

UNIPROG+ V7-3x

**Betriebssystem zu Steuerungen
E-600 Base und E-600 ND**

Version: März 2006

INHALTSVERZEICHNIS

1	Vorwort	6
1.1	Merkmale von UNIPROG+	6
1.2	Kompatibilität mit UNIPROG	7
1.3	Kompatibilität mit POLYTOOL	7
1.4	Gebrauchsweise dieses Handbuches	7
2	SPEICHERORGANISATION	8
2.1	Der Speicherraum	8
2.2	Der Anwender-Speicherraum	8
3	Geometrie und Kinematik	9
3.1	Positions-Referenz	9
3.2	Koordinaten-Verschiebung	10
3.3	Kinematische Faktoren und Konstanten	12
3.3.1	Der Längenmaßstabsfaktor SCALEK	12
3.3.2	Höchstfrequenz, DIV-Parameter	12
3.3.3	Beschleunigungs-Konstante KUP, Brems-Konstante, KDN	13
3.3.4	Geschwindigkeitsfaktor FEEDK	14
3.3.5	Hubbegrenzung, STROKE + / STROKE –	14
3.3.6	Dauernde Koordinatenverschiebung, OFFSET	14
3.3.7	Stromerhöhungs-Parameter, BOOST	15
3.4	Kinematische Parameter der Servo-Achsen	15
3.4.1	Betriebsbereitschaft SERVO ENB. (Servo Enable)	15
3.4.2	Wertebereich des Positionierfehlers POS ZONE. (Position Zone)	15
3.4.3	Wertebereich des Schleppfehlers ALM ZONE. (Alarm Zone)	15
3.4.4	Verstärkungsfaktor LOW GAIN	15
3.5	Referenzprüfung	16
4	Handhabung und Dienstfunktionen	17
4.1	Netzschalter einschalten	17
4.2	Menu-Auswahl	19
4.3	Menu "OTHER"	20
4.3.1	Zugriffskennzeichen und Zugriffscode	20
4.3.2	VER, Versionsnummer	20
4.4	Menu "CONFIGURATION"	21
4.4.1	Konfiguration der Bewegungserzeuger (Motion GENerators)	21
4.4.2	Konfiguration der Referenz-Funktion (REFerence)	21
4.4.3	Konfiguration der Steuereingänge (ConTRoL)	21
4.4.4	Servoachsen-Einstellung (SERVo)	22
4.5	Menu "MOTION CONTROL"	23
4.5.1	Werkzeugtabelle "TOOL"	23
4.5.2	Handbewegung, JOG	23
4.5.3	Die Referenz-Prüfung, CLOS	25
4.5.4	Achsenstellungsanzeige, DISP	25
4.6	Menu "PROGRAMMING"	26
4.6.1	Auswahl des ausgeführten Programmes (start VECTors)	26
4.6.2	Vorschubsvorwahl (FEED rates)	26

4.6.3	Benutzerprogramm und Daten auf BEE-Karte (SAVE) speichern.....	26
4.7	Datei Handhabungen, FILE UTILITIES	27
4.7.1	DIR, Dateiverzeichnis	27
4.7.2	DEL, eine Datei Löschen	28
4.7.3	COPY, Datei kopieren.....	28
4.7.4	LOAD, lädt die gesamte BEE-Karte in den RAM	28
4.8	Menu "DEBUGGING"	29
4.8.1	Dienstfunktion "TRACE".....	29
4.8.2	Eingang/Ausgang-Funktion 'I/O CONTROL'	29
4.8.3	Ausdruck-Funktion, "PRT"	30
4.8.4	Dienstfunktion "SERVO"	30
5	Programmierbefehle	31
5.1	Positionsbefehle	31
5.1.1	Absolut-Positionierung :	31
5.1.2	Relativ-Positionierung :	32
5.1.3	Werkzeugauswahl.....	32
5.2	Weitere Kinematische Befehle	32
5.2.1	Referenz-Aufnahme:.....	32
5.2.2	Referenzkontrolle :	32
5.2.3	Lernbefehl:.....	33
5.2.4	Parameter-Festlegung	33
5.2.5	Bohrzyklus	33
5.2.6	Gewindeschneiden-Befehl	34
5.2.7	Polarkoordinaten.....	35
5.2.8	Winkelverschiebung.....	35
5.3	Eingangs- / Ausgangs-Befehle	36
5.3.1	Warten auf einen Eingang:.....	36
5.3.2	Bedingter Sprung :	36
5.3.3	Ausgangs-Festsetzung:	36
5.3.4	Ausgangs-Komplementierung.....	37
5.4	Spindelmotor-Ansteuerung.....	38
5.4.1	Spindeldrehzahl	38
5.4.2	Motorumschaltung und Spindeldrehzahl.....	38
5.5	Manipulationen numerischer Werte.....	38
5.5.1	Laden des Akkumulators	38
5.5.2	Abspeichern des Akkumulators.....	38
5.5.3	Inkrement /Dekrement	39
5.5.4	Speichern einer Variablen auf die Datenkarte.....	39
5.6	Programmablauf-Befehle.....	39
5.6.1	Unbedingter Sprung :	39
5.6.2	Unterprogramm-Aufruf:.....	39
5.6.3	Programm- und Unterprogramm-Ende :	39
5.6.4	Wiederholungen:.....	40
5.6.5	Aufruf von Simultan-Abläufen:	40
5.6.6	Bedingte Sprünge (Akkumulator-Test):.....	40
5.7	Verzögerung	40
5.8	Arithmetische Befehle.....	41
5.9	NOP und "Pseudo-Befehle"	41
5.10	Pausen-Kennzeichen	41

6	Der Programm-Editor	42
6.1	Lesen eines Programmes.....	42
6.2	Aendern eines Linieninhaltes	42
6.3	Einfügen und löschen einer Linie	43
6.4	Setzen eines Pausen-Kennzeichen	43
7	Programm-Ausführung	44
7.1	Die Ausführungsmodi nach Tasten MOD1, MOD2	44
7.2	Tastenfunktionen START, PAUSE, STOP	44
7.3	Fehlerbearbeitung	45
8	Erzeugung von Vektoren und Umrissen	46
8.1	Einführung : Möglichkeiten und Definition des Raumes.....	46
8.2	Erzeugung eines Vektors	46
8.3	Geometrische Definition einer Kontur	47
	8.3.1 Definition eines linearen Teilstückes.....	47
	8.3.2 Definition eines Kreisbogens.....	48
8.4	Bearbeitung der Umrissdateien.....	51
8.5	Längen-Begrenzung bei der linearen Interpolation	52
8.6	Begrenzung der gesamten Anzahl Segmente.....	53
8.7	Ausführung von Umrissen	53
8.8	Situationen, wobei die Werkzeugkorrektur nicht funktioniert.....	54
8.9	Anzeige der Bahnrechnungs-Fehler.....	55
8.10	Beispiel eines Umrisses	56
8.11	Zusammenfassung der Befehle und Pseudo-Befehle für die Erzeugung von Umrissen.....	57
9	Anschluss der E-600 Steuerungen	58
9.1	Einführung	58
9.2	E-600-base, Stromversorgung, Logik und Frontplatte	58
	9.2.1 I/O-Stecker.....	58
	9.2.2 I/O EXT -Stecker.....	59
	9.2.3 RS 232 -Stecker.....	59
9.3	Verträglichkeit mit E-500	59
9.4	E-600-1, Modul, BERGER-Treiber für 5-Phasen-Motoren.....	59
9.5	E-600-3 -Module, 2-Phasen-EIP-Motortreiber.....	60
	9.5.1 Stromeinstellung	60
10	Anschluss der Steuerung E-600-ND	61
11	Zusammenfassung	62
11.1	Befehle	62
11.2	Ein- und Ausgänge	64
11.3	Diverses.....	64

Abbildungen:

Abb. 3-1 : Verfahrenweg und Referenzstellung	9
Abb. 3-2 : Die Koordinatensysteme von UNIPROG+	11
Abb. 3-3 : Frequenz oder Geschwindigkeit zur Zeit	13
Abb. 4-1 : Menu "I/O Control"	29
Abb. 5-1 : Bohrzyklus	34
Abb. 8-1 : Umriss mit Durchmesser Null	48
Abb. 8-2 : Kreisbogen-Modi.....	49
Abb. 8-3 : Beispiel eines Umrisses.....	50
Abb. 8-4 : Modus gemäss Bewegungsrichtung und Warte-Position	54

Tabellen:

Tabelle 5-1 : Ein- und Ausgänge.....	36
Tabelle 5-2 : Adresse der E/A-Module	37
Tabelle 9-1 : E600, I/O-Stecker, 19-Wege-Burndy	58
Tabelle 9-2 : E600-1, Burndy 19-Wegeschalter	59
Tabelle 9-3 : E600-3, , BURNDY-8-Wege-Stecker	60
Tabelle 9-4 : Achsenbezeichnung	60
Tabelle 9-5 : E600-3, Stromeinstellung	60

1 Vorwort

Die E-600 Bewegungssteuerung richtet sich nach dem Marktbereich: Handhabungseinrichtungen und Spezial-Werkzeugmaschinen. Sie kann viele Bewegungs- und Automationsaufgaben einfach lösen. Sie kann mit bis 4 Zwei- und Fünf-Phasen Hybrid-Schrittmotorentreibern, oder bis 2 Servo-motorentreibern ausgerüstet werden, wobei 4 Achsen im Total angesteuert werden.

Die externen Anschlüsse sind im großen Umfang übereinstimmend mit der E-500 -Serie. Das Format der Speicherkarte ist jedoch **nicht direkt übereinstimmend**. Eine Umwandlung von E-500 Karten zu E-600 ist mit dem UNICOM-Programm möglich.

Das Gehäuse "**E-600 Base**" ist das Hauptbestandteil. Es beinhaltet die Stromversorgung, die Steuerungslogik, die Tastatur und die lokalen Ein- und Ausgänge und bietet Platz für verschiedene Treibermodule.

Erhältlich ist dazu eine sog. "No Drive"-Version, die "**E-600-ND**", ohne Gehäuse und ohne Leistungsteil, die zum Schrankeinbau bestimmt ist.

E.I.P. SA hat für anspruchsvolle Anwendungen eine eigene Sprache, "PINX-E" und ein Programmentwicklungswerkzeug, "APEX" entworfen. Für gängige Situationen liefert jedoch das **UNIPROG+-Programm**¹ einen Satz von Dienstfunktionen und die Möglichkeit, Maschinenprogramme direkt an der Steuerungstastatur zu schreiben.

Die Benutzerprogramme und die Maschinenkonfiguration können auf einer herausnehmbaren Speicherkarte (BEE-CARD) gesichert werden. Diese EEPROM-Karte ist ein sehr praktisches und zuverlässiges Speichermedium.

1.1 Merkmale von UNIPROG+

Das **UNIPROG+** -Betriebssystem selbst ist in PINX-E -Sprache geschrieben und so strukturiert, dass eventuelle spätere Anpassungen problemlos vorgenommen werden können. Es können so zum Beispiel anwendungsspezifische Befehle leicht nachträglich eingeführt werden.

- Das Betriebssystem UNIPROG+ V 7.30 dient dazu, die Systeme UNIPROG und POLYTOOL durch eine sehr ähnliche Ergonomie zu ersetzen.
- Es bringt darüber hinaus den Vorteil einer Bahnkorrektur, die dem Werkzeug-Radius und dem Umriss (Werkstückkontur) Rechnung trägt (Befehle LIN, CIR, POINT).
- Es ist aber ein Vorbehalt vorhanden, dass **der Umriss keine Eckpunkte aufweist**, alle Segmente und Bogen müssen tangierend sein. Dieser Nachteil wird durch das automatische Generieren des Radius und des Modus kompensiert, wenn mehrere Bögen aneinandergelagert sind ; dadurch wird der Befehl RAD überflüssig.
- Um die Programmierung zu vereinfachen, gestattet UNIPROG+ auch das Programmieren der Eckpunkte mit ihren Ausrundungen (früher POLYTOOL).
- Während der Berechnung der Bahn zeigen zahlreiche Meldungen bestimmte Fehler bei der Bahnsteuerung an. Andererseits werden im Modus 2 die Radien angezeigt, die in der Bahn vorkommen.

Für die **Installation von UNIPROG+** auf der E-600 Steuereinheit, die mit den Programmen UNIPROG oder POLYTOOL ausgestattet ist, möchten Sie bitte Abschn. 1.2 und 1.3 lesen, wenn die alten Bearbeitungsprogramme zu bewahren sind.

¹ Der Programmname darf nicht mit UNIPPLUS verwechselt werden, das von der Firma COMBITEC in Biel entwickelt wurde.

