-PC

```
#axis x;  
reference x;  
on_port 65531,2=0,3;  
on_port 65531,8=1,-2;  
set_port 65530,1=1;  
move 100(2000);  
set_port 65530,2=1;  
move -100(2000);  
stop.
```

GW-BASIC

```
100 open"com1:9600,N,8,1,DS,CD"as #1  
110 print#1,"@01":gosub 1000  
120 print#1,"@0i":gosub 1000  
130 print#1,"71":gosub 1000  
140 print#1,"o 65531,2,0,3":gosub 1000  
150 print#1,"o 65531,8,1,-2":gosub 1000  
160 print#1,"p 65530,1,1":gosub 1000  
170 print#1,"0 400,2000":gosub 1000  
180 print#1,"p 65530,2,1":gosub 1000  
190 print#1,"0 -400,2000":gosub 1000  
200 print#1,"9":gosub 1000  
1000 if loc(1)<1 then goto 1000  
1010 a$=input$(1,1)  
1015 if a$="0" then return  
1020 print "Karte meldet Fehler : ";a$  
1030 stop
```

info

Die Signaleingänge sind optoisolierte Eingänge. Durch integrierte Vorwiderstände an den Anoden der Optokoppler ist eine einfache Masse-Beschaltung der Kathoden-Eingänge ausreichend, um das Eingangsbit zu setzen.

Die Information der Signaleingänge wird auf der E/A-Erweiterungseinheit nicht zwischengespeichert. Somit gehen impulsartige Eingangssignale, die während der internen Verarbeitung eines Datensatzes anfallen, verloren.

3.4 Ergänzender Befehl in Verbindung mit einer Programmwahleinheit

3.4.1 Befehl: Tastaturabfrage

Zweck Die Prozessorkarte fragt an der seriellen Schnittstelle den Code einer betätigten Taste der Programmwahleinheit ab. Entsprechend der empfangenen Information wird ein im Programm definierter Offset durchgeführt.

Aufbau	k <Tastenummer>,<Offset>	
	<Tastenummer>	= Zahl zwischen 1 und 20 (255)
	<Offset>	= Anzahl der Programmschritte, um die eine Verzweigung erfolgen soll
Anwendung	k13,0 k1,2	Warte, bis eine Taste gedrückt ist überspringe nächsten Befehl, falls Taste 1 gedrückt
Erläuterung	Der Befehl k veranlasst den Prozessor, an der seriellen Schnittstelle eine Impulskette auszugeben und auf eine Rückinformation der angekoppelten Programmwahleinheit zu warten. Der empfangene Code bewirkt eine im Datenfeld vorgesehene Verzweigung des Programmablaufes. Wird keine Taste betätigt, ergibt sich als rückgesendete Information <Gesamtanzahl Tasten> +1 (Programmwaheinheit mit 20 Taste —> 21).	



Programmierbeispiel

PAL-PC

```
#axis x;
Anfang:
    repeat
        on_key 1, do_move;
        on_key 2, do_reference;
goto Anfang;
do_move:
    move 100(2000);
    move -100(2000);
goto Anfang;
do_reference:
    reference x;
goto Anfang;
stop.
```

GW-BASIC

```
100 open "com1:9600,N,8,1,DS,CD"as#1
110 print#1,"@01":gosub 1000
120 print#1,"@0i":gosub 1000
130 print#1,"k1,4":gosub 1000
140 print#1,"k2,5":gosub 1000
150 print#1,"k3,6":gosub 1000
160 print#1,"3 0,-3":gosub 1000
170 print#1,"0 1000,1000":gosub 1000
180 print#1,"3 0,-5":gosub 1000
190 print#1,"71":gosub 1000
200 print#1,"3 0,-7":gosub 1000
210 print#1,"9":gosub 1000
220 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "Karte meldet Fehler:";a$
1030 stop
```

info

Die Programmwaheinheit ermöglicht keine direkte Programmierung der Prozessorkarte, sie kann nur eine vorher definierte Verzweigung im Programmablauf bewirken.

In Verbindung mit der Software PAL-PC ist eine Programmierung durch "Labelzuweisung" wesentlich vereinfacht. Sie können einzelne Programmteile durch ein Label kennzeichnen und durch Betätigen der Taste mit demselben Label aktivieren.

