isel®
From Components to Systems

isel-Integrierte Steuerung IT

www.isel.com

Hardware Beschreibung

IT 116
IT 116 N
IT 116 G

isel®

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ (06659)981-0 (06659)981-776

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.

isel Germany AG Maschinen und Controller sind CE-konform und entsprechend gekennzeichnet. Für alle sonstigen Maschinenteile und -komponenten, auf die CE-Sicherheitsrichtlinien anzuwenden sind, ist die Inbetriebnahme solange untersagt, bis alle entsprechenden Anforderungen erfüllt sind.

Die Firma isel Germany AG übernimmt keine Gewähr, sobald Sie irgendwelche
Veränderungen an der Maschine vornehmen.

Der EMV-Test gilt nur für die ab Werk gelieferte Originalkonfiguration der Maschine.

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: 08/2008 OR

Inhaltsverzeichnis

1 Einleitung 5

1.1 Sicherheitssymbole 6

1.2 Sicherheitshinweise 6

2 Lieferumfang 8

3 Hardwarebeschreibung 9

3.1 Technische Daten 9

3.2 Bedien-/Anzeigenelemente der IT 116 10

3.3 Serielle Schnittstellen 11

3.4 Funktionselemente der IT 116 13

3.5 Einstellung DIP-Schalter S1 13

3.5.1 Baudrateneinstellung 13
3.5.2 Einstellung der Beschleunigung 14
3.5.3 Einstellung Voll-/Halbschrittbetrieb 14
3.5.4 Schalter S1.6, S1.7, S1.8 14

3.6 Programmiermodus 15

3.7 Spannungsversorgung 15

3.8 Anschluss und Inbetriebnahme 16

3.8.1 Start 16
3.8.2 Stop 16
3.8.3 µP- Reset 16
3.8.4 Referenz-/Endlagenschalter 16
3.8.5 Endstufenabschaltung (Disable) 18
3.8.6 Signaleingänge In1-In4 19
3.8.7 Signalausgang Out1- Out 20
3.8.8 Betriebsstromeinstellung (IPhase) 21
3.8.9 Schrittmotorausgang 22
3.8.10 Datenspeicher 22

3.9 Betriebsstörungen 23

3.10 Steckverbinder 25

3.10.1 IT 116 (Euro-Einschub) 25
3.10.2 IT 116 N (Euro-Kassetteneinschub) 26
3.10.3 IT 116 G (Euro-Profilgehäuse) 27

4 Inkompatibilität 29

4.1 Positionierung 29

4.2 Signalein-/ausgänge 30

4.3 Diagnose 30

Abbildungsverzeichnis

Bild 1: Anschluss serielle Schnittstelle 12
Bild 2: Die Funktions- und Bedienelemente 13
Bild 3: Anschluss Referenzschalter 17
Bild 4: Signaleingang Disable 18
Bild 5: Aufbau der Signaleingänge In 1 - In 4 19
Bild 6: Aufbau der Signaleingänge Out 1 - Out 4 20
Bild 7: Anschlüsse der Rückseite 27

1 Einleitung

IT steht bei **isel Germany AG** für integrierte Technologie, den logischen Zusammenschluss von Leistungselektronik und Mikroprozessortechnik. Die Schrittmotorantriebe IT 116, IT 116 G sowie IT 116 N sind mikroprozessorgesteuerte Ansteuereinheiten für 2(4)-Phasen-Schrittmotoren mit zusätzlichen SPS-Schaltfunktionen. Die Antriebsmodule sind Flachbaugruppen (100 x 160 mm), auf der sowohl die Funktionselemente einer intelligenten Schrittmotorsteuerung als auch die kompletten Elemente einer Schrittmotor-Leistungsendstufe integriert sind. Der Prozessorteil besteht aus einem 8-Bit-Mikrocontroller mit 32 kB-Datenspeicher und 32 kB-Betriebseprom.
Durch Implementierung des *isel*-CNC-Betriebssystemes 5.0 steht der umfangreiche und erprobte Befehlsumfang der Interfacekarten-Serie zur Verfügung. Zur Behandlung von E/A-Funktionen verfügt die Einheit über je vier optoisolierte Eingänge und Open-Collector-Ausgänge. Die Leistungselektronik basiert auf einer Schrittmotor-Steuerkarte. Sie verfügt über DMOS-Leistungsendstufen, die durch gemischte bipolar CMOS/DMOS-Technologie einen sehr guten Wirkungsgrad und eine hohe Schaltfrequenz aufweisen. Eine Stromregelung nach dem Konstantstromprinzip ermöglicht ein optimales Laufverhalten des Schrittmotors. Hierbei wird durch Anlegen einer hohen Spannung ein schneller Stromanstieg innerhalb der Motorspulen vorgegeben. Bei Erreichen des eingestellten Betriebsstromes wird die Endstufe abgeschaltet und durch einen periodischen Chopper wieder aktiviert. Dies gewährleistet ein optimales Laufverhalten des Schrittmotors bei gleichzeitiger geringer Verlustleistung der Endstufe. Der Betriebsstrom des Schrittmotors wird mit Hilfe eines seitlich montierten Trimmpotentiometers (I_{Phase}) bis max. 3,5 A eingestellt.
Um einer Zerstörung der Leistungsendstufe im Kurzschlussfall Phase - Phase, Phase -Masse sowie Phase - Vcc vorzubeugen, wird im Fehlerfall die Endstufe abgeschaltet. Zur Überwachung der Kühlflächentemperatur verfügt die Einheit über einen Temperaturwächter, der im Ansprechfall die Endstufe abschaltet und nach Absinken unterhalb seines Schwellwertes selbsttätig wieder aktiviert. Darüberhinaus werden fehlerhafte Versorgungsspannungen durch entsprechende Schutzschaltungen erfasst und führen zum Abbruch der Bearbeitung.

Die Schrittmotor-Antriebseinheiten sind in drei Ausführungen verfügbar:

als Euro-Einsteckkarte (IT 116)
- einbaufertig zum Einsatz in 19"-Baugruppenträger
- externe Spannungsversorgung für Logik und Motorspannung

als Euro-Kassetteneinschub (IT 116 N)
- einbaufertig zum Einsatz in 19"-Baugruppenträger
- integriertes 90 VA-Netzteilmodul

als Euro-Profilgehäuse (IT 116 G)
- Kompletter Schrittmotorantrieb im Aluminium-Profilgehäuse
- Schraub-Klemm-Steckverbinder für Signalein-/-ausgänge
- 9pol. Sub D-Buchsenstecker als Schrittmotorausgang
- Netzanschlussleitung

1.1 Sicherheitssymbole

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.

1.2 Sicherheitshinweise

- Die Schrittmotorantriebe der Serie IT 116 sind nach dem aktuellen Stand der Technik und den anerkannten sicherheitstechnischen Regeln aufgebaut.
- Betrieben werden darf das Gerät nur im einwandfreien technischen Zustand. Störungen sind umgehend zu beseitigen. Kinder und nicht eingewiesene Personen dürfen das Gerät nicht in Betrieb nehmen.
- Das Gerät darf nur für die bestimmungsgemäße Verwendung eingesetzt werden.
- Alle Arbeiten sind ausschließlich von autorisiertem Fachpersonal und unter Berücksichtigung der Vorschriften der Elektroindustrie sowie der Unfallverhütungsvorschriften durchzuführen.
- Montage und Einsatz der Betriebsmittel ist entsprechend den Normen der Konformitätserklärung durchzuführen. Die vom Hersteller eingehaltenen Vorschriften und Grenzwerte schützen nicht bei unsachgemäßem Gebrauch der Betriebsmittel.
- Das Gerät darf nicht hoher Luftfeuchtigkeit und hohen Vibrationen ausgesetzt werden.
- Bewahren Sie diese Bedienungsanleitung sorgfältig auf und verpflichten Sie jeden Benutzer auf Ihre Einhaltung!
- Die Nichtbeachtung dieser Bedienungsanleitung kann Sachschäden, schwere Körperverletzungen und den Tod zur Folge haben.

isel-Schrittmotorantriebe der Serie IT 116 werden nach den geltenden Vorschriften der VDE 0160 gefertigt!

- Zur Netztrennung der Steuerungen IT 116 N und IT 116 G werden Ringkern-Transformatoren nach VDE 0551 eingesetzt.

- Die Funkentstörung nach VDE 0875 Teil 11 (EN 55011) ist in Vorbereitung; Entsprechende Filter müssen Sie bei den Antriebseinheiten IT 116 N und IT 116 dem Netzeingang vorschalten; in der geschlossenen Schrittmotorsteuerung IT 116 G wird der Filter integriert.

Beachten Sie
-Vor dem Öffnen der Antriebssteuerung IT 116 G Netzstecker ziehen!

-Die Schrittmotorantriebe dürfen Sie nur im spannungslosen Zustand der Anlage in 19"-Einbausysteme ein- und ausbauen.

-Ebenso dürfen Sie die Schrittmotoren und Signalein-/-ausgänge nur im spannungslosen Zustand ankoppeln.

2 Lieferumfang

Zum Lieferumfang der einzelnen Steuereinheiten gehören folgende Komponenten:

IT 116 N (Art.-Nr. 381 010)

- Schrittmotorantriebseinheit in 22 TE-Kassettengehäuse, inkl. Netzteil
- Gegenstecker Steuerungsseite (DIN 41612 F24/H7 Buchsenstecker, Lötpin)
- Gegenstecker Netzteilseite (DIN 41612 H15 Buchsenstecker, Lötpin)
- serielles Schnittstellenkabel für IBM-kompatible Rechner (9pol. Sub D-Buchsenstecker -- 9pol. Sub D-Buchsenstecker)
- Bedienungsanleitung (Art.-Nr. 395 052)

IT 116 (Art.-Nr. 381 011)

- Schrittmotorantriebseinheit als 10 TE-Euro-Einschub (ohne Netzteil)
- Gegenstecker (DIN 41612 F24/H7 Buchsenstecker, Lötpin)
- serielles Schnittstellenkabel für IBM-kompatible Rechner (9pol. Sub D-Buchsenstecker -- 9pol. Sub D-Buchsenstecker)
- Bedienungsanleitung (Art.-Nr. 395 052)

IT 116 G (Art.-Nr. 381 012)

- Schrittmotorantriebseinheit in Euro-Gehäuse, inkl. Netzteil
- 2 x Gegenstecker 5pol. (z. B. Fab. Phönix Combicon)
- Gegenstecker 10pol. (z. B. Fab. Phönix Combicon)
- serielles Schnittstellenkabel für IBM-kompatible Rechner (9pol. Sub D-Buchsenstecker -- 9pol. Sub D-Buchsenstecker)
- Verbindungsleitung 3 m, Schrittmotorausgang zu Schrittmotor (9pol. Sub. D-Stiftstecker -- 9pol. Sub D-Buchsenstecker)
- Netzzuleitung 1,5 m
- Bedienungsanleitung (Art.-Nr. 395 052)

3 Hardwarebeschreibung

3.1 Technische Daten

Gerätetyp/Aufbau IT IT 116 IT 116 N IT 116 G	 19"-Euro-Einsteckkarte (10 TE-Frontplatte) 19"-Euro-Kassetteneinschub (22 TE- Frontplatte) Euro-Profilgehäuse l = 200 mm, b = 105 mm, h = 111mm
Spannungsversorgung Logikspannung Motorspannung Betriebsstrom	 + 5 V, 300 mA (siehe 1.7 Spannungsversorgung) + 24…+ 44 V (siehe 1.7 Spannungsversorgung) max. 3,5 A
Steckverbinder	siehe Kapitel 4.10
Signaleingänge	StartStop µP-Reset Endstufenabschaltung “Disable” Referenzschalter 4 optoisolierte Signaleingänge
Signalausgänge	Open-Collector-Transistor Schaltausgänge (4 Relaisausgänge bei Variante IT 116 G)
Schrittmotorausgang	siehe Kapitel 4.8.9 bzw. Kapitel 4.10
Bedienelemente	Phasenstrompotentiometer Start-Taste Stop-Taste µP-Reset-Tast-Schalter Diagnose-Tast-Schalter
Anzeigenelemente	Betriebs-LED Diagnose-LED Status: Signalein-/-ausgänge

3.2 Bedien-/Anzeigenelemente der IT 116

1 Betriebs-LED Leuchtet bei Betriebsbereitschaft der Steuereinheit auf

2 Diagnose-LED Leuchtet, wenn eine Betriebsstörung der Einheit vorliegt (siehe auch Kapitel 4.9 Betriebsstörungen).

3 µP-Reset-Tast-Schalter Unterbricht - bedingt durch einen Prozessor-Reset sofort alle Aktivitäten der IT 116. Hierbei werden alle Signalausgänge zurückgesetzt und die Schrittmotorimpulse abgebrochen. Eventuell auftretende Schrittfehler des Motors werden ignoriert. Durch Betätigen des µP-Reset-Tasters, bei gleichzeitig gedrückter Start-Taste, wird ein Selbsttest der Einheit eingeleitet.