1.2 Kompatibilität mit UNIPROG

- 1) Wenn in den alten Dateien der Befehl 19 ORG nicht verwendet wird, dann muss der Parameter 'LAST TOOL NB' auf Null gestellt werden. Im Gegenfall muss die Datei '0' freigestellt werden, die für das Speichern der Ursprungspunkte der Werkzeuge verwendet wird. Als Folge muss sie in eine andere Datei kopiert und die Adressen angepasst werden, die sich auf die Datei '0' beziehen.
- 2) Um den Wert des DAC in % zu erhalten, wird der Parameter 'MAX RPM 10 VOLTS' auf 100 gesetzt. Andererseits wird im Programm das Anweisungspaar FLOAD / LDAC durch den Befehl 57 SPVEL ersetzt, von der Analogausgangsnummer und der Rotationsgeschwindigkeit nachgefolgt.
- 3) Die Winkeldefinition, die durch den Befehl CDEF gegeben wird, sollte durch die maximal erlaubte Abweichung ersetzt werden.
- 4) Um den START auf einem einzigen Eingang zu erhalten, muss der Parameter '2 HAND START' auf 8 gesetzt werden.
- 5) Um den Aufruf der Programme im Jogging zu sperren, muss der Parameter 'MANUAL PROGRAMME' auf 100 gesetzt werden (siehe CTRL und JOG).

1.3 Kompatibilität mit POLYTOOL

- 1) Die Datei '0', die für das Speichern der Werkzeug-Ursprungspunkte verwendet wird, geht von 4 auf 5 Werten pro Werkzeug über. Folglich werden die Ursprungspunkte des Werkzeugs verschoben.
- 2) Siehe die Punkte 2,3,4,5 des obenstehenden Abschnittes 'Kompatibilität mit UNIPROG'.
- 3) Die ersten zwei Befehle POINT werden durch die Befehle 40 ORGP ersetzt.
- 4) Der Kode 32 des Befehls POINT wird zum Kode 42 und der Modus 2 muss zur zweiten Koordinate hinzugefügt werden.
- 5) Der Befehl DTOOL verschwindet und das Argument links/rechts ist in dem Befehl DPATH hinzugefügt. Der angegebene Durchmesser muss in der Tabelle 'TOOL' gespeichert werden. Der Durchmesser bezieht sich auf die Werkzeugsnummer, die durch den Befehl TOOLP gewählt wird.
- 6) Der Parameter TOOL L/R INVERT muss sich den Achsenrichtungen anpassen.
- 7) Die Umriss sind immer offen und werden durch die Befehle LINA und LINR abgeschlossen.

1.4 Gebrauchsweise dieses Handbuches

Das Ziel dieses Handbuches ist, dem unerfahrenen Benutzer nach sorgfältigem Durchlesen das Arbeiten mit UNIPROG+ zu ermöglichen. Eine gewisse technische Kenntnis über Schrittmotoren und praktische Erfahrung mit Servoantrieben ist erforderlich, um zu vermeiden, sich durch Ausprobieren dem Problem zu nähern. Die Kenntnis der SPS ist aber nicht erforderlich.

- Dem Leser wird geraten, Kapitel 2 und 3 zu lesen, bevor er versucht, Programme zu schreiben oder Dienstprogramme zu benutzen.
- Anschluss-Informationen sind in Kapitel 9 zu finden. Ausführliche Angaben sind in den Betriebsanleitungen der einzelnen eingebauten Module zu finden.
- Kapitel 4 beinhaltet eine Beschreibung des Betriebsmodus der Tastatur, beginnend mit dem Einschalten des Netzschalters.
- Kapitel 8 behandelt die Erzeugung von Vektoren und Bahnen (Linear- und Kreis-Interpolation).

2 SPEICHERORGANISATION

2.1 Der Speicherraum

Der Speicherraum der E-600 Steuerung ist auf 4 Bausteinen verteilt:

Ein 32 kbyte- Festwertspeicher (EPROM), der sog. "untere Systemspeicher"

- Ein 32 kbyte FEPROM (Flash Memory), im Feld schreibbar, der sog. "obere Systemspeicher"
- Ein 32 kbyte C-MOS RAM mit Batterie-Datensicherung
- Ein 8 kbyte-EEPROM (schreibbar): die "Datenkarte"
- Im unteren Systemspeicher sind das PINX-E –Sprachen-Interpreter und Grundroutinen enthalten.

Durch Laden des FEPROM ist es jederzeit möglich die Steuerung in wenigen Minuten für Anwendungen "umzubauen", die UNIPROG+ nicht verwenden.

Das in beiden Systemspeichern enthaltene Betriebssystem kann nur mit Hilfe der APEX-Entwicklungsinstrumenten und eines EPROM-Programmiergerätes verändert werden. Der FEPROM wird aus einer speziellen Karte (64 k-SRAM oder EPROM-Adapter) geladen.

Die Datenkarte (EEPROM) dient zur Sicherung der Anwender-Programme in UNITOUR-Sprache und der Maschinen-Konfiguration.

Der RAM-Speicher ist der Arbeitsspeicher der Steuerung. Wird die Steuerung mit einer gesteckten Datenkarte eingeschaltet, dann wird den Gesamthalt der Karte in den RAM-Speicher übertragen. Ein Anwender-Programm wird immer ab RAM-Speicher abgespielt. Der Editor und die Konfigurations-Dienstfunktion legen ihre Daten in den RAM-Speicher ab, niemals in die Datenkarte. Die Sicherung des Anwender-Programms und der Maschinen-Konfiguration ist Aufgabe des Bedieners.

2.2 Der Anwender-Speicherraum

Mit Hilfe der UNIPROG+ -Dienstfunktionen kann der Anwender seinen Speicherraum belegen. Dieser Speicherraum wird dann als Ganzes auf der Datenkarte gesichert.

Innerhalb des Anwenderraumes ist ein fester Abschnitt der Maschinenkonfiguration gewidmet, siehe Abschnitt 4.4. Die etwa 7500 übrigen Bytes stehen für die Programme und die numerischen Daten des Anwenders zur Verfügung.

Die Anwenderdaten sind in 6-Bytes-« **Zeilen** » strukturiert. Eine Zeile enthält einen Befehl oder eine Zahl. 1250 Zeilen stehen zur Verfügung und können beliebig für « **Programme** » oder « **Dateien** » benützt werden. Die Bezeichnung « Programm » wird meistens für Befehlsdateien verwendet.

Bis 100 Dateien, nummeriert von 00 bis 99, können in den 1250 Zeilen eröffnet werden. Eine Datei wird durch die Editierfunktion oder durch Kopieren eröffnet.

Mit dem Umstecken der Datenkarte kann der Anwenderraum vervielfältigt werden.

3 Geometrie und Kinematik

In diesem Kapitel wird das von UNIPROG+ benutzte Koordinatensystem definiert. Es werden nützliche Informationen zu den Bewegungserzeugern geliefert.

3.1 Positions-Referenz

Schrittmotoren arbeiten inkremental, d.h. vor jedem Arbeitszyklus wird die Referenzposition aller Achsen festgelegt. Dabei treten zwei Fälle auf:

- das Koordinatensystem ist fest mit der Maschine verbunden (Beispiel : Palettisieretisch),
- der Nullpunkt des Koordinatensystems wird durch den Operator in einem von ihm gewählten Punkt festgelegt (Beispiel : Teilapparat).

Im Fall a) muss die Achse mit einem Referenz- oder Initiatorschalter versehen werden. Der Referenzschalter erlaubt die automatische und genaue Festlegung des Nullpunktes des **Referenzsystemes**.

Das Anpeilen des Nullpunkts kann vom Bediener oder vom Initialisierungsprogramm übernommen werden.

UNIPROG+ akzeptiert einen Schalter überall im Verfahrensweg

Falls der Schalter nicht am Ende des Verfahrensweges festgelegt wurde, muß er **an einer Seite des Schaltpunktes stetig geschlossen und an der anderen Seite stetig geöffnet sein**, damit die Steuerung unzweideutig erkennen kann, auf welcher Seite der Schlitten liegt, siehe Abb. 3-1.

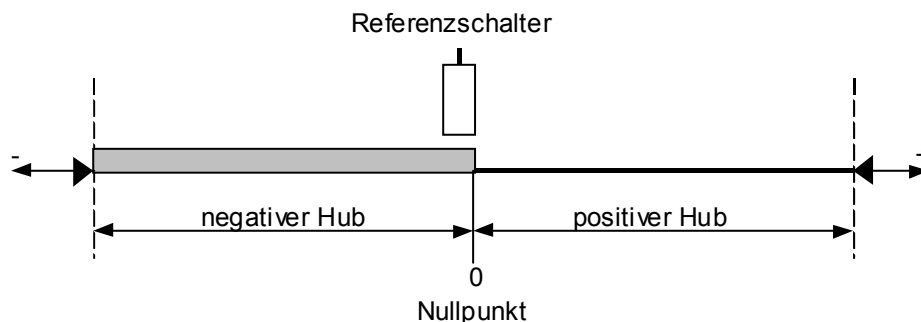


Abb. 3-1 : Verfahrensweg und Referenzstellung

Die Referenzfahrt wird in drei Phasen ausgeführt :

- Phase 1 : Diese Phase findet nur statt, wenn der Referenzschalter am Beginn des Vorgangs bereits aktiviert ist: Fahrt weg vom Schalter bei einstellbarer Geschwindigkeit und einstellbarer Bremsrampe.
- Phase 2 : Fahrt zum Schalter bei einstellbarer Geschwindigkeit; Bremsrampe sobald Schalter erreicht ist.
- Phase 3 : Fahrt weg vom Schalter im Kriechgang (einstellbares Anteil der Referenzfahrtgeschwindigkeit); sofortigem Stop, sobald Schalter nicht mehr betätigt ist.

- Die eingestellte Geschwindigkeit der Phase 3 bestimmt die Genauigkeit der Referenzstellung.
- Die E-600 Steuerung hat 8 Eingänge, die für Referenzschalter benutzt werden können: REF

INPUT 0..7. Die Zuweisung eines einzelnen Eingangs zu einer gegebenen Achse erfolgt im Konfigurationsmenu.

- Wenn eine Achse keinen Referenzschalter benutzt, muß sie konventionell zu REF INPUT 8 zugewiesen werden. Der Ursprung des Referenzsystems wird bei der Durchführung der Referenzfunktion ohne Bewegung festgelegt.
- Mit Servo-Antrieben kann der Indexkanal des Weggebers für einen genauen Nullpunkt genutzt werden. Eine ungefähre Referenzstellung mit einem Schalter ist dennoch nötig. In diesem Fall ist die REF INPUT-Zuweisung von 9 bis 17.
- Die Zuweisung REF INPUT 9 bedingt, dass der Referenzpunkt am Hubende liegt. Dann werden die Endschalter über das Modul E-600-4 / E-600-7 / E-600-12 angeschlossen und die dadurch erzeugte FAULT-Bedingung ausgenutzt.
- Mit dem Adapter E-600-8 für YASKAWA-Servoantriebe ist die Referenz mit Index möglich, mit der festgelegten Referenzschalter-Zuweisung: REF INPUT 0 für die X-Achse, 1 für Y, 2 für Z und 3 für U. Um den Endschalter als Referenzschalter zu nutzen, muß REF INPUT 18 eingestellt sein. Die selben Eingänge (0 bis 3) sind dann belegt.

REF INPUT	Modul	Eingang	Bemerkung
X=0, Y=1, Z=2, U=3	E-600-8	IN(0) à (3)	Referenzschalter + Index
0 à 7	Alle, ausser 4 bis 7 bei E-600-8	IN(0) à (7)	ohne Index
8	Alle	NA	Keine Bewegung
9	E-600-4/7	NA	Endschalter+ Index
10 à 17	E-600-4/7	IN(0) à (7)	Referenz + Index
18	E-600-8	IN(0) à (3)	Endschalter+ Index

- Die Eingabe der Hubbegrenzung STROKE+ und STROKE- im Konfigurations-Menu setzt die positive bzw. negative Grenzen des Bewegungsraumes der Achsen fest.
- Ist STROKE - = 0, dann sind nur Punkte mit positiver Ordinate im Referenz-System erreichbar, siehe Abb. 3-1. Die Richtung der Bewegungen bei der Referenzfahrt werden durch das Vorzeichen vom Parameter REF SPEED in der Konfiguration bestimmt.
- Wenn eine Bewegungsbegrenzung nicht geeignet ist, z.B. bei einem Drehtisch, müssen STROKE+ und STROKE- auf 0 gesetzt werden. Hier wird REF INPUT 8 für diese Achse benötigt.

3.2 Koordinaten-Verschiebung

Die Bewegungen der Achsen können als 'Relativmass' oder als 'Absolutmass' vorgegeben werden. Diese Begriffe haben ihre Bedeutung für tastengesteuerte Bewegungen im 'JOGGING'-Menu und für programmierte Bewegungen.

Das *Relativmass* wird ab gegenwärtigem Standort der Achse gemessen – das Koordinatensystem ist somit ohne Bedeutung.

Das *Absolutmass* ist die Koordinate des zu erreichenden Punktes. In diesem Fall hat das Koordinatensystem seine volle Bedeutung.

Die Koordinaten-Verschiebung steht zur Verfügung, die Koordinaten-Verdrehung jedoch nicht.

Koordinaten-Verschiebungen sind auf zwei Ebenen geboten :

- das **Basis-Koordinatensystem** wird ab Referenzsystem durch die Verschiebung 'OFFSET' verschoben, siehe Abb. 3-2.