4 Fehlermeldungen

4.1 Fehlermeldungen *isel*-Prozesskarten

Fehlerbeschreibung	Ursachen	Beseitigung
0 Kein Fehler	Befehl wurde korrekt ausgeführt oder gespeichert	
1. Fehler in übergeordneter Zahl. Die Interface-Karte hat eine Zahlenangabe empfangen, die nicht korrekt interpretiert werden konnte.	<ul style="list-style-type: none"> Der übergebene Zahlenwert ist außerhalb des zulässigen Bereiches. Für 8-Bit-Angaben beträgt der zulässige Bereich - 128 bis + 127, für 16-Bit-Angaben - 32 768 bis + 32 767, für 24-Bit-Angaben - 8 388 608 bis + 8 388 607. Der übergebene Zahlenwert enthält unzulässige Zeichen. 	Lassen Sie alle Ausgaben, die zur Interface-Karte gehen, protokollieren. Prüfen Sie anschließend an der Stelle, an der der Fehler auftritt, ob alle Wertangaben für den übertragenen Befehl korrekt sind.
2. Endschalter-Fehler	<p>Durch die Verfahrbewegung wurde ein Endschalter angesprochen. Die Schrittausgabe wurde gestoppt. Die Interface-Karte hat keine korrekte Sollposition mehr (Schrittverlust). Falls ein Programm ausgeführt wurde, wird dieses gestoppt.</p> <p>Die Referenzfahrt einer Schrittmotorachse wurde nicht korrekt ausgeführt.</p>	Die Achse, die im Endschalter steht, müssen Sie von Hand aus dem Endschalterbereich bewegen. Anschließend sollten Sie die Fehlerursache prüfen (falsche Verfahrwege, Überlastung der Anlage mit resultierendem Schrittverlust, Schwergängigkeit der Mechanik o. ä.) und nach deren Beseitigung das Programm neu starten. Neue Referenzfahrt durchführen.
3. Unzulässige Achsangabe	Der Interface-Karte wurde eine Achsangabe für einen auszuführenden oder einen zu speichernden Befehl übermittelt, die eine nicht definierte Achse enthält.	Verwenden Sie in den Befehlen nur Werte gemäß Kapitel 1.1.1 Befehl <i>Achsenanzahl</i> setzen.

Fehlerbeschreibung	Ursachen	Beseitigung
4. Keine Achsen definiert	<p>Bevor der Interface-Karte speicherbare Befehle, Bewegungen oder allgemeine Befehle, die eine von der Achsenanzahl abhängige Anzahl v. Parametern haben, übergeben werden, muss der Befehl 'Achsenanzahl setzen' übergeben werden, um die internen Kartenparameter korrekt zu setzen. Die Achsenzahl bleibt nach dem Ausschalten erhalten, falls die Option "Batterie-Backup" verwendet wird. Sollte dennoch ein Fehler 4 auftreten, kann dies auf Batterieprobleme hinweisen.</p>	
5. Syntax-Fehler	<ul style="list-style-type: none"> • Ein Befehl wurde in Großschreibung verwendet, obwohl dieser Befehl nur in Kleinschreibung existiert. • Während der Übertragung eines Datenfeldes wurde versucht, einen speicherbaren Befehl zu verwenden. • Der verwendete Befehl existiert nicht. 	<p>Lassen Sie alle Ausgaben, die zur Interface-Karte gehen, protokollieren. Prüfen Sie anschließend an der Stelle, an der der Fehler auftritt, ob alle übertragenen Befehle korrekt sind.</p>
6. Speicherende	<p>Es wurde versucht, mehr Befehle zu übertragen, als die Interface-Karte speichern kann.</p>	<p>Teilen Sie das Programm in kleinere Abschnitte auf, übertragen Sie jeweils einen Abschnitt, führen Sie diesen aus und übertragen Sie dann den nächsten Abschnitt.</p>
7. Unzulässige Parameterzahl	<p>Die Interface-Karte hat mehr oder weniger Parameter für den Befehl erhalten, als benötigt werden.</p>	<p>Prüfen Sie, ob die Anzahl der Parameter für den Befehl in Verbindung mit der Anzahl der Achsen korrekt ist. Berücksichtigen Sie hierbei die z2-Bewegung.</p>
8. Zu speichernder Befehl nicht korrekt	<p>Der Interface-Karte wurde ein Befehl zur Speicherung übergeben, der in dieser Form nicht vorhanden ist.</p>	<p>Prüfen Sie den übertragenen Befehl. Existiert der Befehlscode? Ist Groß-/Kleinschreibung berücksichtigt worden?</p>