4 Diagnose- Zur Abfrage des Betriebszustandes der Antriebseinheit. Tast-Schalter Bei einer Betriebsstörung bewirkt die Betätigung des Tasters einen internen Check mit anschließender Anzeige des Fehlercode an der Diagnose-LED (siehe Kapitel 4.9 Betriebsstörungen).

5 Start- Taste Startet die Ausführung eines im Datenspeicher abgelegten CNC-Datenfeldes. Der Startimpuls kann sowohl vom Start-Taster als auch durch ein externes aktiv-low-Signal ausgelöst werden.

6 Stop-Taste Unterbricht die Ausführung einer programmierten Bewegung durch Einleiten einer Bremsrampe. Der unterbrochene Prozess wird fortgesetzt mit der Start-Taste bzw. dem Befehl @0S.

7 Status-Anzeige Kontroll-LED zur Anzeige des momentanen Zustandes der Signal-ein-/-ausgänge (siehe auch Kapitel 4.8.6 und 4.8.7).

8 ser. Schnittstelle siehe Kapitel 4.3.

9 Phasenstrom Potentiometer zur stufenlosen Einstellung des Motorstromes

3.3 Serielle Schnittstellen

Zur Datenübertagung zwischen IT 116 und einem Steuerrechner wird eine serielle Schnittstelle (RS232) eingesetzt. Die Verbindung ist über eine 3-Draht-Leitung realisiert; ein Software-Protokoll ermöglicht die fehlerfreie Übertragung der ASCII-Zeichen. Dabei ist es jedoch notwendig, dass sich beide Systeme an das im Folgenden beschriebene Übertragungsprotokoll halten.

- Der angeschlossene Steuerrechner sendet einen Befehl, der mit einem Zeilenendezeichen [char(13)] abgeschlossen ist.

- Die Prozessoreinheit quittiert die Ausführung bzw. Speicherung des Befehles durch das Quittierungssignal 0 [char(48)] oder meldet einen aufgetretenen Fehler mit einem ASCII-Zeichen ungleich 0 (siehe auch im Handbuch CNC-Betriebssystem 5.0, Kapitel 2.1.1).

Als Datenübertragungsparameter sind auf der Antriebssteuerung IT 116 folgende Werte festgelegt:

9600 Baud (einstellbar)
8 Daten-Bit
1 Stop-Bit
keine Parität

Zur Überprüfung des korrekten Anschlusses bzw. der Funktion der seriellen Schnittstelle verfügt die Prozessorkarte über eine Selbsttestroutine. Sie wird ausgeführt, indem Sie die Start-Taste festhalten und den µP-Reset-Tast-Schalter betätigen. Die IT 116 überprüft daraufhin Ihren Speicherbereich, sowie die Schalterstellung des 8-fach-DIP-Schalters. Anschließend werden zum Test der angeschlossenen Schrittmotoren einige Taktimpulse ausgegeben. Abgeschlossen wird die Testroutine durch einen permanent gesendeten ASCII-Zeichensatz an der seriellen Schnittstelle, den Sie durch Betätigen irgendeiner Taste der Rechnertastatur abbrechen. Jedes weiterhin von der IT 116 empfangene Zeichen wird als Echo zurückgesendet.

Den Selbsttestmodus können Sie nur durch einen µP-Reset der Prozessoreinheit beenden !

Zur Kontrolle der Testroutine aktivieren Sie die serielle Schnittstelle des Steuerrechners. Hierzu können Sie z. B. das folgende BASIC-Schnittstellen-Testprogramm verwenden.

Schnittstellen-Testprogramm (GW-BASIC):

```
REM Beispielprogramm für den Befehl Selbsttest
REM im DNC - Mode
REM ====================================
REM ------------------------------------------------------------------------------------------------------
REM serielle Schnittstelle COM 1 Initialisieren
REM 9600 Baud, keine Parität, 8 Datenbit, 1 Stopbit
OPEN „com1:9600,N,8,1,ASC“ FOR RANDOM AS #1
PRINT „com1:9600,N, 8,1,ASC“
REM für COM 2 OPEN „com2:9600,N,8,1,ASC“ FOR RANDOM #1 verwenden

REM ------------------------------------------------------------------------------------------------------

REM Selbsttest auslösen
PRINT #1, „@0?“: GOSUB meldung
PRINT „@0?“
ende:
REM und Ende
STOP
REM ------------------------------------------------------------------------------------------------------
REM Empfangsschleife
REM ------------------------------------------------------------------------------------------------------
meldung:
b$ = INKEY$: IF b$ <> „“ GOTO ende
REM auf ein Zeichen meldung
IF LOC(1) < 1 THEN GOTO meldung
REM Zeichen von der Schnittstelle lesen
a$ = INPUT$(1,1): c$ = c$ + a$: IF ASC(a$) = 13 THEN PRINT C$
IF ASC(a$) = 13 THEN c$ = „“
GOTO meldung
REM ------------------------------------------------------------------------------------------------------
REM Programmende
REM ------------------------------------------------------------------------------------------------------
```

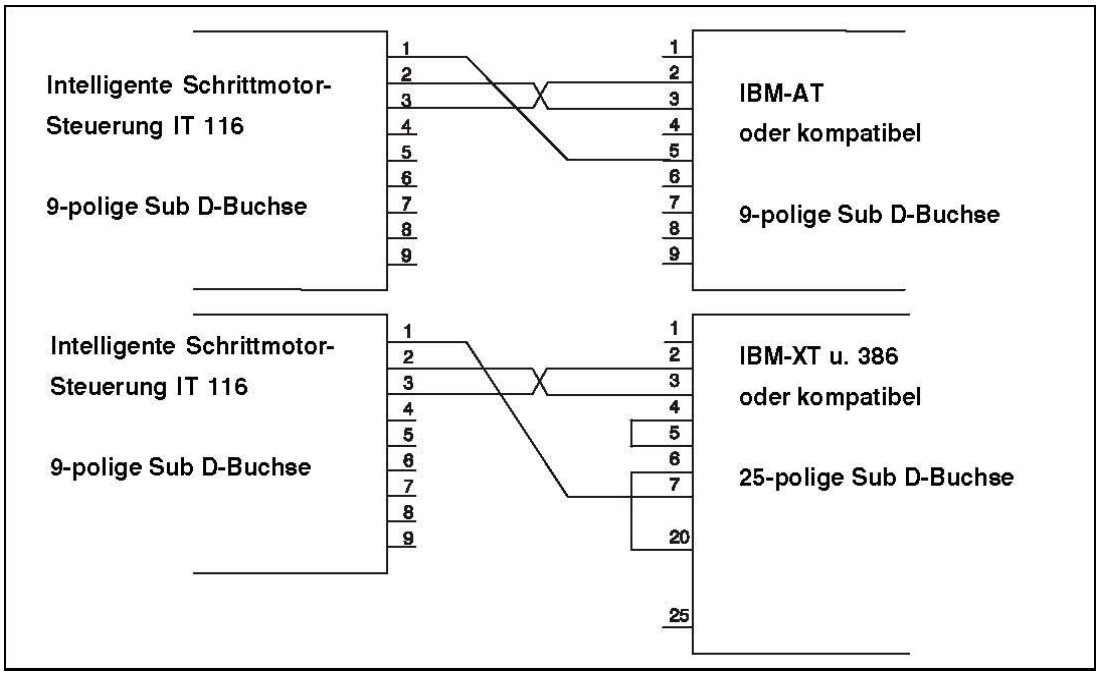


Bild 1: Anschluss serielle Schnittstelle

3.4 Funktionselemente der IT 116

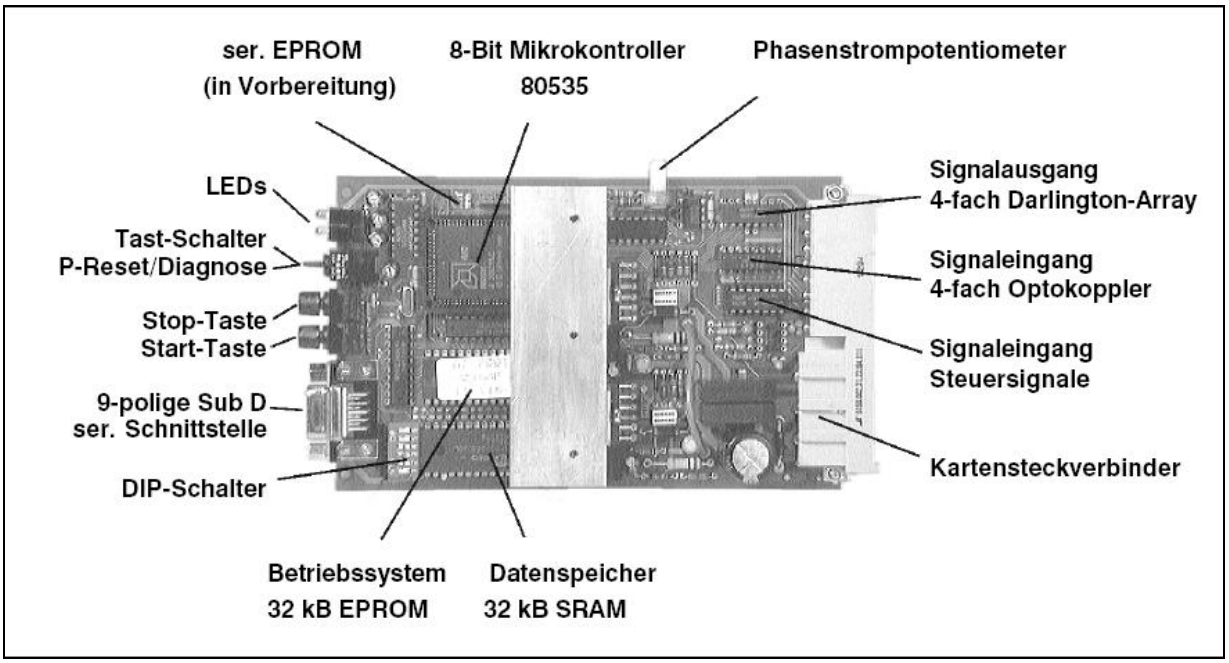


Bild 2: Die Funktions- und Bedienelemente

3.5 Einstellung DIP-Schalter S1

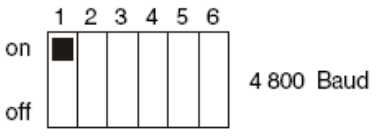
3.5.1 Baudrateneinstellung

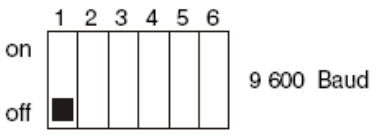
Zur Festlegung der Übertragungsrate der seriellen Schnittstelle wird nach jedem Mikroprozessor-Reset die Schalterstellung des 8-pol. DIP-Schalters S1 abgefragt.