- das **laufende Koordinatensystem** wird ab Basissystem durch die Verschiebung 'ORIGINE' verschoben.

Das Referenzsystem ist normalerweise nicht direkt zugänglich, d.h. es ist nicht möglich, die Ordinate ab Referenzpunkt zu messen (mit OFFSET = 0 sind Referenz- und Basissystem gleichwertig). Der Vektor OFFSET wird durch das Konfigurationsmenu eingegeben.

Der Verschiebungsvektor 'OFFSET' hat seine Bedeutung vor allem bei Maschinen mit einem festen Arbeitspunkt wie Bohrmaschinen, Stanzen, etc. Die Richtung des Vektors 'OFFSET' zeigt vom Referenz- zum Arbeitspunkt.

Die im 'JOGGING'-Menu definierten Absolutbewegungen werden im Basis-Koordinatensystem gemessen.

Das **laufende Koordinatensystem** wird durch den Befehl TOOL oder TOOLP definiert. Der Programmierer kann das laufende System so oft wie nötig verändern. Wenn kein TOOL- oder TOOLP -Befehl ausgeführt wird, sind Basissystem und laufendes System gleich. Änderungen des laufenden Systems sind vor allem in Anwendungen wie Mehrspindelmaschinen oder Handhabungsaufgaben mit einem Greifraum und einem Ablegeraum nützlich.

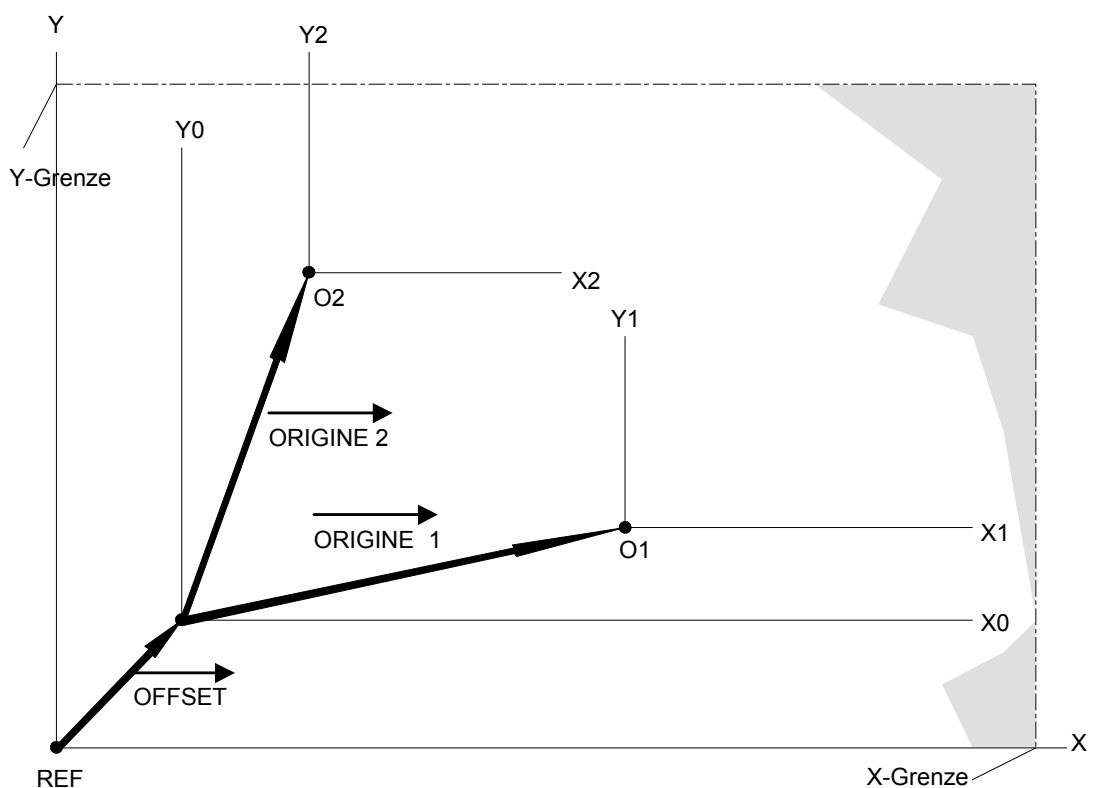


Abb. 3-2 : Die Koordinatensysteme von UNIPROG+

3.3 Kinematische Faktoren und Konstanten

Die in diesem Abschnitt besprochenen Faktoren und Konstanten sind für jede Achse im Konfigurationsmenu vorgegeben.

3.3.1 Der Längenmaßstabsfaktor SCALEK

Dieser Faktor ermöglicht die Programmierung der Bewegungen direkt in technischen Einheiten.

Für einen Schrittmotor ist SCALEK die Anzahl an Impulsen, die am Eingang des Treibers erforderlich sind, um eine Bewegungseinheit zu bewirken.

Der 2-Phasen Mikroschritttreiber (Leistungsstufe) E-600-3 benötigt 8 Impulse für einen ganzen Schritt. Bei den meisten Schrittmotoren (1.8 Grad/Schritt) ergeben 1600 Impulse genau einen Umlauf.

Berger 5-Phasen-Motoren und -Treiber (Modul E600-1) benötigen 500 oder 1000 Impulse pro Umlauf, abhängig von den Einstellungen an der Schalttafel.

Für einen Servomotor ist SCALEK die vierfache Anzahl an Impulsen, die von dem Kodierer während des Durchlaufs einer Längeneinheit erzeugt werden. (Vierfach, weil alle Übergänge des Quadratursignals gezählt werden).

Beispiele:

a) Von einem 1.8 Grad Schrittmotor angetriebener Leitspindelschlitten

Riemenübersetzung von Motor zur Spindel 1:2, Gewindesteigung 5 mm, Längeneinheit 1 mm.

1600 Impulse für eine Motorumdrehung

3200 Impulse für eine Leitspindelumdrehung

also $3200/5$ Impulse für 1 mm, **SCALEK = 640**

b) Drehtisch:

5-Phasenmotor 1000 Impulse pro Motorumdrehung, Motor von Schneckengetriebe

1:40 direkt angetrieben, Einheit: 1 Grad.

40.000 Impulse für eine Umdrehung des Tisches,

$40.000/360$ Impulse pro Grad, **SCALEK = 111.111** .

c) Riemenantrieb-Schlitten:

Servoantrieb mit 1000 Zeilenkodierer, Untersetzungsgetriebe 1:10, Transportriemen

Teilung 3 mm, Antriebsrad 35 Zähne, Längeneinheit 1 Inch.

$4 \cdot 1000$ Impulse für eine Motorumdrehung,

40.000 Impulse für eine Radumdrehung oder für 105 mm,

$105 \text{ mm} = 105/25,4 = 4,133858$ Inch

$40.000/4,133858 = 9676,19$ Impulse pro Inch,

SCALEK = 9676,19

3.3.2 Höchsthfrequenz, DIV-Parameter

Die Abb. 3-3 ist eine grafische Darstellung der Pulsrate (oder die Geschwindigkeit der Achsen),

die durch den Bewegungserzeuger während einer einzelnen Bewegung erzeugt wird. Die Geschwindigkeits-Zunahme und –Abnahme reduzieren sich linear mit der Geschwindigkeit, um eine Schwächung des Motordrehmoments auszugleichen.

Die Höchstgeschwindigkeit ergibt sich aus dem DIV-Parameter und muss für jede Achse experimentell so ermittelt werden, dass bei hohen Geschwindigkeiten eine genügend grosse Drehmomentreserve behalten wird.

Dazu ist die Geschwindigkeit auf die jeweils programmierte Geschwindigkeit begrenzt. Die Beschleunigungs- und Bremsrampen bleiben erhalten.

Bei Servomotoren ist es vorteilhaft, die Höchstgeschwindigkeit sehr hoch zu setzen, wobei die Geschwindigkeit immerhin durch den viel tieferen programmierten Wert begrenzt ist. Die Rampenkennlinien werden quasi-linear.

Wenn f_{max} die Pulsfrequenz zum Schrittmotorentreiber (oder 4 mal die Pulsfrequenz des Lagegebers bei Servo-Antrieb) bei der Höchstgeschwindigkeit, dann ergibt sich DIV durch :

$$\text{DIV} = 11906/f_{max} \text{ [kHz]}$$

Bei DIV = 120 ist ungefähr $f_{max} = 100$ kHz, bei 300, 40 kHz.

Beispiel:

500-Teilung-Drehgeber, Höchst-Geschwindigkeit 1500 U/min

1500 U/min = 25 U/sec,

$f_{max} = 25 \cdot 500 \cdot 4 = 50'000$ Hz,

DIV = 11906/50 = 238.

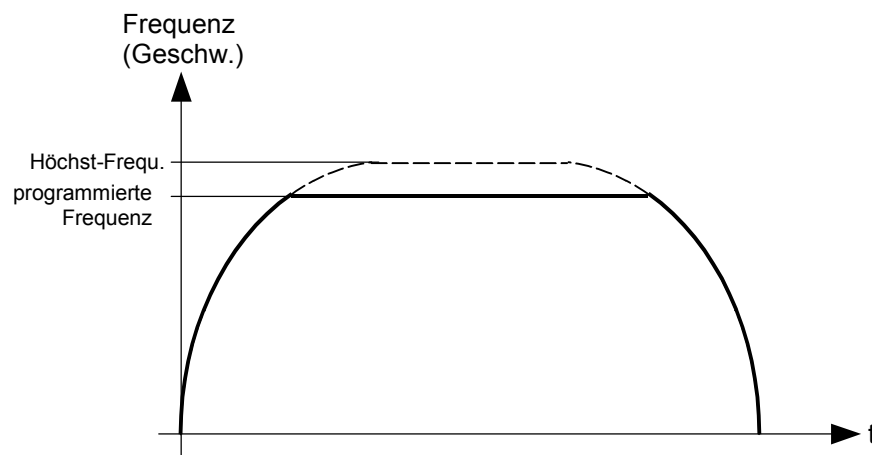


Abb. 3-3 : Frequenz oder Geschwindigkeit zur Zeit

3.3.3 Beschleunigungs-Konstante KUP, Brems-Konstante, KDN

Diese zwei Parameter definieren die Ursprungs- oder Endverläufe der Rampen. Ihre Werte werden in kHz/s oder kImpulse/s² eingegeben, wobei die Frequenz wiederum die Treiber-Schrittfrequenz oder das vierfache des Geberfrequenz ist.

Werte von 200 bis 10.000 kHz/s sind gewöhnlich zu benutzen.

3.3.4 Geschwindigkeitsfaktor FEEDK

Der Geschwindigkeitsfaktor - oder Vorschubkonstante – dient zur Eingabe der Vorschubwerte in technischen Einheiten: mm/s, m/min, U/s, etc.

Für Schrittmotorenantriebe:

$$\text{FEEDK} = \text{Impulsfrequenz für eine Vorschubeinheit [kHz]}$$

Für Servoantriebe:

$$\text{FEEDK} = 4 * \text{Geberfrequenz für eine Vorschubeinheit [kHz]}$$

Beispiele:

(siehe Beispiele in Abschnitt 3.3.1)

a) Der Vorschub wird in m/min ausgedrückt.

SCALEK = 640, d.h. 640 Impulse für 1 mm,

oder 640'000 Impulse für 1 m,

oder eine Frequenz von 640 kHz für 1 m/s,

640/60 kHz für 1 m/min, **FEEDK = 10,667**

b) Drehung in Grad pro Sekunde

SCALEK = 111.111; 111.111 Impulse für 1 Grad,

oder 0,111111 kHz für ein Grad/s, **FEEDK = 0,111111**

c) Geschwindigkeit in Yard/min

1 Yard = 36 Inch, 9676,19*36 für ein Yard,

für ein Yard/min ist die erforderliche Frequenz

$$\text{FEEDK} = \frac{9,6762 * 36}{60} = 5,806 \quad [\text{kHz}]$$

Der aktuelle Vorschub kann höchstens gleich groß wie die Höchst-Geschwindigkeit sein, die von DIV gesetzt ist. Die höchste erreichbare Geschwindigkeit, in der gewählten Einheit ausgedrückt, ergibt sich durch:

$\frac{11906 / \text{DIV}}{\text{FEEDK}}$

Der Vorschub ist nur in Übereinstimmung mit den programmierten Werten, wenn der Schalttafelpotentiometer vollständig im Uhrzeigersinn aufgedreht (im Uhrzeigersinn) ist, insofern das Potentiometer betriebsbereit ist (Leuchtdiode hell), was bei jeder Bewegung ausser Schliessungskontrolle und Referenzfahrt gilt.

3.3.5 Hubbegrenzung, STROKE + / STROKE –

Diese in der Längeneinheit des Abschnitts 3.3.1 angegebenen Parameter setzen die gestatteten positive, bzw. negative Achsenhübe bei Positionier- und Tastbewegungen.