Fehlerbeschreibung	Ursachen	
A Impulsfehler	Die übergebene Option für den Befehl "Impuls" ist nicht im zulässigen Bereich zwischen 1 und 6.	
B Tell-Fehler	Die Tell-Funktion (starte zweite Interface-Karte) hat kein Endezeichen nach der Höchstzahl der zu sendenden Zeichen gefunden. Dieser Fehler deutet auf Speicherprobleme hin, da die Eingabe des Tell-Befehles immer ein Endezeichen anfügt.	
C (CR) erwartet	Die Interface-Karte hat auf das (CR)-Zeichen als Befehlsende gewartet. Sie haben jedoch ein anderes Zeichen übergeben. Dies ist hauptsächlich ein Problem der Parameteranzahl. Sie versuchen mehr Parameter zu übertragen, als für die Funktion notwendig sind.	
D Unzulässige Geschwindigkeit	Beachten Sie, dass auch für Achsen ohne Bewegung eine zulässige Geschwindigkeit benötigt wird, d. h. 0,0 als Wertepaar ist nicht zulässig.	
E Schleifenfehler	Es wurde versucht, eine Vorwärtsschleife auszuführen. Beachten Sie, dass bei Schleifen immer die letzten n-Befehle wiederholt werden, d. h. 34,4 ist unzulässig.	

Fehlerbeschreibung	Ursachen	
F BenutzerStopp	Sie haben Stopp an der Interface-Karte gedrückt. Die Befehlsausführung können Sie mit der Starttaste oder dem Befehl @0s wieder aufnehmen. Nach einem Stopp sind die Befehle @0P und @0Q zulässig! (@0P, um die erreichte Position abzufragen und @0Q, um die Ausführung endgültig abubrechen). Fehler F fällt aus dem üblichen Software-Handshake-Protokoll. Ein zusätzliches F kann jederzeit auftreten, falls Sie an der Interface-Karte während einer Verfahrbewegung den Stopp-Taster drücken. Um diesem Verhalten Rechnung zu tragen, sollte die Unteroutine der Software, die die Rückmeldung verarbeitet, mit der Behandlung dieses Sonderfalles ergänzt werden.	
= (cr)-Fehler	Die Interface-Karte hat ein (CR) empfangen, obwohl noch weitere Parameter für den aktuellen Befehl erwartet wurden.	

In Ergänzung zur Fehlerart "Referenzschalter angefahren" stellt das Betriebssystem der Prozessoreinheiten ab Produktionsdatum 10/93 eine zusätzliche Fehlerart mit diesem Syntax (Fehler 2) dar.

Es handelt sich hierbei um eine inkorrekt ausgeführte Referenzfahrt, d. h. eine Referenzfahrt-Bewegung, die nicht durch einen Interrupt des Referenzschalters abgebrochen wurde (der Referenzschalter wurde beim An- oder Freifahren nicht erkannt), vgl. "Ausführung einer Referenzfahrt", siehe Befehl Referenzfahrt Seite 7 bzw. 41.

In den verschiedenen Betriebsmodi der Prozessorkarte führt der Fehler zu folgenden Funktionen:

Mode	Start durch	Ergebnis
DNC-Mode	@0R7 @0r7	als Rückmeldung wird Fehlercode 2 ausgegeben zusätzlicher Fehlercode 2 am Ende der Bewegung
CNC-Mode	@0S @0s Start-Taste	als Rückmeldung wird Fehlercode 2 ausgegeben Die Ausführung des Programmes wird abgebrochen zusätzlicher Fehlercode 2 am Ende der Bewegung Die Ausführung des Programmes wird abgebrochen Die Ausführung des Programms wird abgebrochen