Aus den Schalterstellungen der Schalter S1.1 und S1.2 ergeben sich folgende Baudraten:

S1.1	S1.2	Baudrate
ON	OFF	4 800 Baud
OFF	ON	9 600 Baud*

* Auslieferungszustand 9 600 Bd



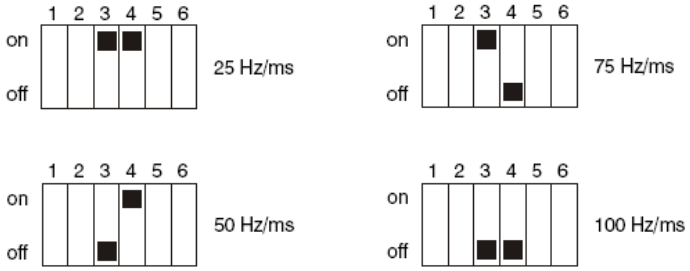


3.5.2 Einstellung der Beschleunigung

Bei Betrieb eines Schrittmotors außerhalb des Anlaufbereiches, ist eine Beschleunigungs-und Bremsrampe erforderlich.
Während bei der Beschleunigungsrampe die Schrittfolgefrequenz des Motors kontinuierlich von der Startfrequenz auf die Betriebsfrequenz gesteigert wird, erfordert die Verzögerungsrampe den umgekehrten Vorgang. Durch unterschiedliche Steigungen lassen sich die Kurven in Bezug auf Beschleunigungszeit und Last optimieren. So stehen Ihnen standardmäßig vier verschiedene Rampen zur Verfügung. Sie sind durch Schalter 1.3 und 1.4 des 8-pol. DIP-Schalters S1 entsprechend nachstehender Tabelle definierbar.

S1.3	S1.4	Rampe
ON	ON	25 Hz/ms*
OFF	ON	50 Hz/ms
ON	OFF	75 Hz/ms
OFF	OFF	100 Hz/ms

* Auslieferungszustand 25 Hz/ms



3.5.3 Einstellung Voll-/Halbschrittbetrieb

Schalter S1.5 legt die Betriebsart der IT 116 fest. In der Betriebsart Vollschritt werden bei einem Standard-Schrittmotor mit 1.8° Schrittwinkel 200 Schrittimpulse pro Umdrehung benötigt. In der Betriebsart Halbschritt erhöht sich die Anzahl der notwendigen Impulse auf 400, die Auflösung verbessert sich dadurch auf 0,9° pro Schritt.

S1.5	Betriebsart
ON	Vollschritt
OFF	Halbschritt

Die Betriebsart Halbschritt verfügt neben der höheren Schrittauflösung auch über ein besseres Laufverhalten im Hinblick auf Resonanz-Erscheinungen des Schrittmotorantriebes. Aus diesem Grunde sind die IT 116 ab Werk generell auf Halbschritt eingestellt.

3.5.4 Schalter S1.6, S1.7, S1.8

Diese Schalter werden zur Zeit nicht überwacht und stehen für Erweiterungen zur Verfügung.

3.6 Programmiermodus

Zum optimalen Einsatz der IT 116 ermöglicht das Betriebssystem sowohl eine Programmierung im DNC-Modus (direkte Ausführung der übergebenen Befehle) als auch im CNC-Modus, d. h. auszuführende Programme werden im internen Datenspeicher abgelegt und später durch ein Start-Signal gestartet (vgl. CNC-Betriebssystem 5.0).
Im **DNC-Modus** werden der Prozessoreinheit die Bearbeitungsparameter einzeln übergeben und von ihr direkt ausgeführt. Durch Auswertung der Quittierungssignale der IT 116 ist der übergeordnete Steuerrechner in der Lage, kontinuierlich und ohne Begrenzung Daten zu übergeben.
Im **CNC-Modus** (Speicherbetrieb) wird der Prozessoreinheit ein komplettes Datenfeld übergeben. Die Daten werden nach Erhalt vom Prozessor quittiert und in einem Daten-speicher abgelegt. Die Ausführung des Datenfeldes (ca. 1 200 Befehlssätze) erfolgt anschließend durch Betätigen der Start-Taste bzw. eines Startbefehles des Steuerrechners.

3.7 Spannungsversorgung

Zur Spannungsversorgung benötigt die Schrittmotorantriebseinheit IT 116 zwei getrennte Gleichspannungen.

- Als Versorgung der Logikelemente ist eine Gleichspannung von +5 V bei einem mittleren Stromverbrauch von ca. 300 mA notwendig.

- Als Motorspannung wird eine Gleichspannung von + 24 V ... + 44 V/2,0 A erwartet. Beide Spannungen werden intern durch spezielle Schutzschaltungen überwacht.

Hierbei führen Unterspannungen zur Reset- bzw. Störungsanzeige, Überspannungen werden mit Hilfe von Transil-Schutzdioden kurzgeschlossen.

Bereitstellung und Anschluss der Versorgungsspannung ist abhängig vom Gerätetyp:

IT 116 - Euro-Einschub
• Die Antriebseinheit verfügt über kein integriertes Netzteil. Somit sind entsprechend der Anschlussbelegung des Steckverbinders *Antriebsmodul* die Versorgungsspannungen zur Verfügung zu stellen.

IT 116 N - Euro-Kassetteneinschub
• Der Kassetteneinschub verfügt über ein integriertes Netzteilmodul mit einen 90 VA-Netztransformator. Die Spannungsversorgung von 230 V (50 Hz) wird über eine separate Hochstrom-Steckerleiste DIN 41612 H15 auf das Netzteilmodul geführt. Andere Versorgungsspannungen z. B. 125 V (60 Hz) sind auf Wunsch lieferbar.

IT 116 G - Euro-Profilgehäuse
• Das Schrittmotorantriebsmodul im geschlossenen Aluminium-Profilgehäuse wird über ein integriertes 120 VA Netzteil versorgt. Als Netzanschluss des 230 V (50 Hz) Ringkern-Transformators verfügt die Einheit über einen IEC-Stecker mit integriertem Filter. Sonderspannungen sind auf Wunsch lieferbar!

3.8 Anschluss und Inbetriebnahme

Zum Anschluss des Schrittmotors, der Signalein-/-ausgänge sowie von externen Bedienelementen verfügen die Einheiten über rückseitige Steckverbinder. Bei den Antriebseinheiten IT 116 N und IT 116 handelt es sich um Mischpolleisten F24/H7 nach DIN 41612, die in 19"-Systemgehäusen eine direkte Kontaktierung mit entsprechenden Gegensteckern erlauben. Bei der Antriebseinheit IT 116 im Euro-Profilgehäuse stehen zum Anschluss der externen Bedienelemente diverse Steckverbinder zur Verfügung (siehe Kapitel 4.10 Steckverbinder).
Die Signalein-/ausgänge im einzelnen:

3.8.1 Start

Der Signaleingang arbeitet parallel zur Start-Taste. Durch kurzzeitiges Verbinden mit dem GND-Potential wird ein in der Steuerung gespeichertes Programm gestartet.

3.8.2 Stop

Der Signaleingang arbeitet parallel zur Stop-Taste. Durch kurzzeitiges Betätigen eines Schalters (zwischen Eingang und GND-Potential) wird ein gezielter, geordneter Abbruch der aktuellen Befehlsausführung erreicht. Eventuelle momentane Positioniervorgänge werden durch Einleiten einer Stop-Rampe unterbrochen.

3.8.3 µP- Reset

Der Steuerungseingang µP-Reset liegt schaltungstechnisch parallel zum µP-Reset-Tast-Schalter. Das Verbinden des Einganges mit GND-Potential sperrt den Mikroprozessor und unterbricht somit alle Aktivitäten. Hierbei werden Positioniervorgänge der angeschlossenen Schrittmotoren abrupt beendet und die Schalt-Ausgänge zurückgesetzt.

3.8.4 Referenz-/Endlagenschalter

Zur Positionsbestimmung innerhalb eines Schrittmotorantriebssystems besteht die Notwendigkeit einen Maschinennullpunkt bzw. Referenzpunkt anzufahren. Zur Auswertung von entsprechenden Sensoren verfügt die Einheit über den Eingang Referenz-Schalter (Ref.Sw.).
Dieser aktiv-high-Eingang ist intern über einen Pull-up-Widerstand auf + 5 V gelegt. Die Auswertung des Signales erfolgt, wenn auf dem im Ruhezustand GND-Potential führenden Eingang ein + 5 V-Signal auftritt.
In isel-Lineareinheiten hat sich als Referenz-/Endlagenschalter ein Mikro-Schalter (Öffner-Schaltkontakt) durchgesetzt, der zw. GND und Signaleingang Ref.Sw. geschaltet ist. Wird während einer Verfahrbewegung der Schalter betätigt (d. h. der Signaleingang wird logisch 1) stoppt die Prozessoreinheit abrupt die Impulsausgabe an die Leistungsendstufe. Erfolgt eine Aktivierung des Schalters während einer Referenzfahrt, wird die Impulsausgabe ebenfalls unterbrochen, jedoch nach Ändern des Richtungs-Bit mit einer geringen Schrittfrequenz wieder gestartet.

Ein erneuter Interrupt (durch Verlassen des Schalterbereiches) stoppt den Schrittmotor exakt am Maschinen-Nullpunkt. Hierbei wird eine Wiederholgenauigkeit von ± 1 Schritt erreicht.

Bei Verwendung eines induktiven, kapazitiven oder optischen Näherungsschalters verbinden Sie den Minus-Pol des Sensors mit dem GND-Signal der Antriebseinheit sowie den Signalausgang des Sensors (open-collector) mit dem Steuerungseingang Ref.Sw.