3.3.6 Dauernde Koordinatenverschiebung, OFFSET

Die Bedeutung dieses Parameters geht aus Abb. 3-2 hervor. Er muß in der Anwendungs-Längeneinheit eingegeben werden.

3.3.7 Stromerhöhungs-Parameter, BOOST

Dieser Parameter bestimmt die Wirkung der /BOOST-Leitung und ist auf die Werte 0,1,2,3 begrenzt.

- BOOST = 0 : /BOOST ist immer unaktiv (hoch).
- BOOST = 1 : /BOOST ist während einer Bewegung aktiv (tief) und unaktiv, wenn die Achse nicht arbeitet
- BOOST = 2 : /BOOST ist immer aktiv (tief).
- BOOST = 3 : /BOOST ist hoch während einer Bewegung und tief bei Stillstand

Mit dem E600-3 -Treiber ist BOOST normalerweise auf 1 gesetzt; so ist der Strom während der Bewegung nominell und wird im Stillstand zu etwa 60 % des eingestellten Wertes reduziert.

Es ist möglich, mit kleinen Motoren BOOST auf 0 zu setzen, oder BOOST auf 2 zu setzen, wenn das volle Drehmoment im Stillstand benötigt wird.

Mit dem 5-Phasentreiber E-600-1 kann BOOST verschiedene Funktionen haben, übereinstimmend mit den Einstellungen am Berger-Leistungsmodul oder an der Anschlussplatine, siehe E-600-1 -Handbuch.

Bei Servo-Antrieb ist BOOST bedeutungslos.

3.4 Kinematische Parameter der Servo-Achsen

Es sind zwei Servo-Kanäle vorhanden, CHANNEL 0 und CHANNEL 1. Im Rahmen von UNIPROG+ entspricht CHANNEL 0 zur X-Achse und CHANNEL 1 zu Y.

Unter beiden Kanäle (CHANNEL) wird über Taste F1 gewählt.

3.4.1 Betriebsbereitschaft SERVO ENB. (Servo Enable)

Bei Wert 0 ist die X-Achse im Schrittmotor-Betrieb; bei Wert 1 ist sie im Servo-Betrieb. Ebenso bei der Y-Achse.

3.4.2 Wertebereich des Positionierfehlers POS ZONE. (Position Zone)

Die Achse ist positioniert, wenn der Absolutwert des Positionierfehlers kleiner oder gleich dem POS ZONE-Wertes ist. Die Positionierung endet erst, wenn diese Bedingung erfüllt ist.

Wertebereich des Parameters: 0... 32'767 [Geberübergänge] (4 Uebergänge = 1 Impuls)

3.4.3 Wertebereich des Schleppfehlers ALM ZONE. (Alarm Zone)

Wenn der Absolutwert des Schleppfehlers den ALM ZONE-Wert überschreitet, hält die Achse an und die Fehlerbedingung wird gemeldet.

Wertebereich des Parameters: 0... 32'767 [Geberübergänge] (4 Uebergänge = 1 Impuls)

3.4.4 Verstärkungsfaktor LOW GAIN

Numerische Verstärkungsfaktor der Lageregelung.

Beim Gleichstrommotor-Verstärker-Modul E600-12, Testpunkt TP4, hat das Analog-Schleppfehlersignal den Wert :

$$AERR = NERR * 5/500 * GAIN/2^{16} \quad [VOLT]$$

$$= \text{NERR} * \text{GAIN} * 1.525 * 10^{-7}$$

wobei

NERR : Schleppfehler [Geberübergänge] (4 Uebergänge = 1 Impuls)

GAIN : Wert von LOW GAIN

- Im SERVO-Testmenu wird der doppelte Spannungswert angezeigt
- HI GAIN ist ein vorbehaltener Parameter für Sondersystemversionen und hat keine Funktion in Standard-UNIPROG+.

Siehe auch die Anleitung zu E600-12.

3.5 Referenzprüfung

Schrittmotorenantriebe arbeiten meistens ohne Rückmeldung. Es ist aber oft wünschenswert, Störsituationen, die Positionsfehler verursachen, melden zu können, indem die Positionsreferenz periodisch geprüft wird.

Dies erfolgt, indem der Schlitten zurück zum Referenzpunkt fährt und den Schalter abtastet. Da wird verglichen, ob die von der Steuerung gezählte Absolutposition mit dem Anfangswert innerhalb einer gegebenen Toleranz übereinstimmt.

Diese Toleranz wird im Konfigurationsmenu unter "CLOSURE GAP" in der Anwender-Einheit eingegeben.

Die Prüfung erfolgt in der gleichen Richtung wie bei der Referenz-Aufnahme, um allfällige Spiel- oder Hysterese-Einflüsse zu beseitigen. Ueberschreitet die hervorgehende Abweichung die Toleranz, so stoppt die Programm-Ausführung und wird die Abweichung angezeigt. Auf jeden Fall liegt die Achse anschliessend am Referenzpunkt.

Diese Funktion ist nur bei den REF INPUT –Zuweisungen 0 bis 7 möglich.

Jedoch beim Modul E-600-8 sind nur REF INPUT 4 bis 7 gestattet (ohne Index-Referenzierung).

4 Handhabung und Dienstfunktionen

Dieses Kapitel beschreibt den Betrieb der E-600 Steuerung unter UNIPROG+. Die Beschreibung beginnt mit dem Einschalten und setzt voraus, daß alle Verbindungen nach außen eingerichtet sind. Die Menus werden der Reihe nach behandelt, wie sie für eine erste Einstellung der Steuerung erforderlich sind. Die Programmierung, der Editor und das Austesten sind Gegenstand der folgenden Kapitel.

Für die Tastenbezeichnungen beziehe man sich auf das Diagramm, das diesem Handbuch beigelegt ist.

4.1 Netzschalter einschalten

Die Anzeige zeigt die Version der Betriebssysteme. Dieser Bildschirm kann angehalten werden, indem man die Taste F2 sofort nach dem Einschalten drückt.

```
* UNIPROG+ V x.xx *  
INT x.xx NEC x.xx
```

Nach dem Laden des UNIPROG+ Programmes zeigt die E-600 Steuereinheit wie folgt an :

```
RAM FEHLER  
FORMATIERT JA NEIN
```

Nach dem Laden mit **NEIN antworten**, damit der Parameter 'LAST TOOL NB' konfiguriert werden kann und der Verlust der Datei '0' verhindert wird.

- Durch die Antwort NEIN wird der Parameter 'LAST TOOL NB' zum Öffnen der Datei 0 nicht verwendet und die Kontrolle für das Löschen des RAM-Speichers wird nicht aktualisiert.
- Durch die Antwort JA wird die Datei geöffnet und sein Inhalt gelöscht. Die RAM-Kontrolle wird aktualisiert.

An dieser Stelle sind zwei Situationen möglich:

1. Der Kartenschlitz ist leer (oder beinhaltet eine unformierte oder eine nicht unter UNIPROG+ formierte Karte)

Die folgende Nachricht wird für ein paar Sekunden angezeigt

```
LADEN  
0.00, WARTEN
```

dann

```
Karte nicht formatiert
```

eine Taste druecken

In diesem Abschnitt weiß der Bediener, daß die Karte nicht in das RAM geladen wurde, also bleibt der vorherige Inhalt des RAM erhalten und wird für die folgende Operation der Steuerung benutzt.

Das Drücken irgendeiner Taste startet das Initialisierungsprogramm (POWER-ON –Program, siehe Kapitel 7.

Menu 1 wird angezeigt:

1 MOTION CONTROL :
TOOL JOG CLOS DISP

2. Eine gültige Karte ist eingesteckt:

Während der Karteneingabe zeigt der Bildschirm:

LADEN
x.xx Warten

x.xx ist das Kennzeichen der Karte, das beim Sichern der Daten gegeben wurde.

Das Initialisierungsprogramm wird gestartet und Menu 1 wird angezeigt:

1 MOTION CONTROL:
TOOL JOG CLOS DISP

Während der Ausführung eines Programmes hat der Bediener Zugang zu allen Menus und er kann die Dienstfunktionen benutzen. Ein Programm kann durch Drücken des STOP-Knopfes (rot) angehalten werden.

4.2 Menu-Auswahl

Die Pfeiltasten (↓ oder ↑) werden zum Auswählen der folgenden Menus benutzt.

1 MOTION CONTROL: TOOL JOG CLOS DISP

2 PROGRAMMING EDIT VECT FEED SAVE

3 DEBUGGING TRACE PRT SERV I/O

4 FILE UTILITIES DIR DEL COPY LOAD

5 CONFIGURATION SERV MGEN REF CTRL

6 OTHER VER ACCES

Auf Anfrage wird ein sog. Menu 0 zur Klartext-Eingabe von Parameter hinzugefügt, die in Datei 0 gespeichert werden. Die Texte und Zahlentypen (Ganze oder Reell) werden in PINX-E - Sprache programmiert, bitte rückfragen.

Ein Menu bietet bis zu vier Funktionen an. Ist nur eine vorhanden, wird sie mit der ENTER-Taste gewählt. Sind es mehrere, wird die gewünschte durch der entsprechenden Funktionstaste, F1 ... F4, gewählt. **Die ESC-Taste ist immer aktiv, um das Untermenü zu verlassen.**

Wenn die Meldung

NO ACCESS press any key

(kein Zugriff) bei dem Versuch, eine Funktion einzugeben, erscheint, so wird der Zugriff zu dieser Funktion nicht gewährt, siehe Abschnitt 4.3.1.

4.3 Menu "OTHER"

4.3.1 Zugriffskennzeichen und Zugriffscodes

Um ausgewählte Zugriffe zu gewähren, können individuelle Zugriffskennzeichen den Funktionen zugewiesen werden. Z.B.: Der Bediener soll Zugriff zum Tastbewegungs-Menü haben, jedoch nicht zum Editor.

Wie auch immer der Zustand der Zugriffskennzeichen ist, die Eingabe des Zugriffscodes gewährt den generellen Zugriff. Nach dem Einschalten gibt es keinen Zugriff zu den Funktionen, die mit einem Kennzeichen versehen sind.

Um einen generellen Zugriff zu haben, das Zugriffsmenü ACCES wählen. Die Meldung "SCHLUESSEL EINGEBEN" verlangt vom Bediener den Code

31415

(Generell wird eine Dateneingabe mit ENTER beendet, Tippfehler können mit CLR korrigiert werden).

Um zur Menuauswahl zurückzukehren, ESC drücken.

Bei der Eingabe des Zugriffscodes, während der generelle Zugriff gewährt ist, werden alle Funktionen geschützt, die mit einem Zugriffskennzeichen versehen sind.

Um individuelle Zugriffskennzeichen zu setzen, wählen Sie das ACCESS-Menü und geben Sie den Code, wie oben beschrieben, ein. Die Funktionen - oder Funktionsgruppen- können mit den Pfeiltasten ausgewählt werden. Die Eingabe "1" bewilligt den Zugriff, bei "0" ist die Funktion nur nach Eingabe des Codes verfügbar.

4.3.2 VER, Versionsnummer

Während des Drückens der Taste F1 werden die Versionsnummern der installierten Systeme angezeigt.

Diese Daten können sich für Wartungszwecke als wertvoll herausstellen.

4.4 Menu "CONFIGURATION"

4.4.1 Konfiguration der Bewegungserzeuger (Motion GENERators)

Mit der Taste F2 erreicht man dieses Untermenü, gestartet aus dem Konfigurationsmenü. Die Parameter für die Spezifizierung der Steuerung sind in rechtwinkliger Anordnung organisiert: vertikal wählen die Pfeiltasten den physikalischen Parameter; horizontal spezifizieren die Achsentasten (X, Y, Z, U) die Achse, die dem jeweiligen Parameter angehört.

Alle diese Parameter wurden in Kapitel 3 besprochen und sind hier kurz zusammengefasst:

DIV	Frequenzteiler
KUP	Geschwindigkeitszunahme
KDN	Geschwindigkeitsabnahme
SCALEK	Längenmaßstabsfaktor
FEEDK	Vorschubfaktor
STROKE+	Hub in positive Richtung
STROKE-	Hub in negative Richtung
OFFSET	gemeinsamer Verschiebungsvektor
BOOST	Stromerhöhung, siehe Unterlagen zu den Treibern

4.4.2 Konfiguration der Referenz-Funktion (REFerence)

Das REF-Untermenü wird mit der Taste F3 erreicht. Wie zuvor sind die Parameter in tabellenförmiger Anordnung vorgestellt.

SPEED TO REF	Geschwindigkeit der Achse beim Suchen des Nullpunktes. Geben Sie die Geschwindigkeit in technischen Einheiten, wie sie bei FEEDK definiert sind, ein. Ein Minuszeichen wechselt die Richtung .
CLOSURE GAP	Referenz-Prüfungs-Toleranz, siehe Absch. 3.5.
REF. INPUT	Zuweisung der 8 Eingänge, IN (0) bis IN (7) als Referenzschalter-Eingang. Für eine Achse ohne Referenzschalter den Wert 8 eingeben. Für eine Achse mit Index, von 9 bis 17, siehe Abschnitt 3.1.
SWITCH	Art des Referenzschalters. "1" für einen Schliesser, "0" für einen Oeffner, siehe Abschnitt 3.1.
REF SPEED BACK	Teiler der Referenz-Geschwindigkeit für das Verlassen des Schalters. Je grösser der Teiler, desto höher die Genauigkeit.