4.2 Fehlermeldungen PAL-PC

1	unexpected End of File	unerwartetes Fileende
2	' ; ' expected	' ; ' erwartet
3	illegal axis-entry	falsche Achsangabe
4	'x', 'xy', 'xz' or 'xyz' expected	'x', 'xy', 'xz' oder 'xyz' erwartet
5	axis already defined	Achsen schon definiert
6	'mm', 'cm', 'zoll', 'zoll/10' or 'zoll/20' expected	'mm', 'cm', 'zoll', 'zoll/10' oder 'zoll/20' erwartet
7	missing 'stop.', stop assumed	'stop.' fehlt, stop. ergänzt
8	input already active	Inputbefehl schon aktiv
9	too much nested repeats (Limit is 20)	zu viele geschachtelte Wiederholungen (maximal 20)
10	repeat without until detected	repeat ohne until gefunden
11	#-command not recognized	#-Befehl ist unbekannt
12	duplicate axis entry in command	doppelte Achsangabe in Befehl
13	'x', 'y' or 'z' expected	'x', 'y' oder 'z' erwartet
14	integer expected	Integer erwartet
15	' , ' expected	' , ' erwartet
16	positive integer expected	Positive Integergröße erwartet
17	until without repeat	until ohne repeat
18	real number expected	Realgröße erwartet
19	positive real number expected	Positive Realgröße erwartet
20	missing '#input'	vermisse '#input'
21	' (' expected	' (' erwartet
22	') ' expected	') ' erwartet
23	' . ' expected	' . ' erwartet
24	too much definitions	zu viele Definitionen (maximal 50)
25	definition name expected	Name für Definition erwartet
26	illegal character for send or wait(number between 1..126 expected)	illegales Zeichen für send oder wait (Zeichen zwischen 1 ... 126 erwartet)
27	' " ' or unit number expected	' " ' oder Gerätenummer erwartet
28	' " ' expected	' " ' erwartet
29	' wait ' expected	' wait ' erwartet
30	unit entry expected	Gerätenummer erwartet
31	command not recognized	Befehl wird nicht unterstützt
32	too much label definitions	zu viele Marken definiert (maximal 50)

33	positive integer between 1 and 126 expected	Positive Integergröße zwischen 1 und 126 erwartet
34	label not found	Marke nicht gefunden
35	no label definition in text	keine Markendefinition im Text
36	' , ' or 'times' expected	' , ' oder 'times' erwartet
37	'in' or 'out' expected	'in' oder 'out' erwartet
38	'on', 'off', 'in', 'out' or 'sync' expected	'on', 'off', 'in', 'out' or 'sync' erwartet
39	end of remark missing	Ende einer Bemerkung nicht gefunden
40	serial transmission error (time out in receive)	Übertragungsfehler (time out beim Empfangen)
41	elevation must be > 0.001	Steigung muss größer 0,001 sein
42	file not found	File nicht gefunden
43	letter or ' _ ' expected	Buchstabe oder ' _ ' erwartet
44	replace text exceeds 250 chars	Textersatz zu lang (max. 250 Zeichen)
45	line exceeds 250 chars after replace of definition	Zeile ist nach Textersatz länger als 250 Zeichen
46	illegal definition occurred	nicht erlaubte Definition
47	' " ' or ' < ' expected	' " ' oder ' < ' erwartet
48	' " ' expected	' " ' erwartet
49	' > ' expected	' > ' erwartet
50	include file not found or i/o error	Includefile nicht gefunden o. E/A-Fehler
51	i/o error on read	E/A-Fehler beim Lesen
53	illegal unit-no	unerlaubte Gerätenummer
54	'xy', 'xz' or 'yz' expected	'xy', 'xz' oder 'yz' erwartet
55	positive real number expected	Positive Realgröße erwartet
56	no matching definition for redefine	keine gültige Definition für Redefinition
57	'*' expected	'*' erwartet
58	forward loop not allowed	Schleife mit positiv. Offset nicht erlaubt
59	'=' expected	'=' erwartet
60	GUZ or UZ expected	GUZ oder UZ erwartet
61	starting angle must be less ending angle	Startwinkel muss < als Endwinkel sein
62	starting angle must be greater ending angle	Startwinkel muss > als Endwinkel sein
63	Zero circle not allowed	Bögen mit Längen 0 nicht erlaubt
149	invalid number (Interface)	Fehler in übergebener Zahl (Interface)
150	reference switch (Interface)	Endschalter (Interface)

151	invalid axis (Interface)	unzulässige Achsangabe (Interface)
152	no axis information (Interface)	keine Achsen definiert (Interface)
153	syntax error (Interface)	Syntax-Fehler (Interface)
154	out of memory (Interface)	Speicherende (Interface)
155	invalid number of parameters (Interface)	unzulässige Parameterzahl (Interface)
156	incorrect command (Interface)	Befehl inkorrekt (Interface)
161	(cr) error	(cr)-Fehler (Interface)
164	self test not terminated or cable error	Selbsttest nicht abgeschlossen oder Übertragungsfehler (Interface)
165	pulse error (Interface)	Impulsfehler (Interface)
166	tell error (Interface)	Tellfehler (Interface)
167	(cr) expected (Interface)	(cr) erwartet (Interface)
168	invalid speed (Interface)	unzulässige Geschwindigkeit (Interface)
169	loop error (Interface)	Schleifenfehler (Interface)
170	user stop (Interface)	BenutzerStopp (Interface)
100...199	Interface-Card error (100+Error)	Fehlermeldungen der Interface-Karte 100+Fehler)