- als Sensor muss ein NPN-Typ eingesetzt werden
- der Sensor muss als Öffner arbeiten (Ruhezustand: Ausgang leitend)

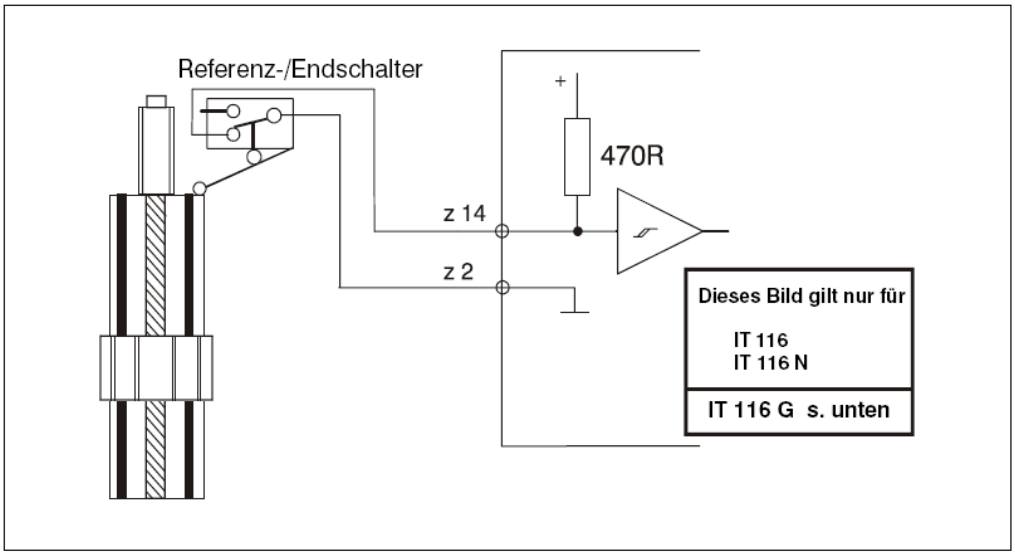


Bild 3: Anschluss Referenzschalter

Bei nicht oder falsch angeschlossenem Referenz-Schalter geht die Antriebseinheit in den Fehler-Zustand über (Diagnose-LED leuchtet). Wird kein Referenzschalter benötigt, ist der Signaleingang direkt auf GND-Potential zu legen.

info

IT 116 G	Bei dieser Antriebssteuerung befindet sich der Referenzschaltereingang an Pin 9 des 9pol. Sub D-Buchsenstecker *Steppermotor*. An Pin 5 liegt + 24 V an (siehe Kapitel 4.10.3).
IT 116 N	Bei dieser Antriebssteuerung ist der Referenzschaltereingang gegen Masse geschaltet (siehe Bild 3).
IT 116	Bei dieser Antriebssteuerung ist der Referenzschaltereingang gegen Masse geschaltet (siehe Bild 3).

3.8.5 Endstufenabschaltung (Disable)

Dieser Signaleingang dient zum Abschalten der Leistungsendstufe. Hierbei wird der angeschlossene Schrittmotor stromlos geschaltet.

Der Eingang ist über einen internen Pull-up-Widerstand (470R) auf + 5 V gelegt. Zur Aktivierung ist der Eingang mit GND-Potential zu verbinden.

Signaleingang “H” (offen) Endstufe aktiv (Schrittmotor bestromt)

Signaleingang “L” Endstufe inaktiv (Schrittmotor ist stromlos)

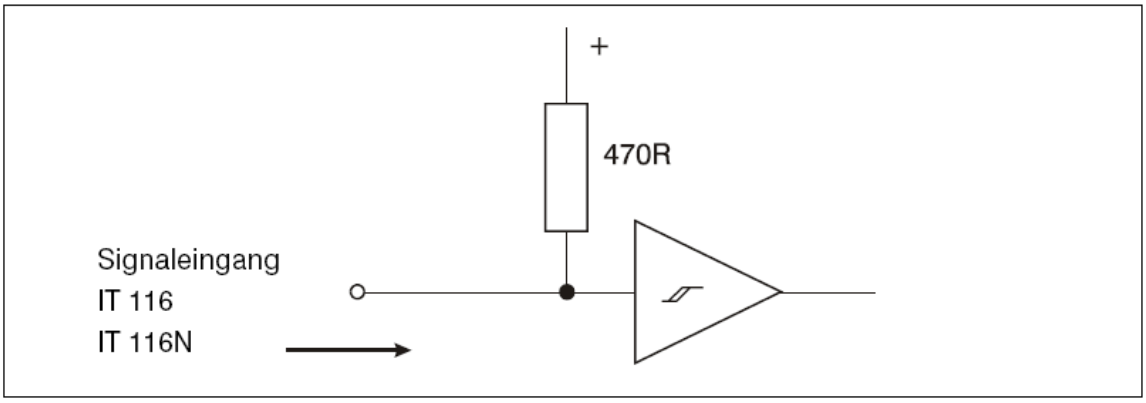


Bild 4: Signaleingang Disable

3.8.6 Signaleingänge In1-In4

Die Schrittmotorantriebseinheit IT 116 verfügt zur Bearbeitung von externen Ereignissen über vier optoisolierte Signaleingänge. Diese sind über ein gemeinsames Widerstandsnetzwerk (R = 2,2 k) auf + Vs-Potential gelegt. Somit bewirkt ein GND-Potential am Signaleingang das Setzen des Einganges.

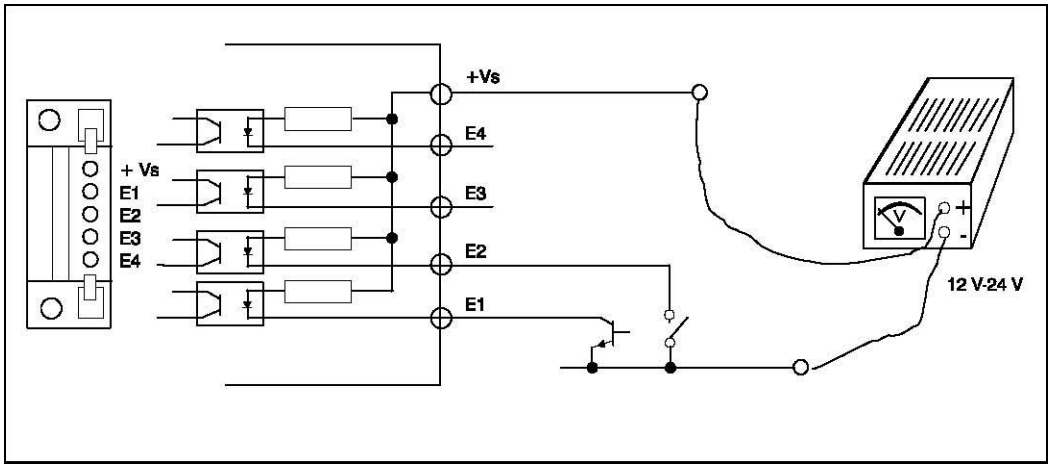


Bild 5: Aufbau der Signaleingänge In 1 - In 4

Die Bearbeitung des Signaleinganges innerhalb des zu verarbeitenden Datenfeldes geschieht durch Auslesen der Adresse 65535 Bit 0 - Bit 3 (siehe Handbuch CNC-Betriebssystem 5.0). Dabei ergibt sich folgende Zuordnung:

Signaleingang beschaltet	Wert der Speicheradresse (DNC-Mode)
kein	0
1	1
2	2
3	4
4	8
2 und 3	7
alle	15

Programmierbeispiel

@0b65535	Liest den Inhalt des kompletten Eingangsports (DNC-Betrieb)
.	
o65535,3,1,2	Bewirkt bei gesetztem Signaleingang 3 einen Vorwärtssprung um 2 im Programmablauf (CNC-Modus, bitweise Lesen).
o65535,128,8,-3	Bewirkt bei gesetztem Signaleingang Bit 4 einen Rückwärtssprung um 3 im Programmablauf (CNC-Modus, byteweise Lesen).

Der Zustand des Eingangsports wird durch eine LED angezeigt.

3.8.7 Signalausgang Out1- Out

Als Schaltausgänge verfügen die Antriebseinheiten IT 116 und IT 116 N über vier Open-Collector-Transistorausgänge. Die maximale Belastbarkeit der mit einem gemeinsamen GND-Potential ausgestatteten Signalausgänge beträgt in der Summe 500 mA bei einer maximalen Betriebsspannung von 50 V DC.

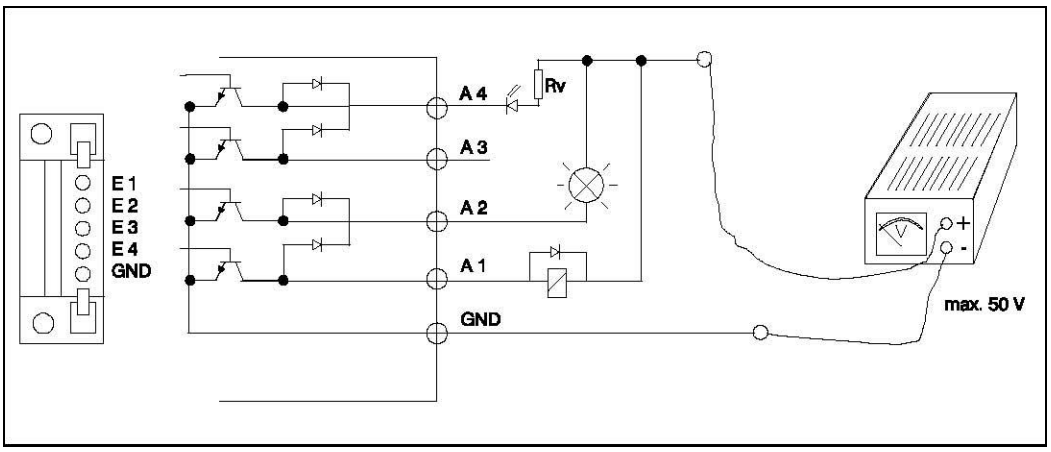


Bild 6: Aufbau der Signaleingänge Out 1 - Out 4

Die Ausgänge können innerhalb der Programmierung durch Beschreiben der Adresse 65534 sowohl bit- als auch byteweise verändert werden (siehe Handbuch CNC-Betriebssystem 5.0).
Beachten Sie, dass lediglich die Bits 0 - 3 programmierbar sind.

Wert der Speicheradresse	**Ausgang gesetzt (DNC-Mode)**
0	kein
1	1
2	2
4	3
8	4
5	1 und 3
15	alle

Der IT 116 G besitzt eine andere Signalausgangsbeschaltung! (s. Kapitel 4.10.3)

Programmierbeispiel

@0B65534,4	Beschreibt den Ausgangsport mit dem digitalen Wert von 4, d. h. Ausgang 3 wird gesetzt (DNC-Mode).
p65534,3,1	Setzt den Ausgang 3, d. h. der Transistor ist durchgeschaltet (CNC-Mode, bitweise Setzen).
p65534,2,0	Setzt den Ausgang 2 zurück, d. h. der Transistor 2 ist gesperrt (CNC-Mode, bitweise Löschen)

Als Status-Anzeige des Signalausgangsports befinden sich in der Frontplatte aller IT 116 entsprechende Kontroll-LED.

info

Die Signaleingänge Reset, Ref.Sw., Enable, Start und Stop der Antriebseinheiten IT 116 und IT 116 N sind intern über ein Widerstandsnetzwerk (470 Ohm) an + 5 V-Versorgungs-spannung gelegt und anschließend auf eine Schmitt-Trigger-Eingangsstufe geführt. Dies ermöglicht eine relativ störungsfreie Verarbeitung von externen Steuersignalen. Längere Verbindungen von Bedieneinheiten und Steuerung sollten jedoch nur mit geschirmten Leitungen ausgeführt werden! Bei der Steuerung IT 116 G im Euro-Profilgehäuse werden die Eingänge zusätzlich über Opto-Koppler entkoppelt.