4.4.3 Konfiguration der Steuereingänge (ConTRoL)

Dieses mit Taste F4 erreichbare Untermenü weist Eingänge den folgenden Programmausführungs-Funktionen zu. Diese Zuweisung ist logisch parallel zur Tastatur, siehe auch Kapitel 7 und Abschnitt 5.3.

Wird ein externer Steuereingang nicht benötigt, wird 64 eingegeben.

EXTERNAL START	Programm-Start, Schliesser-Schalter
EXTERNAL PAUSE	Programm-Halt, Oeffner-Schalter

EXTERNAL STOP Programm-Abbruch, Oeffner-Schalter

Diese Untermenu beinhaltet noch folgende Parameter, woraus manche neu mit UNIPROG+ eingeführt wurden:

- DISPLAY FORMAT 1-6.** Setzt die Anzahl der Nachkommastellen der Steuerungsanzeige für die Achsenpositionen.
- 2 HANDS START** ermöglicht, den Start mit zwei Händen auf zwei gewählten Eingängen zu konfigurieren. Parameter: EXTERNAL START und '2 HANDS START' (Wert 0 bis 8).
- In diesem Fall sind nur die Eingänge 0 bis 7 für diese zwei Parameter verwendbar.
 - Der Wert 8, wenn er im Parameter '2 HANDS START' eingeführt wird, stellt den einfachen Start wieder her.
- MANUAL PROGRAMME** ermöglicht den Start eines Programmes mit einer Taste des numerischen Tastaturfeldes zu verbinden. Vom Jogging-Menü aus (nach dem Verschwinden des Cursors). Wert von 0 bis 100.
- LAST TOOL NB** (Letzte Werkzeugnummer): ermöglicht, die Anzahl der verfügbaren Werkzeuge zu begrenzen, damit der Programmspeicher nicht unsonst gefüllt wird. (Wert von 0 bis 63).
- MAX RPM 10 volts** ermöglicht die Spindelgeschwindigkeit für 10 Volt am Eingang des Frequenzwandlers (Wert 0 bis 99999) einzustellen.
- LAST DELAYED OUT** legt eine Anzahl von Ausgängen fest, deren Rücksetzung verzögert wird (Wert von 0 bis 7).
- Der Parameter hat die Werte 0 bis 7, er gibt die totale Anzahl der verzögerten Ausgänge von 0 aus an.
 - Mit 0 ist kein Ausgang verzögert. Mit 7 sind die Ausgänge 0,1,2,3,4,5,6 und 7 verzögert.
 - Der Vorgabe-Verzögerungswert ist 1 Sek., dieser Wert kann im Programm geändert werden (Siehe SET BRK_D).
- TOOL L/R INVERT** ermöglicht, die Übereinstimmung der Lage des Werkzeugs gegenüber dem Werkstück beim Bahnfahren mit Werkzeugkorrektur wieder herzustellen. Mit dem Wert '0' wird die Achsenrichtung als normal angenommen : die erste Achse positiv nach rechts und die zweite nach oben (Wert von 0 bis 1).
- LANGUAGE** Sprachenwahl für bestimmte Meldungen: 0 = Englisch, 1 = Französisch, 2 = Deutsch. (Wert von 0 bis 2).

4.4.4 Servoachsen-Einstellung (SERVo)

Dieses Untermenu ist Aus Menu CONFIGURATION, durch Taste F1 erreichbar.

Wie bei Menu MGEN sind die Parameter tabellenförmig angeordnet: vertikal wählen die Pfeiltasten den Parameter; horizontal wählt die Taste F1 den Kanal, die dem jeweiligen Parameter angehört.

- Es sei daran erinnert, dass die X- und Y-Achsen den Kanälen 0 bzw. 1 entsprechen.
- Alle diese Parameter wurden in Kapitel 3 besprochen und sind hier kurz zusammengefasst:

SERVO ENB Betriebsbereitschaft des Kanals

POS ZONE	Positionier-Spanne
ALM ZONE	Schleppfehler-Spanne
LOW GAIN	Verstärkungsfaktor

4.5 Menu "MOTION CONTROL"

Mit diesem Menu können vier Funktionen aufgerufen werden, die die tastengeführte Bewegung (JOG) und die Achsenlage-Anzeige einbeziehen.

Die Funktionen TOOL (Werkzeugtabelle), JOG und CLOS (Closure check) sind während der Ausführung eines Programmes nicht aufrufbar. In diesem Falle erscheint die Meldung :

PROG. IN EXECUTION
eine Taste druecken

4.5.1 Werkzeugtabelle "TOOL".

In diesem Untermenü hat man Zugang zu den Verschiebungen und dem Durchmesser der Werkzeuge. Zu jedem Werkzeug gehören 4 Ursprungskordinaten und ein Durchmesser für die Korrektur der Werkzeugsbahn.

Die Werte, die in dieser Werkzeugtabelle sichtbar sind, werden in der Datei '0' gespeichert, die automatisch geöffnet ist, wenn man den Parameter 'LAST TOOL NB' einstellt. Wenn irrtümlicherweise die Datei '0' nicht die passende Grösse aufweist, erscheint die folgende Meldung :

DATEI 0 NICHT FORMAT
eine Taste druecken

In diesem Falle muss der Parameter 'LAST TOOL NB' neu eingestellt werden.

4.5.2 Handbewegung, JOG

In dieser Dienstfunktion wird in der oberen Zeile die Stellung der gewählten Achse im Basissystem, in der unteren Zeile das Weginkrement angezeigt, welches bei jedem Drücken auf die Tasten JOG+ und JOG- ausgeführt wird.

Die Referenzaufnahme ist wie bei Abschnitt 4.5.1 vorhanden.

- **Wirkung der Tasten JOG+ und JOG-**

Diese Tasten lösen eine Bewegung aus. Das angezeigte Weginkrement wird nur bei dauerndem Drücken der 'JOG'-Taste ausgeführt. Bei vorzeitigem Loslassen der Taste wird die Bewegung mit Rampe gestoppt und der Vorgang ist beendet. Ein weiteres Drücken fängt mit einem vollen Weginkrement an.

- **Wirkung der Tasten F3 et F4**

Mit diesen Tasten wird das angezeigte Weginkrement in Zehnerpotenzen ausgewählt (10^{-3})

bis 10^3 Einheiten).

- **Wirkung der Pfeiltasten**

Mit den vertikalen Pfeilen wird der Cursor (ganz rechts in der Anzeige) von einer Zeile zur anderen umgeschaltet. Ist der Cursor unten kann mit der numerischen Tastatur das Weginkrement eingegeben werden (anstelle der Zehnerpotenzen).

Ist der Cursor oben, kann die Ordinate im Basissystem eingegeben werden. Durch Drücken der taste ENTER nach dem Eintippen wird eine Bewegung gestartet, die beim Loslassen oder Erreichen der Ordinate stoppt. Wird die Taste vorzeitig losgelassen, muss die Ordinate neu eingegeben werden.

- **Einstellung der Werkzeug-Ursprungspunkte**

Dieses Hilfsfunktion ermöglicht, ausser seiner Funktion der manuellen Bewegungen das Speichern der Ursprungspunkte von 64 verschiedenen Werkzeugen (von 0 bis 63). Die Anzahl der verwendeten Werkzeuge ist durch den Parameter 'LAST TOOL NB' festgelegt.

- Die Auswahl der Werkzeugnummer erfolgt mit den Tasten 'F1 und F2'
- Die neuen Ursprungspunkte werden in der offenen Datei 0 automatisch gespeichert (diese enthält die 'FDATA'). Um diese Ursprungspunkte leicht untersuchen zu können, ist eine Werkzeugtabelle mit dem Namen 'TOOL' vom Menü 'MOTION CONTROL' aus zugänglich.
- Diese neuen Ursprungspunkte werden bei einer Umrissdatei durch den Befehl 47, TOOLP, sonst durch den Befehl 19 TOOL hervorgerufen.
- Ein einziges Argument folgt dem Befehl TOOL (Werkzeugnummer). Zwei Argumente folgen dem Befehl TOOLP (Werkzeugnummer und Nummer der Umrissdatei).

- **Diverses**

- Die Taste 'PROG LINE' ruft den Editor auf, wo dieser das letzte Mal verlassen wurde.
- Während des Programmablaufes ist es möglich, im Menü JOGGING zu verbleiben, die Bewegungen sind aber gesperrt.
- Die numerischen Tasten von 0 bis 9 starten ein Anwender-Programm.
- Die Zuweisung eines Programmes einer Taste wird durch den Parameter MANUAL PROGRAMME im Konfiguration-Menü CTRL festgelegt. Die Nummer des ausgeführten Programmes ist die Summe des Parameterwertes und der numerischen Taste.

Zum Beispiel :

MANUAL PROGRAMME = 90

Die Taste 0 startet Programm 90,

Die Taste 1 startet Programm 91, usw.

Wenn der konfigurierte Wert 90 überschreitet, dann gibt es keine manuelle Programme mehr.

Diese manuellen Funktionen sind nur im Jogging-Menü aktiv (JOG) und wenn die rote LED STOP und die grüne LED START erloschen sind.

- **Um das Bewegung-Inkrement zu verändern** oder eine Absolut-Bewegung im Jogging auszuführen, muss man den blinkenden Cursor mit Hilfe der Pfeiltasten (unten und oben) holen und ihn über den zu ändernden Wert setzen.
 - Wenn der Cursor verschwindet sind die manuellen Funktionen aktiv.
 - Als Vorgabe befindet sich der Cursor im Inkremental-Modus.
 - Die Taste '-' ist der Taste '0' gleichgesetzt.
- **Das Speichern einer Position erfolgt auf folgende Art :**
 - 1) Annähern der Position im Jogging.

- 2) Wahl des Werkzeugs durch 'F1' oder 'F2' und der entsprechenden Achse.
- 3) Drücken der Taste 'F5' um die Eingabe der neuen Ursprungskoordinate einzuleiten.
- 4) Eingabe des Wertes auf dem numerischen Eingabefeld, dann Bestätigen durch die Taste 'ENTER'.

Anmerkung :

Für kleine Anpassungen kann die Korrektur des Ursprungspunktes auf inkrementale Art mit den Jogging-Tasten '+' und '-' eingegeben werden. Der unten rechts angezeigte Wert (das Inkrement) wird vom Ursprungspunkt abgezogen oder hinzugefügt. Dies erfolgt ohne eigentliche Bewegung.

- **Werkzeugverschiebung**

Unten links steht das Funktions-Kennwort 'ALL'. Wenn die entsprechende Taste (F1) betätigt wird, wird die Ursprungskorrektur auf alle Werkzeuge angewandt. Dieser Modus wird durch das Blinken der LED F1 und durch den Text 'ADJ ALL' (alles anpassen) bestätigt :

AXIS X = 87.677
ADJ ALL .100000

In dieser Lage verschiebt jede Betätigung der Jogging-Tasten '+' oder '-' den Ursprungspunkt aller Werkzeuge, wobei 0.1 mm hinzugefügt bzw. abgezogen wird. Der Inkrementalwert kann durch 'F1' oder 'F2' verändert werden.

Die Einstellung der Ursprungskordinaten kann auch von der Werkzeugstabelle 'TOOL' vorgenommen werden.

4.5.3 Die Referenz-Prüfung, CLOS

Die manuelle Positions-Referenz-Prüfung (siehe Absch. 3.5) ist eine wertvolle Hilfe für die Prüfung einer Anlage. Nach der Achsen-Auswahl genügt ein Druck auf die Taste REF, um diese Kontrolle auszulösen.

Erscheint die Meldung

CLOSURE WITHIN GAP
press any key

ist die Referenz in Ordnung. Ansonsten wird ein Positionsfehler angezeigt.

POS ERROR 1.234
press any key

Ohne Referenzschalter oder bei Index-Referenz wird die Kontrolle als richtig gemeldet.

Nach jeder Schliessungskontrolle, d.h. auch bei einem Positionsfehler, befinded sich die Achse beim Referenzschalter.

4.5.4 Achsenstellungsanzeige, DISP

Im Gegensatz zu TOOL, JOG, CLOS kann diese Funktion immer aufgerufen werden, d.h. auch während der Ausführung eines Programmes. Die vier Achsen werden dabei Angezeigt. Anzeigeformat : siehe Abschnitt 4.4.3

Das laufende Koordinatensystem ist massgebend für die Positionsanzeige.

4.6 Menu "PROGRAMMING"

Dieses Menu enthält alle Funktionen, die für das Editieren, die Ausführung und die Sicherung der Programme nötig sind. Der Editor ist in Kapitel 6 beschrieben.