3.8.8 Betriebsstromeinstellung (I_{Phase})

Die Schrittmotor-Antriebssteuerungen der IT 116-Serie stellen einen gechoppten Betriebsstrom von max. 3,5 A zur Verfügung. Zur Einstellung des betriebsbedingten Betriebsstromes verfügen die Steuerungen über ein seitlich angeordnetes Potentiometer I_{Phase} (siehe Kapitel 1.2 Bedien-/Anzeigenelemente).

Die Ermittlung des Schrittmotor-Betriebsstromes (Phasenstrom) geschieht normalerweise mit einer Phasenstromzange. Näherungsweise kann jedoch ein Dreheisen-Messinstrument in eine Motorleitung eingeschleift werden. Bei einer programmierten Schrittfrequenz von ca. 400 Hz im Halbschrittbetrieb zeigt das Messinstrument:

$$I_{Mess} = I_{Phase} \times 0{,}7 \quad \rightarrow \quad I_{Phase} = I_{Mess} / 0{,}7$$

Eine weitere Möglichkeit den Phasenstrom zu bestimmen, besteht darin, einen Shunt-Widerstand von 0,1 Ohm/2 W in die Motorphase 1A einzuschleifen und den Spannungsabfall während des Stillstandes des Motors zu messen. Hierbei ist die Steuerung zuvor durch einen Reset in eine definierte “Startposition” zu bringen.

$$I_{Phase} = U_{Phase} / R_{Shunt}$$

Im Lieferzustand ist der Phasenstrom der Schrittmotoren auf ca. 2 A eingestellt.

Zur Minimierung von Verlustleistung und damit Erwärmung der Schrittmotorendstufe reduziert die Einheit bei Motorstillstand automatisch den Betriebsstrom auf 60 % des eingestellten Phasenstromes.

3.8.9 Schrittmotorausgang

Der Ausgang der Schrittmotorsignale liegt - bedingt durch die unterschiedlichen Gehäuseausführungen - bei den Antriebssteuerungen IT 116 und IT 116 N am rückwärtigen Steckverbinder des Antriebsmodules; bei der IT 116 G im Euro-Profilgehäuse am 9-poligen Sub D-Buchsenstecker *Steppermotor.*

Die Schrittmotorsignale sind vier Impulsspannungen von max. 44 V. Dabei bilden jeweils zwei Ausgänge eine Schrittmotorphase (1A - 1B = Phase 1, 2A - 2B = Phase 2).

Zum Schutz vor Kurzschlüssen werden die Ausgänge elektronisch überwacht.

Bei Ankopplung der Motoren achten Sie jedoch unbedingt darauf, dass sich die IT 116 in einem spannungslosen Zustand befindet.

3.8.10 Datenspeicher

Zur Speicherung von systembedingten Variablen und programmierten Funktionsabläufen im CNC-Betrieb, verfügen die Schrittmotorantriebseinheiten über ein 32 kB statisches RAM. Da dieser Speicher nach Wegfall der Versorgungsspannung die gespeicherten Informationen verliert, ist ggf. in Stand-Alone-Applikationen eine Pufferung der Versorgungsspannung des RAM notwendig. Dies geschieht bei der IT 116 durch Einsatz eines speziellen Batteriesockels, der bei Unterschreiten der Versorgungsspannung 4,75 V auf die interne Batterie (2 x 35 mAh) umschaltet und gleichzeitig einen Schreibzugriff auf das RAM durch Anheben des CE-Signal auf “high”-Potential verhindert. Bei Verwendung eines CMOS-RAMs ist ein theoretischer Datenerhalt von bis zu 5 Jahren möglich (Art.-Nr. 328122).

Einbauhinweis

1. Obere Kühl- bzw. Gehäuseplatte abschrauben.

2. RAM-Speicher entfernen.

3. Batterie-Sockel in vorhandenen IC-Sockel einsetzen und sichern.

4. RAM-Speicher in Batterie-Sockel einsetzen. Achten Sie auf die Markierungen (Pfeil), sie geben die Einbaurichtung vor. Setzen Sie nur CMOS -Speicher mit geringem Stand-by-Strom ein, da der Stromverbrauch maßgeblich die Datenerhaltezeit beeinflusst.

5. Gehäuse schließen

6. Selbsttest durchführen

3.9 Betriebsstörungen

Baugruppen der IT 116-Serie sind komplette Schrittmotor-Antriebssysteme. Sie umfassen alle notwendigen Funktionsgruppen zum direkten Betrieb eines 2(4)-Phasen-Schrittmotors von einem übergeordneten Steuerrechner.

Zur Überwachung des Systems befinden sich auf der Einheit verschiedene Schutzfunktionen. Sie führen bei Aktivierung zum Abbruch der Programmbearbeitung des CNC-Datenfeldes bzw. zum Abschalten des angeschlossenen Schrittmotors. Als Meldesignal leuchtet im Fehlerfall die LED *Diagnose* auf.

Zur Differenzierung der vorhandenen Fehlerursache verfügt die Antriebseinheit über ein Selbsttestmodus, den Sie mit Hilfe des Diagnose-Tast-Schalters aktivieren können. Daraufhin wird die Fehlerquelle von der Diagnose-LED durch Blink-Impulse ausgegeben. Die Anzahl der Blink-Impulse gibt den entsprechenden Fehler an.

Hierbei gilt folgende Zuordnung:

LED blinkt	nicht	kein Fehler
	1 x	Referenzschalterauswertung
	2 x	Enable-Eingang aktiv
	3 x	Unterspannung der Motorspannung
	4 x	Übertemperatur der Endstufe
	5 x	Überstromabschaltung der Endstufe

Berücksichtigen Sie, dass zu Beginn der Anzeige die LED aufleuchtet. Erst nach der folgenden Dunkelphase der LED wird das erneute kurzzeitige Aufleuchten als ein Impuls gewertet. Nach dem letzten Blinkimpuls mit anschließender Dunkelphase leuchtet die LED wiederum permanent auf.

Fehlercode	**Fehlerart**	**Fehlerbeseitigung**
Referenzschalterauswertung	-Signaleingang Ref.Sw. ist unbeschaltet (offen) -mechanische Antriebs-achse befindet sich im Referenzschalterbereich -inkorrekte Verbindung von Steuerung und Motor	-Signaleingang auf GND-Potential legen. -Endstufe abschalten und Linearvorschub von Hand aus dem Schalterbereich bewegen. -Verbindungsleitung testen
Enable-Eingang	-Signaleingang ist aktiv (Brücke zwischen Eingang und GND)	
Unterspannung der Motorspannung	-fehlende Versorgungsspannung + 24 V bis + 44 V -Eingangsspannung ist kleiner als + 24 V	-Spannung kontrollieren -Sicherung kontrollieren (evtl. durch Überspannung ausgelöst) -höhere Eingangsspannung wählen
Übertemperatur der Endstufe	-integrierter Temperaturschutzschalter der Endstufen -Kühlfläche ist aktiv	-Antriebseinheit fremdbelüften -ggf. Phasenstrom reduzieren
Überstromabschaltung der Endstufe	-Kurzschluss Phase - Phase Phase - Masse Phase - Motorspannung - Überstrom infolge falsch (zu hoch) eingestellten Phasenstromes	- Kurzschluss beheben und Steuerung erneut einschalten -Phasenstrom reduzieren •auf den im Motordatenblatt vorgeschrieben Strom • auf den in der Applikation notwendigen Betriebsstrom

Nach Beheben des Fehlers und nochmaliger Betätigung des Diagnose-Tast-Schalters erlischt die Diagnose-LED. Die Antriebseinheit ist wieder betriebsbereit.

In Verbindung mit einem an der seriellen Schnittstelle angeschlossenem PC oder Datenterminal ist die Diagnose der Antriebseinheit durch den DNC-Befehl @0D möglich. Hierbei antwortet die Antriebssteuerung direkt mit der aktuellen Fehlernummer.

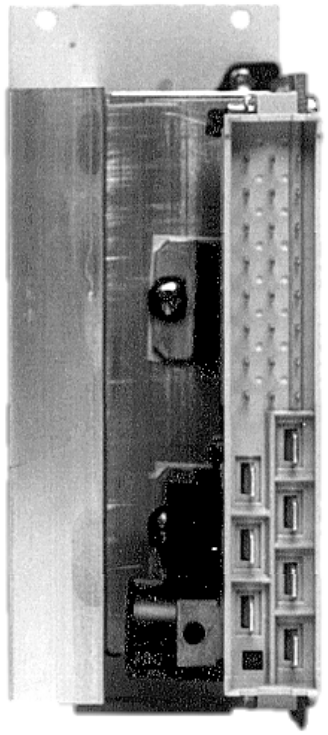
Bei Fehler 1 (Referenzschalterauswertung) wirkt der Befehl ähnlich dem Initialisierungscode @01. Somit können Sie ggf. mit einer positiven Positionierbewegung ein im Endschalter-bereich liegenden Linearantrieb freifahren.

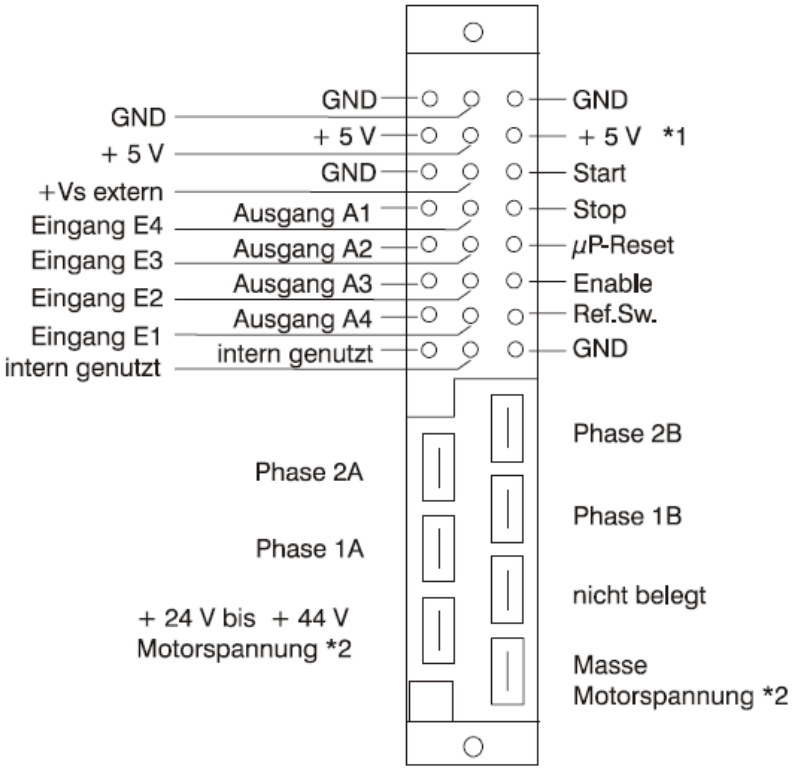
3.10 Steckverbinder

Die drei Ausführungen der IT 116-Antriebseinheiten besitzen verschiedene Anschluss-und Kontaktbelegungen.

3.10.1 IT 116 (Euro-Einschub)

- Mischpolleiste DIN 41612 H7/F24 -



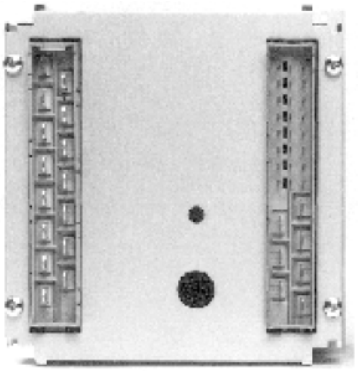


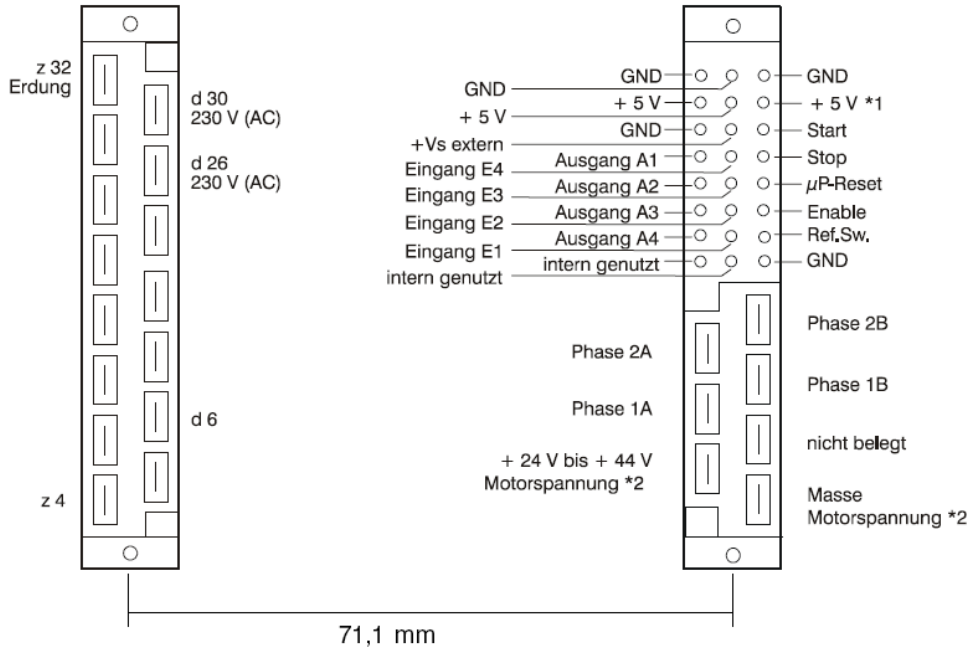
* 1 Spannungsversorgung + 5 V extern
(nur bei Art.-Nr. 381 011)

*2 nur bei Art.-Nr. 381 011

3.10.2 IT 116 N (Euro-Kassetteneinschub)

Antriebsmodul: - Mischpolleiste DIN 41612 H7/F24 -
Netzteilmodul: - Hochstromleiste DIN 41612 H15 -





* 1 Spannungsversorgung + 5 V extern
(nur bei Art.-Nr. 381 011)

*2 nur bei Art.-Nr. 381 011

3.10.3 IT 116 G (Euro-Profilgehäuse)

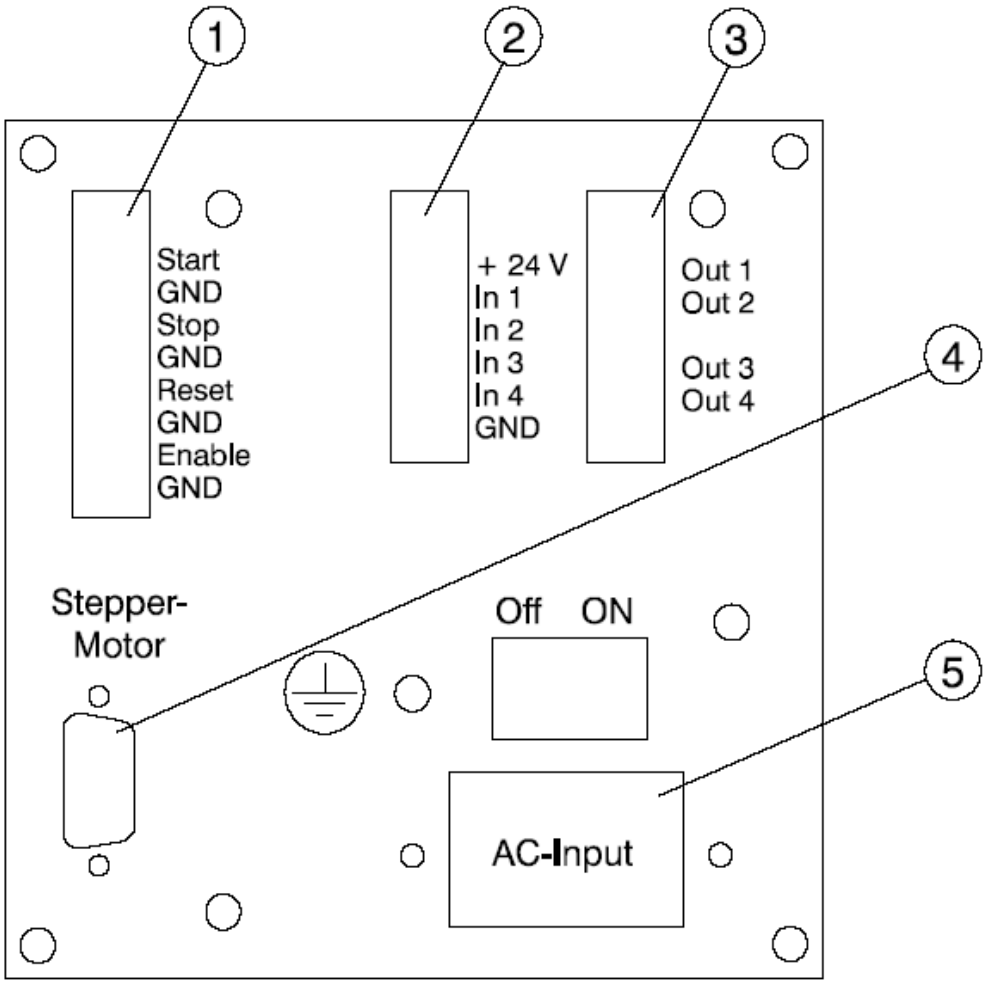


Bild 7: Anschlüsse der Rückseite

1. **Signaleingang *Steuersignale***

   8pol. Schraubklemm-Steckverbinder (Gegenstecker im Lieferumfang enthalten)

   Der Signaleingang ermöglicht eine mittels Opto-Koppler getrennte Fernbedienung der IT 116 G. Durch interne Beschaltung auf + 24 V-Potential (2,2 k Vorwiderstand) genügt hier die Verbindung von *Signaleingang* und *GND*-Signal um die entsprechende Funktion auszulösen.

2. **Signaleingang *In 1 - In 4***

   6pol. Schraubklemm-Steckverbinder (Gegenstecker im Lieferumfang enthalten)

   Der Signaleingang In 1 - In 4 stellt Ihnen vier frei belegbare Signaleingänge zur Verfügung. Durch Auswerten der Speicheradresse 65534 (Bit 0 - Bit 3) ist eine Einbindung der Signaleingänge innerhalb der CNC-Programme möglich (siehe hierzu Kapitel 4.8.6). Durch interne Beschaltung auf das + 24 V-Potential (2,2 k Vorwiderstand) führt ein GND-Signal am Eingang zum Setzen des entsprechenden Bits.

3. Signalausgang *Out 1 - Out 4*

6pol. Schraubklemm-Steckverbinder (Gegenstecker im Lieferumfang enthalten)

Bei dem Signalausgang handelt es sich um vier programmierbare Relaiskontakte, die zu zwei Gruppen mit jeweils zwei Schaltkontakten zusammengefasst sind (siehe Bild 7).

Das Setzen der Ausgänge ist durch Beschreiben der Adresse 65535 Bit 0 - Bit 3 möglich (siehe hierzu Kapitel 4.8.7).

4. Schrittmotorausgang

9pol. Sub D-Buchsenstecker (Motoranschlussleitung im Lieferumfang enthalten, l = 3 m)

Kontaktbelegung:

Pin	**Signal**
1	Schrittmotorphase 2B
2	Schrittmotorphase 2A
3	Schrittmotorphase 1B
4	Schrittmotorphase 1A
5	+ 24 V
6	nicht belegt
7	nicht belegt
8	nicht belegt
9	Referenzschalter (Öffner-Kontakt, N. C.)

Die Eingänge 5 und 9 des Steckverbinders bilden einen optoisolierten Signaleingang, der als Referenz- bzw. Nullpunktschalter ausgewertet wird. Bei Montage des entsprechenden Schalters einer NC-Achse achten Sie darauf, dass der Schaltkontakt in der Ruhestellung geschlossen ist und erst bei Betätigung öffnet.

5. Netzeingang

Als Netzeingang verfügt die IT 116 G über ein Netzeingangsmodul mit integriertem Filter nach EN 50082-1. Zur Netzabsicherung verfügt der Netztransformator über eine integrierte Primärsicherung.

4 Inkompatibilität

4.1 Positionierung

Die Schrittmotor-Antriebseinheit IT 116 ist ein kompletter Ein-Achs-Controller. Als Betriebssystem der integrierten Mikroprozessoreinheit wurde das CNC-Betriebssystem 5.0 eingesetzt, das für max. drei Antriebsachsen ausgelegt ist. Somit ergeben sich bei der Programmierung der IT 116 folgende Einschränkungen:

- Bei der Initialisierung der Einheit ist generell nur die X-Achse zu definieren:
  @01 - Direktmodus
  #axis x; - PAL-PC

- Zur Programmierung von Positioniervorgängen ist lediglich ein Zahlenpaar bestehend aus Weg- und Geschwindigkeitsinformation zu übergeben.

DNC-Betriebsmodus

@0A 2000,5000	Bewegung relativ um 2 000 Schritte mit einer Geschwindigkeit von 5 000 Schritten/s
@0M 1000,3000	Positionierung absolut auf den Punkt 1 000 (Bezug: Referenzpunkt) mit einer Geschwindigkeit von 3 000 Schritten/s

CNC-Betriebsmodus

0 2000,5000	Bewegung relativ
m 1000,3000	Bewegung absolut

PAL-PC

move 50(2000)	Bewegung relativ um 50 mm
moveto 50(1000)	Bewegung absolut zu Position 50 mm (Bezug: Referenzpunkt)

Eingabefehler führen zu der Fehlermeldung 3 des CNC-Betriebssystems 5.0

4.2 Signalein-/ausgänge

Die Antriebssteuerung IT 116 verfügt über je vier Signalein- und -ausgänge. Da das CNC-Betriebssystem 5.0 jeweils das komplette 8-Bit-Datenwort liest bzw. beschreibt (siehe Peek- und Poke-Befehle im Handbuch CNC-Betriebssystem 5.0), sind folgende Einschränkungen zu beachten:

- Zur Auswertung der Eingänge werden die niederwertigen Bits 0 - 3 der Adresse #65535 abgefragt.
  Eingang 1 - Bit 0
  Eingang 2 - Bit 1
  Eingang 3 - Bit 2
  Eingang 4 - Bit 3

- Die oberen vier Bit (4 - 7) liegen auf logisch 1

- Die Programmierung der Ausgänge geschieht durch Beschreiben der unteren 4 Bit der Ausgangsadresse #65534. Bei bitweisem Beschreiben stehen somit die Bits 0 - 3 zur Verfügung (Bit 0 = Ausgang 1, Bit 1 = Ausgang 2 usw.). Bei byteweiser Verarbeitung des Ports beinhalten - entsprechend dem BCD-Code die Ausgangswerte zwischen 0 und 15 alle möglichen Signalzustände.