4.6.1 Auswahl des ausgeführten Programmes (start VECTors)

In diesem Untermenu wählt der Bediener zwei Programme

- das "POWER-ON" (Einschaltprogramm), das gleich nach dem Einschalten ausgeführt wird,
- das Startprogramm, das bei erloschener STOP-Leuchte nach jedem Drücken des START-Knopfes beginnt (oder nach jedem Ansprechen des im Konfiguration-Menu festgelegten "Start"-Eingangs).

Darüber hinaus wird das "POWER-ON"-Programm nach jedem Vollstop der Steuerung (STOP-Leuchte hell) ausgeführt. Ist dieses Initialisierungsprogramm nicht erwünscht, weisen Sie einfach die Programmnummer 100 dem "Power-On"-Programm zu.

4.6.2 Vorschubsvorwahl (FEED rates)

Die Vorschübe können in einer Tabelle gespeichert werden. Sieben Werte, von 0 bis 6, werden dabei im voraus eingegeben. Die Einheit dieser Geschwindigkeiten ist durch die Konstante 'FEEDK gegeben.

4.6.3 Benutzerprogramm und Daten auf BEE-Karte (SAVE) speichern

Die Maschinenkonfiguration und alle geöffneten Dateien werden auf die Karte geschrieben. Die Zeit, um die Daten zu schreiben, ist sehr von der Menge der auf der Karte neuzuschreibenden Daten abhängig.

Um ein Überschreiben wertvoller Daten zu vermeiden, erscheint die Warnung

AUF KARTE SICHERN 2.23 ? JA NEIN
--

Die Kennungsnummer (hier 2.23) ist der aktuelle Code der eingesteckten Karte. Wenn der Bediener die Daten ohne Codewechsel schreiben möchte, muß er F3 drücken und die Speicherung beginnt. Um den Code der Karte zu ändern oder den Vorgang abzubrechen, antwortet der Bediener "NEIN" durch F4.

NAME AENDERN ? NEIN

F4 bricht die Speicherung ab. Die Eingabe eines Codes -oder NAME- startet nach Drücken von ENTER das Schreiben der Karte.

Während des Sicherns erscheint eine Wartemeldung. Die Zeit, um die Daten zu schreiben, ist von den Daten abhängig und kann möglicherweise ein paar Minuten dauern. Der Abschluss wird gemeldet:

SICHERN GUELTIG
eine Taste druecken

oder

SICHERN UNGUELTIG
eine Taste druecken

Diese letzte Meldung weist auf einen Schreibfehler hin (fehlende Karte oder defekte Karte).

4.7 Datei Handhabungen, FILE UTILITIES

Die Dienstprogramme spielen sich immer auf dem RAM ab; um eine Datei auf der BEE-Karte zu ändern, laden Sie zuerst die Karte in den RAM, wie in Abschnitt 4.7.4 erklärt.

4.7.1 DIR, Dateiverzeichnis

Der Bildschirm liefert Informationen über alle geöffneten Dateien. Eine Datei kann durch den Editor oder eine existierende Datei geöffnet werden.

FILE	SIZE	PROT	FREE
12	45	NO	670

Das obere Beispiel bedeutet:

- Datei 12 ist geöffnet
- deren Größe sind 45 Zeilen
- sie ist nicht geschützt
- es sind noch 670 Zeilen im Benutzerabschnitt frei

Wenn die verlangte Datei nicht existiert (nicht geöffnet) ist, erscheint folgende Meldung:

FILE	SIZE	PROT	FREE
18	NOT FOUND		670

Das DIR Dienstprogramm kann auf mehrere Arten genutzt werden:

- um alle Dateien zu sehen:
Benutzen Sie die Pfeiltasten um das Verzeichnis zu durchsuchen.
- um den Zustand einer speziellen Datei zu sehen:
Geben Sie die Dateinummer (gefolgt von ENTER) ein. Einer der oberen Bildschirme zeigt den Dateizustand an.
- um den Schutzzustand einer Datei zu ändern
Die Taste F3 schaltet den Schutz-Bit. (YES = geschützt; NO = Zugriff gewährt) Eine geschützte Datei kann nicht editiert oder gelöscht werden. Um einen Dateinachtrag mit

einem ausgewählten Zugriff zu beenden, öffnen Sie den Editor und schließen DIR.

4.7.2 DEL, eine Datei Löschen

Der Bildschirm fordert Sie auf, die Nummer der zu löschenden Datei einzugeben. Um ungewolltes Löschen zu vermeiden, fordert Sie die Nachricht "CLR to DELETE" auf, die Taste zu drücken. CLR löscht die Datei, ESC kehrt zum Grundmenu 'FILE UTILITIES', ohne zu löschen, zurück. Bei dem Versuch, eine geschützte Datei zu löschen, wird der Verzeichnisbildschirm mit der Anzeige der Dateizustände eingefügt.

4.7.3 COPY, Datei kopieren

Der Bildschirm fordert Sie auf, die QUELL-DATEI ? nummer und dann die ZIEL-DATEI ? nummer einzugeben.

Verschiedene Arten sind möglich:

- die Quelldatei ist nicht geöffnet; kein Effekt, zurück zu " Datei Dienstprogramm"
- die Zieldatei ist nicht geöffnet; eine neue Datei wird angelegt
- die Zieldatei ist schon offen; die Quelldatei und die Zieldatei sind verkettet
- der vorhandene Speicherplatz für die zu kopierende Datei ist zu klein: kein Effekt, der Bildschirm warnt den Bediener:

SPEICHER VOLL
eine Taste drücken

4.7.4 LOAD, lädt die gesamte BEE-Karte in den RAM

LOAD löscht den RAM-Inhalt. Deshalb erscheint eine Warnmeldung:

KARTE LADEN
1.03 ? JA NEIN

Der Identifikationscode (hier 1.03) wird von der Karte ausgelesen. F4 kehrt zum Basismenu zurück, F3 startet das Laden.

LADEN GULTIG
eine Taste drücken

oder

K. NICHT FORMATIERT
eine Taste drücken

Der letzte Bildschirm erkennt eine fehlende oder falsch formatierte Karte.

4.8 Menu "DEBUGGING"

4.8.1 Dienstfunktion "TRACE"

Dieses Dienstprogramm ist nur während der Ausführung eines Programms sinnvoll. Es zeigt die gerade auszuführenden Befehle an. Ebenso hat UNIPROG+ ein Mehrablaufbetrieb. Der zu überwachende Ablauf wird mit F1 gewählt, welche die Aufgabennummern rotieren läßt.

Der Bildschirm zeigt den Befehl im Editorformat an. Die untere Zeile zeigt die Aufgabennummer, die Zeile und das Programm, das gerade verfolgt wird, z.B.:

S:1 L:45 P:12

Das bedeutet: überwachter Simultan-Ablauf 1, Programm 12 bei Zeile 45.

4.8.2 Eingang/Ausgang-Funktion 'I/O CONTROL'

Dieses Dienstprogramm ist bestimmt, um Hardware-Funktionen zu testen und instandzusetzen. Der Zustand aller Steuerungsein- und -gänge kann sichtbar gemacht werden. Der Zustand der Ausgänge und des D-A-Wandlers (DAC) kann eingestellt werden.

Die Bedeutung und die Nummerierung der Ein- und Ausgänge können der Tabelle 5-1 : Ein- und Ausgänge entnommen werden.

Die Anzeige und die Leuchtdioden der Funktionstasten geben gleichzeitig den Zustand eines Enganges, eines Ausganges und den Wert der Analogspannung an.

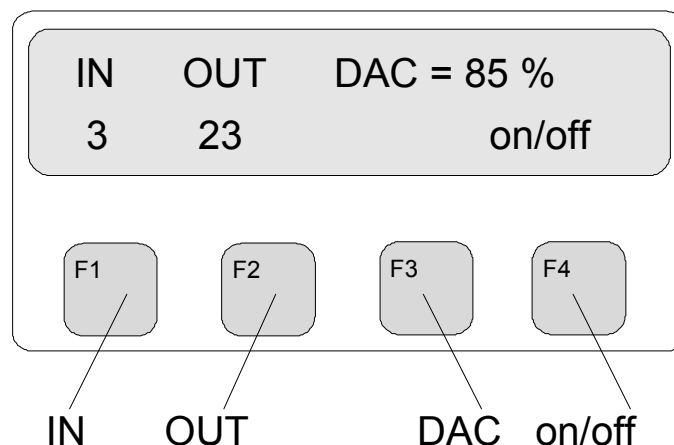


Abb. 4-1 : Menu "I/O Control"

Im dargestellten Beispiel hat die Anzeige folgende Bedeutung:

- die Leuchtdiode über F1 zeigt den Zustand des Einganges 3 an ;
- die Leuchtdiode über F2 zeigt den Zustand des Ausganges 23 an ;
- der DAC hat 85 % vom Endwert (8,5 V / 10V)
(unter UNIPROG+ ist nur das Unipolar-DAC-Betrieb, 0.. 10 V, Brücke auf der Platine in Stellung "a", möglich).
- Die Tasten F1, F2 un F3 setzen den Cursor auf 'IN', 'OUT' oder 'DAC'. Durch die numerische Tastatur wird eine Eingangs-oder Ausgangsnummer sowie der DAC-Wert eingegeben.
- F4 schaltet den gewählten Ausgang "EIN" und "AUS"

- Diese Dienstfunktion kann während der Ausführung eines Programmes benützt werden, was die Diagnose von Fehlern erleichtert (zum Beispiel das Wegbleiben einer Quittierung).
- Nach Tabelle 5-1 : Ein- und Ausgänge sind die 'Pseudo-I/O' 8, 9 und 10 die Zustände der Parallelabläufe (Absch. 5.6.5). F4 hat keine Wirkung auf diese drei 'I/O's. Die Pseudo-I/O 11...15 sind Merker.

4.8.3 Ausdruck-Funktion, "PRT"

Ein Drucker oder ein PC-kompatibler Computer kann durch einen RS 232 -Verbindungsstecker angeschlossen werden, um ein Programm auszudrucken. Die Nummer des zu druckende Programms wird eingegeben und der Druckvorgang mit 'ENTER' ausgelöst.

Das Format eines Linienausdruckes ist folgendes:

- Nummer der Linie
- Nummer der Datei
- Kennzahl des Befehles
- Befehlsabkürzung und Argumente

Der serielle Anschluß des Druckers oder PC's muß folgendermaßen eingestellt sein:

9600 Bauds, 8 bits, 1 stop bit, keine Parität

Bei einem PC unter DOS benutzt man das COPY-Befehl, z.B.:

COPY COM1: <Dateiname>

4.8.4 Dienstfunktion "SERVO"

Diese Funktion ist zur Einstellung und Diagnose lage geregelter Achsen sehr nützlich. Siehe auch die Anleitung der E600-Servo-Module.

Der Kanal wird durch Taste F1 "CHA" angewählt und entsprechend angezeigt.

- **Bei geschlossenem Regelkreis (Lageregelung)**, Leuchtd. F2 hell, wird an der oberen Zeile den Schleppfehler "P-ERR =" in der Anwandereinheit echtzeitig angezeigt. Durch die Taste "POSL" (POSition Loop) wird der Lageregelkreis geöffnet, um direkte Drehzahl-Regelung zu betreiben.

Die Taste F3 "ENB" (ENaBle) schaltet die Leistungsstufe ein und aus. Helle Leuchtd. F3 heisst eingeschalteter Zustand.

Taste F4 "HIGH" ist eine vorbehalten Funktion für Sondersystemversionen und soll in Standard-UNIPROG+ nicht angesprochen werden. Die zugehörige Leuchtd. soll also dunkel bleiben.

- **Bei offenem Regelkreis (Drehzahlregelung)**, Leuchtd. F2 dunkel, wird an der oberen Zeile die Geschwindigkeitsvorgabe ± 10 V, "A-OUT [V] =" angezeigt.

Der Taste F4 wird die Funktion zugewiesen, die Halbperiode eines Zyklus einzugeben, wobei die Geschwindigkeitsvorgabe symmetrisch abwechselnd, positiv und negativ verläuft.

Der Zyklus wird mit der numerischen Eingabe der Vorgabe ausgelöst. Bei Zeitwert 0 ist die Vorgabe stetig.

Durch F2 wird zurück zum Lageregelungsbetrieb umgeschaltet.

F1 und F3 erhalten die gleichen Funktionen

CLR setzt die Geschwindigkeitsvorgabe auf Null.

Die Taste STOP stellt die Parameter des Menu SERV wieder her.

Die Funktionen "POSL" und "ENB" sind während der Programm-Ausführung gehemmt.

5 Programmierbefehle

Die Befehle sind nachstehend anhand mnemotechnischen Text-Symbole oder Abkürzungen, beschrieben, die in den Editor- und TRACE-Dienstfunktion erscheinen. Die Kennzahlen, die benötigt werden, um Befehle mit der Tastatur einzugeben, werden in der Beschreibung angegeben. In der unteren formellen Darstellung sind die Symbole in Großbuchstaben und die Argumente in Kleinbuchstaben geschrieben.

Die Befehle und Pseudo-Befehle für die Erzeugung von Vektoren und Umrissen werden in Kapitel 8 beschrieben.