4.3 Diagnose

In Verbindung mit einem an der seriellen Schnittstelle der Steuerung angeschlossenen PC- oder Datenterminal stellt der zusätzliche DNC-Befehl @0D eine direkte Diagnose-Auswertung zur Verfügung. Dabei antwortet die Antriebssteuerung auf den übergebenen Befehl mit der aktuellen Fehlernummer. Bei anliegendem Fehler 1 (Referenzschalterauswertung) ermöglicht der Befehl außerdem eine nachfolgende Programmierung einer Verfahrbewegung. Hierdurch kann ggf. ein im Endschalter-Bereich liegender Linearantrieb freigefahren werden.

isel®
From Components to Systems

isel-CNC-Betriebssystem 5.x

Programmieranleitung

www.isel.com

isel Germany AG, D-36124 Eichenzell, Bürgermeister-Ebert-Str. 40 ☎ (06659)981-0 🖷 (06659)981-776

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 *isel*-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 Betriebssytemes 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

#axis xyz;

GW-BASIC

```
100 open“com1:9600,N,8,1,DS,CD“as #1
110print#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 ***isel-*** 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

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:

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

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 wurde. Hier kann ggf. die getrennte Referenzfahrt der Achsen Kollisionen mit dem Werkstück verhindern.

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.

Beschränkung

Den Befehl können Sie verwenden, nachdem eine Initialisierung der Prozessorkarte durch den Befehl *Achsenanzahl setzen* erfolgt ist und beschränkt sich auf die dort vorgegebene Achsenkonfiguration.
Bei falscher Achsangabe erfolgt die Fehlerrückmeldung 3.
Befindet sich die Karte im 3D-Modus, schaltet der Befehl wieder in die 2,5D-Ausführung zurück.

Programmierbeispiel

PAL-PC	GW-BASIC
#axis xyz;	100 open“com1:9600,N,8,1,DS,CD“ast#1
reference xyz;	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

info

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 abzubrechen.

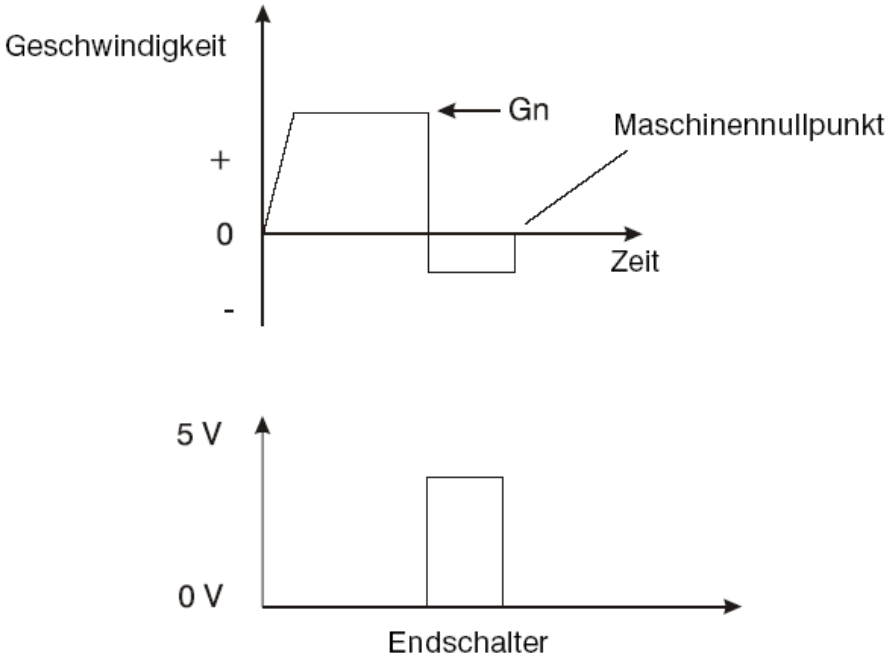


Bild 1: Verlauf einer Referenzfahrt (zum Maschinennullpunkt)

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>	(x)
@<GN>d<Gx>, <Gy>	(X-y)
@<GN>d<Gx>, <Gz>	(X-z)
@<GN>d<Gx>, <Gy>, <Gz>	(X-Y-z)
<GN>	= Gerätenummer, Standard = 0
<Gx>	= Referenzgeschwindigkeit x
<Gy>	= Referenzgeschwindigkeit y
<Gz>	= Referenzgeschwindigkeit z
<Gx>, <Gy>, <Gz>	= ganze Zahl zwischen 30 und 10 000 Hz

Anwendung

@0d2500	(1 Achse)
@0d2400,3000	(2 Achsen)
@0d1000,3000,9000	(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 *Memory-Back-up* 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 Schrittanzahl und Schrittgeschwindigkeit für jede Leistungsendstufe eine Impulskette aus. Die Verfahrbewegung wird sofort ausgeführt oder gespeichert.

Aufbau

@<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2>

<GN> = Gerätenummer, Standard = 0
<Sx> = Schrittanzahl x, Wert zwischen 0 und +/- 8 388 607
<Gx> = Geschwindigkeit x, Wert zwischen 30 und 10 000

-
-

<Gz2> = Geschwindigkeit Z-Achse (2. Bewegung)

Anwendung

@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)

Erläuterung

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 Schrittanzahl und Geschwindigkeit.
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.
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.

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	GW-BASIC
#axis xy; **move50(500),300(900);**	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	000010	002000	FFFFFE
	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 Postion der Mechanik entspricht, da kein Regelkreis vorhanden ist.

Programmierbeispiel

PAL-PC

(Terminalmode)

@0P

GW-BASIC

-

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 *Verfahre absolut*-Anweisungen berücksichtigen diesen virtuellen Nullpunkt als neuen Bezugspunkt

Aufbau

@<GN>n<Achsen>

<GN>	= Gerätenummer, Standard = 0
<Achsen>	= ganzzahliger Wert zwischen 1 und 7

Anwendung

@0n7	@0n1
•	•
•	•

Erläuterung

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.
Soll für mehrere Achsen eine Nullpunktverschiebung durchgeführt werden, müssen obige Werte addiert werden:

Achsen	Wert	Achsen	Wert
x	1	xy	5
y	2	yz	6
z	4	xyz	7
xy		3	

Der Rechner meldet sich nach erfolgter Ausführung mit der Rückmeldung (vgl. Software-Handshake).

Beschränkung

Der virtuelle Nullpunkt hat nur für den Befehl *Verfahre absolut* eine Bedeutung. Relativpositionierung wird vom virtuellen Nullpunkt nicht beeinflusst, da hier ein Verfahrvektor 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
```

info

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

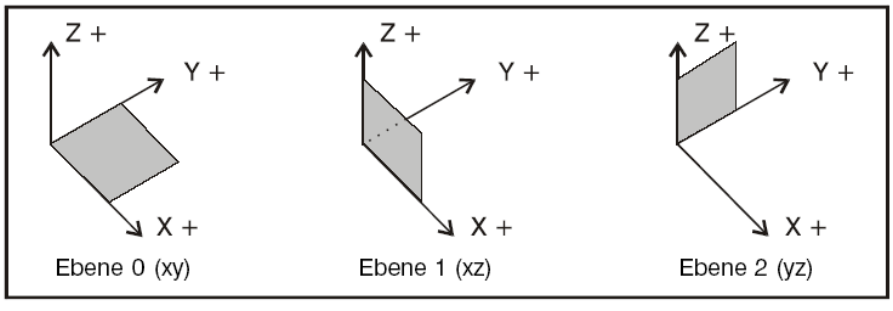


Bild 2: Umschaltbare Interpolationsebenen

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

(Terminalmode)

GW-BASIC

@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

(Terminalmode)

@0B33211,128

GW-BASIC

-

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

(Terminalmode)

GW-BASIC

@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

(Terminalmode)

@0G1

GW-BASIC

```
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

01234	00001	30	000001	FE87...	FFFF01	FE01
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 Verfahrweges in der 2er-Komplement-Darstellung.
E Geschwindigkeitswert der X-Achse - Um die Geschwindigkeit rückzuwandeln, kann der Bruch 921600/(HI-Byte*(256-LO-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 Verfahr- 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 @<GN>z<STATUS>

<GN> = Gerätenummer, Standard = 0
<STATUS> = 0 - 3D-Interpolation aus
= 1 - 3D-Interpolation ein

Anwendung @0z1

Erläuterung Die Prozessorkarte wird mit dem Dateneröffnungsteil @0 auf einen neuen Befehl vorbereitet. z1 ändert die Interpolation von 2-Achs- auf 3-Achs-Betrieb.

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.

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
```

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 3DInterpolation 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*Anfangswinkel/180