Ein Befehl belegt eine Zeile im Benutzerspeicherbereich. Wir nennen "**Adresse**" die Zahl, die sich aus der Verkettung der dezimalen Zeilennummer und der dezimalen Dateinummer ergibt. Die Dateinummer wird mit zwei dezimalen Ziffern geschrieben.

Beispiele: 1245 ist die Adresse der Zeile 12 im Programm 45,
 102 ist die Adresse der Zeile 1 im Programm 2,
 6 ist die Adresse der Zeile 0 im Programm 6

Die Schreibweise einer Adresse im Editor ist 'LINE/PROG'.

Numerische Werte (Hubweg, Zeitverzögerungen etc.) können von Befehlen verschieden erfasst werden.

- **unmittelbar** numerisches Argument: der numerische Wert befindet sich in der gleichen Zeile wie der Befehl.
- **direktes** numerisches Argument: der Befehl gibt eine Zeile an, in dem sich der numerische Wert befindet.
- **indirektes** numerisches Argument: der Befehl gibt die Adresse eines *Zeigers* bekannt, in der die Adresse mit dem numerischen Wert angegeben ist.

Beispiel: siehe untere Befehle POSA, POSAD, POSAI

Mit UNIPROG+ können drei sog. *Simultan-Abläufe* gleichzeitig ausgeführt werden. Jeder Ablauf besitzt seinen eigenen **Akkumulator**, d.h. ein Register, mit dem numerische Werte handgehabt werden können.

5.1 Positionsbefehle

Es gibt 6 Positionsbefehle: 3 sind für 'absolute' Bewegung oder Positionierungen, d.h. sie benötigen einen gegebenen Wert im bestehenden Koordinatensystem (siehe Kap. 3); die 3 anderen sind für 'relative' Bewegungen oder Verschiebungen.

Es sei hier erwähnt, dass der Endpunkt einer 'relativen' Bewegung so berechnet wird, dass eine Folge von solchen Verschiebungen zu keiner Kumulierung der Motorschritt-Rundungsfehler.

5.1.1 Absolut-Positionierung :

10	POSA	<Achse>	<Geschw.>	<Koord.>	<mode-e>
11	POSAD	<Achse>	<Geschw.>	<Koord.-Adresse>	<mode-e>
12	POSAI	<Achse>	<Geschw.>	<Zeiger-Adresse>	<mode-e>

5.1.2 Relativ-Positionierung :

10	POSR	<Achse>	<Geschw.>	<Versch.>	<mode-e>
11	POSRD	<Achse>	<Geschw.>	<Versch.-Adresse>	<mode-e>
12	POSRI	<Achse>	<Geschw.>	<Zeiger-Adresse>	<mode-e>

Argument "Geschw. ": Werte 0..7. Von 0 bis 6 wird die Geschwindigkeit aus derTabelle von Absch. 4.6.2 gelesen. Bei Wert 7 wird die Geschw. durch den Akkumulator bestimmt.

Argument "mode-e" : =0, 1, 2, 3. Ausführungsart

- 0: Bewegung aufnehmen aber nicht ausführen
- 1: aufgenommene Bewegungen der jeweiliger Achse ausführen
- 2: aufgenommene Bewegungen aller Achsen ausführen
- 3: einen Vektor im jeweiligen Raum erstellen, siehe Kap. 8

Es is also möglich, Verschiebungen zu addieren, dann auf einmal auszuführen. Ist eine Absolut-Positionierung mit Ausführungsart 1 oder 2 behaftet, sind zuvor aufgenommene Bewegungen wirkungslos.

5.1.3 Werkzeugauswahl

19	TOOL	<Werkzeugnummer>
----	------	------------------

Der Befehl TOOL setzt ein neues Bezugssystem fest, das für alle folgenden Befehle für die Positionierung gültig ist. Die Komponenten des Verschiebungs-Vektors zum Ursprungspunkt werden in der Datei '0' gespeichert.

- Anhand der Tabelle 'TOOL' können diese Verschiebungen überprüft und verändert werden.
- Die Werkzeug-Verschiebungen können auch aus dem Jogging angelernt und eingegeben werden.
- Das Argument 'Werkzeugnummer' wählt die Gruppe der 4 Komponenten aus, die sich auf eine Werkzeugnummer beziehen.

5.2 Weitere Kinematische Befehle

5.2.1 Referenz-Aufnahme:

17	REF	<Achse>
----	-----	---------

Die Referenznahme erfolgt wie in Abchnitt 3.1. Die Geschwindigkeit der Bewegung ist in der Konfiguration REF gegeben (siehe Agschnitt 4.4.2).

5.2.2 Referenzkontrolle :

18	CLOS	<Achse>	<Geschwind.>
----	------	---------	--------------

Die Funktion der Referenzprüfung ist im Abschnitt 3.5 beschrieben. Das Argument 'Geschwind.' ist dasjenige des Positionierbefehls (Absch. 5.1.2).

5.2.3 Lernbefehl:

13 TEACH <Achse> <Geschwind.> <Adresse>

Die Argumente 'Achse' und 'Geschwindigkeit' sind dieselben wie bei der Positionierung (siehe Abschn. 5.1.2).

Dieser Befehl erlaubt es, im Verlaufe eines Programmes, Positionen durch 'Lernen' zu korrigieren oder einzugeben. Der Befehl 'TEACH' wird nur im Schritt-Modus (MOD2) ausgeführt (siehe Kapitel 7). Er muss so auch mit einem Pausen-Merker versehen sein (siehe Abschn. 5.10).

Die Anzeige von TEACH lautet:

TEACH-IN (+ - START)	
Y	123.345

Die zweite Linie bezeichnet die Achse und den Inhalt der 'Adresse'. Es gibt zwei Möglichkeiten, diesen Inhalt zu korrigieren: die effektive Bewegung kann mit den Tasten 'JOG+' und 'JOG-' eingegeben werden, oder man fügt die numerische Korrektur direkt ein ohne die Achse zu bewegen). Die beiden Methoden können im gleichen 'TEACH'-Befehl vorkommen. Korrekturen werden immer an der durch den Befehl adressierten Zeile aufaddiert. Diese Linie kann ein unmittelbares, direktes oder indirektes Argument eines Positionierbefehls sein. Es ist auch möglich, die Korrektur in eine Linie zu setzen, die dann durch einen komplizierteren Algorithmus aufgerufen wird.

Mit den Tasten 'JOG +' und 'JOG-' kann eine Position durch mehrere Bewegungen ziemlich genau angenähert und korrigiert werden. Mit Hilfe des Potentiometers und der Wahl einer kleinen Geschwindigkeit kann eine präzise Korrektur vorgenommen werden. Durch Betätigen der 'START'-Taste wird das Programm anschliessend weitergeführt.

5.2.4 Parameter-Festlegung

83 SET <Parameternummer> <Parameterwert>

- | | |
|--------------|---|
| 0: SET PASSE | Entspäntiefe, immer positiv, siehe Befehl PECK |
| 1: SET GAP | Schutzabstand, siehe Abbildung Befehl PECK |
| 2: SET DELAY | Verweilzeit im Lochgrund |
| 3: SET BRK-D | Verzögerung beim Abschalten der Ausgänge 0 bis 7 (LAST DELAYED OUTPUT). |

5.2.5 Bohrzyklus

84 PECK <Achse> <Vorschub> <Position Bohrungsende> <Mode-d>

Der Bohrzyklus bestimmt die Anzahl der auszuführenden Entspanzyklen. Die Entspäntiefe ist immer eine positive Zahl. Die am Ende des Bohrzyklus erreichte Position bezieht sich auf den

gewählten Werkzeug.

4 Modi sind vorhanden :

Mode-d = 0 : Entspänen, Rückkehr zur Startposition am Zyklusende

Mode-d = 1 : Spanbrechen, Rückkehr am Ende des Zyklus

Mode-d = 2 : Entspänen, der Bohrer bleibt in Loch am Ende des Zyklus

Mode-d = 3 : Spanbrechen, der Bohrer bleibt in Loch am Ende des Zyklus

- Nach jedem Entspänen geht der Bohrer im Eilgang bis zum Schutzabstand, dann wird der Bohrvorschub wieder eingeschaltet, siehe Abbildung. Der Schutzabstand wird durch den Befehl SET 1 GAP programmiert. Vorgabewert ist 0.1 mm.
- Eine durch den Befehl SET 2 DELAY eingestellte Verzögerung kommt bei Modi 0 und 1 am Ende des Bohrvorgangs vor. Die Verzögerung hat den Vorgabewert 0.1 s.
- Die Modi 2 und 3 ermöglichen, mehrere Bohrzyklen zu verketteten, um einen progressiven Entspänvorgang zu erhalten.

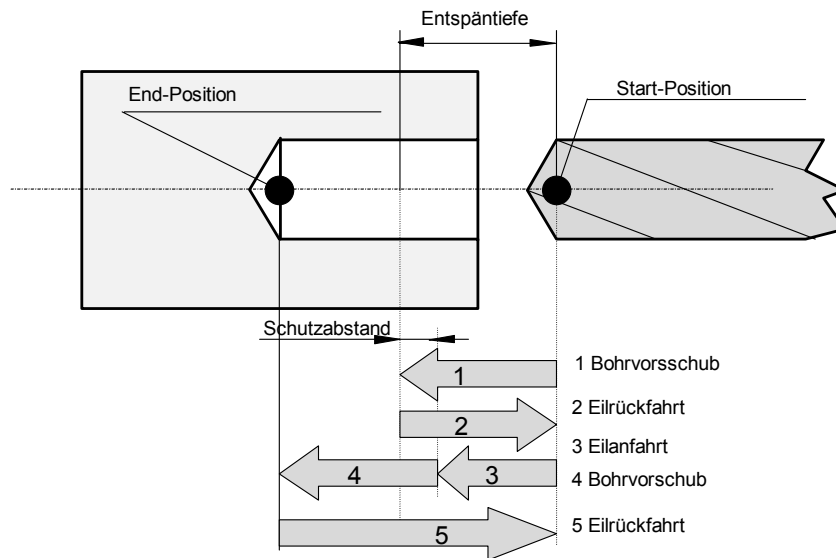


Abb. 5-1 : Bohrzyklus

5.2.6 Gewindegewinde-Befehl

81 TPING <Achse> <<Steigung>> <Endposition des Gewindegewindens>

Dieser Befehl erlaubt das Gewindegewinde mit einem Schneideisen, indem der Vorschub aus der Spindeldrehzahl und der Steigung bestimmt wird.

- Der Befehl SPVEL bestimmt die Rotationsgeschwindigkeit der Spindel, die von einem Frequenzumwandler angetrieben wird.
- Die Steigung des Gewindebohrers wird in die FEED-Tabelle eingegeben und der Parameter <<Steigung>> gibt die FEED-Nummer an.
- Um die Rotation an der Endposition des Gewindebohrers umzukehren wird der Ausgang 7 komplementiert. Er kann also am 'Richtung'-Eingang des Frequenzumwandlers angeschlossen sein.
- Die Rotation der Spindel kehrt um, wenn die Endposition erreicht ist. Die Spindel-Drehzahl wird nicht gemessen, die tatsächliche Tiefe des Gewindes wird durch die Eigenschaften des Frequenzwandlers und durch das Drehmoment beeinflusst.

Zum Beispiel :

Gewindesteigung 1 mm ; Drehzahl 600 U/Min ; Tiefe 50

Menu FEED:

RATE # 3 = 1.00 = Steigung

00 30 10 POSA X 4 0.00 1

01 30 57 SPVEL 600; setzt dieDrehzahl fest

02 30 29 ON 7 ; setzt die Drehrichtung fest

03 30 29 ON 6 ; Startet den Umwandler

04 30 72 TPING X 3 50; Startet das Gewindebohren

Bemerkung :

Da die Spindel-Drehzahl nur annähernd bestimmt ist, ist ein Werkzeughalter mit Längen-Ausgleich unbedingt erforderlich.

5.2.7 Polarkoordinaten

86 ANGLE <Geschwindigkeit> <Wert> <Modus>

87 RADIUS <Geschwindigkeit> <Wert> <Modus>

Die Befehle ANGLE und RADIUS dienen zum Programmieren von Positionierbewegungen in Polarkoordinaten. Das Bezugssystem ist wie folgt definiert:

- Der Modul (RADIUS) wird ab Werkzeugursprung gemessen,
- Der Winkel (ANGLE) wird im Gegenuhrzeigersinn ab 3 Uhr gemessen. Siehe auch Befehl ORGA.

<Modus> bestimmt die Ausführung, ähnlich wie bei den Befehlen POSA/POSR:

Modus = 0: Die Werte werden nur gespeichert, keine Bewegung wird gestartet,

Modus = 1,2: Die Achsen verfahren mit ihren jeweiligen Geschwindigkeiten,

Modus = 3: Vektorgeneration, die Achsen sind in einer Linearinterpolation verknüpft.

5.2.8 Winkelverschiebung

88 ORGA <Winkelverschiebung>

In Verbindung mit den Befehlen ANGLE und RADIUS erlaubt ORGA, das Bezugssystem zu drehen. Die Winkelverschiebung darf positiv oder negativ sein.