E - Endpunkt der Bewegung
E = pi*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*Radius*(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+Raduis*(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;
```

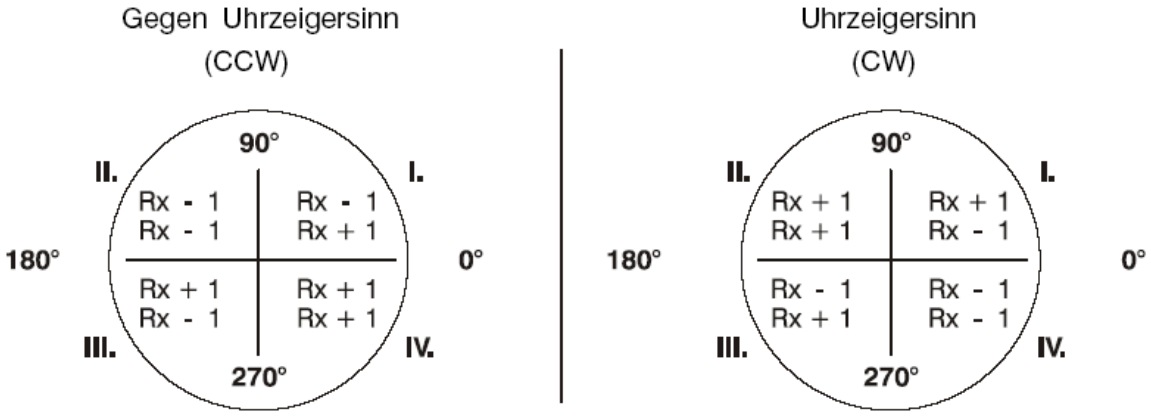
Die berechnete Bogenlänge muss auf den nächsten, ganzzahligen Wert gerundet übergeben werden. Hierbei sind Werte im Bereich von 3 bis 8 000 000 Schritten zulässig.

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 = Radius * cos(A)
Ys = 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 Verfahrgeschwindigkeit 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*Radius*(E-A) /180
	= 2*200*(5*pi /4-4*pi) /pi
	= 400

3. Richtung Rx, Ry:

Anfangswinkel	= 135°, Gegenuhrzeigersinn
Rx	= - 1
Ry	= - 1

4. Startpunkt Xs, Ys:

Xs	= Radius*cos(Anfangswinkel)
	= 200*cos(135)
	= -141
Ys	= Radius*sin(Anfangswinkel)
	= 200*sin(135)
	= 141

5. Interpolationsparameter D: (Gegenuhrzeigersinn)

D = (Rx*Ry*Radius + Rx*Ry*Summe(Radius -1.0)
- Rx*Summe(Xs+(Rx-Ry) /2.0) + Ry*Summe(Ys+(Rx+Ry) /2.0)) /2

Summe(Radius-1) = Summe (200-1)
= Summe (199)
= 199*(199+1)
= 39 800

Summe(Xs+(Rx-Ry) /2.0) = Summe (-141 + (-1 -(-1)) /2.0)
= Summe (-141)
= -(-141)*((-141) -1)
= - 20 022

Summe (Ys+(Rx+Ry) /2.0) = Summe (141 +(-1 +(-1)) /2.0)
= Summe (141-1)
= 140*(140+1)
= 19 740

D = (Rx * Ry*Radius + Rx* Ry*39800 - Rx*(-20022) + Ry*19740) /2
= ((-1)*(-1)*200 + (-1)*(-1)*39800 - (-1)*(-20022) + (-1)*19740) /2
= (200 + 39800 - 20022 - 19740) /2
= 119

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 *Break*-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> <Achsanzahl> 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 Schrittanzahl und Schrittgeschwindigkeit für jede Leistungsendstufe eine Impulskette aus.

Die Verfahrbewegung wird sofort ausgeführt oder gespeichert.

Aufbau

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

0	= Bewegung relativ
<Sx>	= Schrittanzahl x, Wert zwischen 0 und ± 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 Schrittanzahl 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 *Verfahre absolut*-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

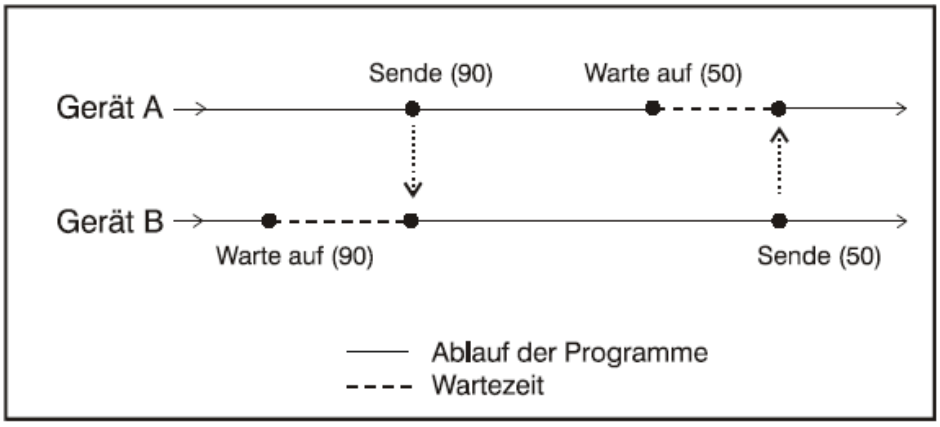
Die Prozessorkarte teilt einer zweiten Prozessorkarte oder einem übergeordneten Rechner mit, dass ein bestimmter Punkt im Ablaufschema (NC-Programm) erreicht ist. Der Befehl dient dazu, die Prozessorkarte mit einer externen Einheit zu synchronisieren bzw. eine externe Einheit aufzufordern, eine Tätigkeit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



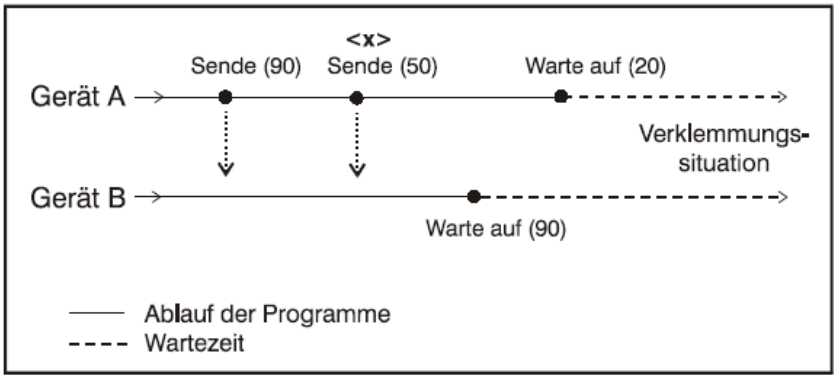
Beschränkung

Durch die Befehle *Synchronisationszeichen senden* und *Warte auf Synchronisationszeichen* können ohne übergeordneten Rechner nur zwei Prozessorkarten synchronisiert werden.
Die übergebene 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 eröffnet. Die seriellen Schnittstellen der Geräte sind über das Verbindungskabel Interface - Interface zu verbinden.

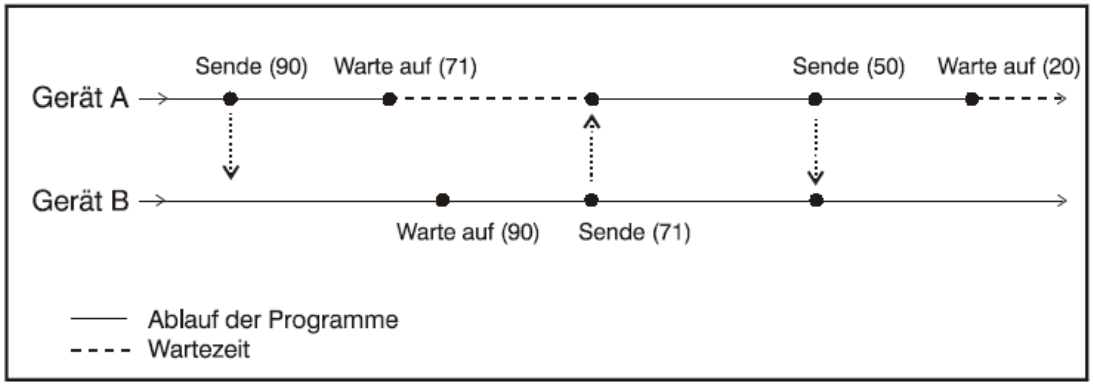
Programmierbeispiel

PAL-PC	GW-BASIC
#axis xyz;	100 open“com1:9600,N,8,1,DS,CD“as #1
#input	110 print#1,“@07":gosub 1000
send 90;	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>,	<Offset>
<SyncChar>	= Synchronisationszeichen von 33 bis 125
<Offset>	= Verzweigung bei Empfang <SyncChar>+1 Zahl zwischen - 32 767 und + 32 767

Anwendung

2 50,0	Warte auf Synchronisationszeichen
502 55,7	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.

Programmierbeispiel

PAL-PC

```
#axis x;
input
label: move 3(1000);
wait 50,label;
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,“0 500,5000":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>	= Schleifen:	0 < Zahl < 32 767
	Verzweigung:	0
<Offset>	=	Anzahl der zu wiederholenden Befehle bzw. Verzweigungsziel relativ Schleifen: - 1 > Zahl > - 3 000 Verzweigung: - 3 000 < Zahl < 3 000

Anwendung

3 25,-1	Wiederhole letzten Befehl 25 mal
3 0,-5	Verzweige immer 5 Schritte zurück
3 0,5	Überspringe die nächsten 4 Befehle
3 6,-5	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	GW-BASIC
#axis x;	100 open“com1:9600,N,8,1,DS,CD“as #1
#input	110 print#1,“@01":gosub 1000
repeat	120print#1,“@0i“:gosub1000
repeat	130 print#1,“0 200,2000":gosub 1000
move 2(1000);	**140 print#1,“3 5,-1":gosub 1000**
until 5;	150 print#1,“0 -1000,1000":gosub 1000
move -10(2000);	**160 print#1,“3 10,-3":gosub 1000**
until 10;	170 print#1,“9":gosub 1000
stop.	180 print#1,“@0S“:gosub 1000
#start	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

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.

Programmierbeispiel

PAL-PC

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

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 Starttastereingang
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 Verfahrweg 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> = Schrittanzahl 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
```

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

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 tangentiale Richtung des zu beschreibenden Kreises.

Erläuterung

(siehe Befehl *Zirkularinterpolation* Kapitel 2.2.2)

Programmierbeispiel

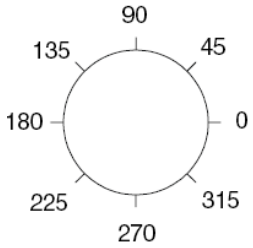
Nach einer relativen Positionierung von 150 mm (600 Schritte) soll ein Viertelkreis entgegen dem Uhrzeigersinn ausgeführt werden. Der Radius des Kreises ist mit 50 mm (200 Schritte) vorgegeben, der Startwinkel beträgt 0° der Endwinkel 90°. Die Geschwindigkeit über den gesamten Verlauf soll 200 Schritte/s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

PAL-PC	GW-BASIC
#axis xyz; reference xyz; move 150(200),150(200), 15(800),0(21); circle_ccw50(300),0,90; stop.	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, Verfahrgeschwindigkeit 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	GW-BASIC
#axis x;	100 open“com1:9600,N,8,1,DS,CD“as #1
reference x;	110 print#1,“@01":gosub 1000
set_port 65529,5=0;	120 print#1,“@0i“:gosub 1000
set_port 65530,128=27;	**130 print#1,“p 65529,5,0":gosub 1000**
stop.	**140 print#1,“p 65530,128,27":gosub 1000**
#start	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

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
```

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 <Tastennummer>,<Offset>

<Tastennummer>	= Zahl zwischen 1 und 20 (255)
<Offset>	= Anzahl der Programmschritte, um die eine Verzweigung erfolgen soll

Anwendung

k13,0	Warte, bis eine Taste gedrückt ist
k1,2	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 (Programmwahleinheit 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
```

Die Programmwahleinheit 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.	• 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 + 8388607. • 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	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. Die Referenzfahrt einer Schrittmotorachse wurde nicht korrekt ausgeführt.	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 *Achsenanzahl setzen*.

Fehlerbeschreibung	Ursachen	Beseitigung
4. Keine Achsen definiert	Bevor der Interface-Karte speicherbare Befehle, Bewegungen oder allgemeine Befehle, die eine von der Achsanzahl 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.	
5. Syntax-Fehler	• 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.	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.
6. Speicherende	Es wurde versucht, mehr Befehle zu übertragen, als die Interface-Karte speichern kann.	Teilen Sie das Programm in kleinere Abschnitte auf, übertragen Sie jeweils einen Abschnitt, führen Sie diesen aus und übertragen Sie dann den nächsten Abschnitt.
7. Unzulässige Parameterzahl	Die Interface-Karte hat mehr oder weniger Parameter für den Befehl erhalten, als benötigt werden.	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.
8. Zu speichernder Befehl nicht korrekt	Der Interface-Karte wurde ein Befehl zur Speicherung übergeben, der in dieser Form nicht vorhanden ist.	Prüfen Sie den übertragenen Befehl. Existiert der Befehlscode? Ist Groß-/Kleinschreibung berücksichtigt worden?

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 abzubrechen). 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 Unterroutine 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	als Rückmeldung wird Fehlercode 2 ausgegeben
	@0r7	zusätzlicher Fehlercode 2 am Ende der Bewegung
CNC-Mode	@0S	als Rückmeldung wird Fehlercode 2 ausgegeben Die Ausführung des Programmes wird abgebrochen
	@0s	zusätzlicher Fehlercode 2 am Ende der Bewegung Die Ausführung des Programmes wird abgebrochen
	Start-Taste	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 occured	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)