5.3 Eingangs- / Ausgangs-Befehle

Mit diesen Befehlen können die Eingänge den Programm-Ablauf durch Springen oder Anhalten, beeinflusst und die Ausgänge festgesetzt werden.

5.3.1 Warten auf einen Eingang:

20	WAIT0	<Eingang >
21	WAIT1	<Eingang>

Das Programm wartet bis der bezeichnete Eingang 0 bzw. 1 wird.

5.3.2 Bedingter Sprung :

22	BRINO	<Eingang>	<Adresse>
23	BRIN1	<Eingang>	<Adresse>

Wenn der Eingang 0 bzw. 1 ist, springt das Programm auf die Linie 'Adresse'. Anderenfalls wird der Folgebefehl ausgeführt. (Was die Sprungadresse anbelangt, siehe Abschnitt 5.6.1).

5.3.3 Ausgangs-Festsetzung:

28	OFF	<Ausgang>
29	ON	<Ausgang>

Der bezeichnete Ausgang wird ein- oder ausgeschaltet. Die als Ein- und Ausgang anerkannten booleschen Variablen sind in der Tabelle 5-1 : Ein- und Ausgänge angegeben.

Nummer in Befehl	Physikal. Eingang	Nummer in Befehl	Physikal. Ausgang
0	IN(0)	0	OUT(0)
1	IN(1)	1	OUT(1)
2	IN(2)	2	OUT(2)
3	IN(3)	3	OUT(3)
4	IN(4)	4	OUT(4)
5	IN(5)	5	OUT(5)
6	IN(6)	6	OUT(6)
7	IN(7)	7	OUT(7)
8	SIM0	8	SIM0
9	SIM1	9	SIM1
10	SIM2	10	SIM2
11..15	FLAG(1..5)	11..15	FLAG(1..5)
16..63	IN(16..63)	16..63	OUT(16..63)

Tabelle 5-1 : Ein- und Ausgänge

- Die Eingänge IN(0..7) sind die in Abschnitt 4.4.2 erwähnten Endschalter oder REF INPUT und können durch die Befehle 20..23 getestet werden, auch immer wenn sie als Referenzschalter eingestellt sind.
- Die Pseudo-I/O SIM0, SIM1, SIM2 sind die Aktivierungszustände der drei Simultan-Abläufe.
- Die Elemente FLAG(1..5) sind keine eigentliche Eingänge, sondern eher interne Merker, die durch den Programmierer nach Wunsch benützt werden können. Diese Merker werden durch die OFF- und ON-Befehle 0 bzw. 1 gesetzt.

- Die Eingänge IN(16..63) und die Ausgänge OUT(16..63) beziehen sich auf die am Erweiterungsbus anschliessbare E/A-Module. Jedes Modul ist mit einer Adresse versehen, die anhand 4 Schalter eingestellt wird.

Schalter 4 3 2 1	Adresse IN OUT
O O C O	16...23
O O C C	24...31
O C O O	32...39
O C O C	40...47
O C C O	48...55
O C C C	56...63
C O O O	64...71
C C C C	0...7

O = Open
C = Closed

Tabelle 5-2 : Adresse der E/A-Module

OUT(0) bis OUT(7) sind im Gehäuse der Steuerung 'E-600 Base' eingebaut und am Rückplatten-Stecker verfügbar, siehe Kap. 9.

Die Version E-600 ND Base weist keine eingebauten Ausgänge auf, siehe Kap. 10.

5.3.4 Ausgangs-Komplementierung

95 CPL <Ausgang>

Der bezeichnete Ausgang wird logisch komplementiert (ein- oder ausgeschaltet).

Beispiel:

Komplementierung von Ausgang 18 durch Taste 0 (Tastengestartete Programm 90)

```
90 00    95  CPL    18
90 01    90  NOP
```

5.4 Spindelmotor-Ansteuerung

5.4.1 Spindeldrehzahl

57 SPVEL <Drehzahl>

Das Argument bezeichnet die (unmittelbar adressierte) Drehzahl in Umdrehungen pro Minute. Der Parameter « MAX RPM 10 Volt » muss eingestellt sein (Mit dem Wert 100 ist die Drehzahl in Prozent dargestellt).

5.4.2 Motorumschaltung und Spindeldrehzahl

85 MOTOR <Motor-Nummer> <Drehzahl>

Wenn ein einziger Frequenzumformer zwei oder mehrere Spindelmotoren abwechselnd über Schützen antreibt, kommt der Befehl MOTOR zur Anwendung, der das stromlose Umschalten der Spindelmotoren auführt.

Der Frequenzumwandler muss nämlich unbedingt ohne Strom, also bei Drehzahl Null umgeschaltet werden.

Der Befehlszyklus MOTOR setzt die Geschwindigkeit auf Null, wartet die Verzögerung des Befehls SET 4 BRK-D (BReaK-Delay) ab, und daraufhin setzt die unmittelbar adressierte Drehzahl U/min fest.

Mit 'MOTOR 0' alle Motoren sind auf Null gesetzt.

Mit 'MOTOR 1' der Motor 2 ist auf Null gesetzt und der Motor 1 startet, OUT 1.

Mit 'MOTOR 2' der Motor 1 ist auf Null gesetzt und der Motor 2 startet, OUT 2.

Usw... bis MOTOR 7.

5.5 Manipulationen numerischer Werte

Es werden reelle Zahlen (Floating Numbers) und ganze Zahlen (Integers) unterscheidet. Reelle Zahlen sind : Geschwindigkeit, Masse (inkl. Koordinaten-Verschiebungen) und Zeit. Ganze Zahlen sind : Adressen, Anzahl Zyklen.

Die Befehle für die Manipulation von Zahlen oder der Arithmetik verlangen a priori reelle Zahlen in der adressierten Linie. Ausnahme bildet dabei der Befehl 'IDATA', dessen Linieninhalt eine ganze Zahl ist. Das Programm kann also den numerischen Wert einer Linie (auch ein unmittelbares Argument) verändern. Der Inhalt muss eine reelle Zahl sein.

5.5.1 Laden des Akkumulators

Unmittelbares Laden des Akkumulators :

50 FLOAD <reelle Zahl>
51 ILOAD <ganze Zahl>

Direktes Laden des Akkumulators :

52 LOADD <Adresse>

Indirektes Laden des Akkumulators :

53 LOADI <Zeiger-Adresse>

5.5.2 Abspeichern des Akkumulators

Direktes Abspeichern des Akkumulators

55 STORD <Adresse>

Indirektes Abspeichern des Akkumulators

56 STORI <Zeiger-Adresse>

5.5.3 Inkremente /Dekremente

Diese zwei Befehle werden in Zusammenhang mit der indirekten Adressierung verwendet. Sie ermöglichen, von der laufenden Linie einer Datentabelle zur nächsten bzw. vorhergehenden zu springen, indem der adressierten Zeile der Wert 100 zu- bzw. weggezählt wird, siehe Anfang dieses Kapitels. Ein Sprung ausserhalb der Datei ist nicht möglich.

58 INCD <Adresse>

59 DECD <Adresse>

5.5.4 Speichern einer Variablen auf die Datenkarte

54 SAVE <Adresse>

Die in der Position 'Adresse' stehende Variable wird auf die EEPROM-Karte gespeichert (auf dieselbe Adresse).

Dies ist eine ziemlich langsame Operation (im schlechtesten Falle einige Sekunden). Sie sollte deshalb nicht in einem schnell auszuführenden Programmabschnitt stehen.

5.6 Programmablauf-Befehle

5.6.1 Unbedingter Sprung :

60 JMP <Adresse>

Das Programm fährt bei der Linier 'Adresse' fort.

Bemerkung: Der Programm-Editor kann Linien in einem Programm löschen oder einfügen. Die Nummerierung in einer Datei wird dabei verändert. Es ist deshalb empfohlen (jedoch nicht vorgeschrieben), Sprünge (JMP, BRIN) und Unterprogramm-Aufrufe (CALL) an die Null-Linie einer Datei zu adressieren.

5.6.2 Unterprogramm-Aufruf:

61 CALL <Adresse>

Das Programm fährt bei der Linier 'Adresse' fort. Am Ende des Unterprogramms wird zurück zur Aufruflinie unmittelbar nachfolgenden Linie gesprungen.

- Die Aufrufe dürfen bis 10 mal verschachtelt werden

5.6.3 Programm- und Unterprogramm-Ende :

62 END

Dieser Befehl beendet ein Programm oder Unterprogramm. Wenn der letzter Befehl eines Programmes oder Unterprogramm auf der letzten Linie in einer Datei steht, so braucht es keinen 'END'-Befehl.

5.6.4 Wiederholungen:

Die folgenden Befehle dienen zum Aufbau von Programm-Schleifen. Die Schleifen dürfen bis 10 mal verschachtelt werden. Eine Schleife darf sich über die Datei hinaus erstrecken. Sie wird mit ENDRP abgeschlossen.

63	REP	<Anzahl >	unmittl. Argument
64	REPD	<Adresse>	direktes Argument,
65	ENDRP		Grenze der Schleife

Im REP-Befehl muss das unmittelbare Argument eine ganze Zahl sein ; die Zahl in der adressierten Linie für den REPD-Befehl kann eine ganze oder reelle Zahl sein. Ist die Zahl reell, wird nur der ganzzahlige Teil berücksichtigt.

5.6.5 Aufruf von Simultan-Abläufen:

67	SIM1	<Adresse>	
68	SIM2	<Adresse>	

Das Simultan- (oder Parallel-) Ablauf wird bei 'Adresse' gestartet. Durch ein END wird es automatisch gestoppt. Ein Simultan-Ablauf kann angehalten werden, indem man den entsprechenden Merker SIM0..SIM2 (siehe Tabelle 5-1 : Ein- und Ausgänge) durch einen OFF-Befehl rücksetzt. Durch einen ON-Befehl wird das Ablauf wieder weitergeführt. Das Testen der SIM-Merker erlaubt es dem Programm, die Zustände der anderen Simultan-Abläufe zu kennen.

Der Aufruf eines bereits laufenden Simultan-Ablaufs bricht die laufend ausgeführten Befehlsreihenfolge ab, um den Aufruf normal durchführen zu können (Ausführungs-Start ab der vorgeschriebenen Adresse). Die STOP-Taste bricht alle Simultan-Abläufe ab.

5.6.6 Bedingte Sprünge (Akkumulator-Test):

Durch diese Befehle wird der Inhalt der Akkumulatoren abgefragt, meistens nach arithmetischen Operationen.

Es sei darauf hingewiesen, dass die Kennziffer dieser Befehle unmittelbar nach BRINO und BRIN1 liegt.

Falls die Bedingung für einen Sprung nicht erfüllt ist, wird der Folge-Befehl normal ausgeführt.

Was das Argument "Adresse" anbelangt, siehe die Bemerkung des Abschnittes 5.6.1.

24	BRM	<Adresse>	Sprung, wenn der Inhalt des Akk. negativ ist
25	BRP	<Adresse>	Sprung, wenn der Inhalt des Akk. positiv oder null ist
26	BRZ	<Adresse>	Sprung, wenn der Inhalt des Akkum. Null ist
27	BRNZ	<Adresse>	Sprung, wenn der Inhalt des Akkum. nicht null ist

5.7 Verzögerung

70	WAIT	<Dauer>	Unmitt. Argument
71	WAITD	<Adresse>	Directes Argument direct, Dauer in Adresse

Mit diesen Befehlen können Verzögerungen in Programmen eingebaut werden. Die Zeit wird in Sekunden ausgedrückt. Der Inhalt der adressierten Linie für den WAITD-Befehl muss eine reelle Zahl sein.

5.8 Arithmetische Befehle

Diese Befehle führen arithmetische Operationen aus dem Akkumulator und dem Inhalt ihres direkten Argumentes aus. Das Resultat ist in den Akkumulator zurück geladen. Die Zahlenart wird aus dem Argumentes abgeleitet : ist die adressierte Linie eine 'IDATA' so werden die Operanden als Ganzzahlen behandelt ; in den übrigen Fällen, als Reellzahlen.

91	ADDD	<Adresse>	Accu =Accu + [Adresse]
92	SUBD	<Adresse>	Accu =Accu - [Adresse]
93	MULD	<Adresse>	Accu =Accu x [Adresse]
94	DIVD	<Adresse>	Accu =Accu / [Adresse]

5.9 NOP und "Pseudo-Befehle"

90	NOP
-----------	------------

Der NOP-Befehl (No Operation) gewährleistet offene Linien, die bei der Nachbearbeitungen des Programms nötig sind. Beim Editieren erscheint 'NOP' immer bei einer noch zu öffnenden Linie.

98	FDATA	<reelle Zahl>
99	IDATA	<ganze Zahl >

FDATA un IDATA sind keine eigentlichen Befehle, sondern Vorwörter zur Eingabe von Zahlen in ein Programm. Diese Vorwörter dienen zur Bestimmung der Zahlenart.

Diese Kennzeichen (98 und 99) haben die gleiche Wirkung wie ein NOP, wenn innerhalb eines Programmes. Es können also beliebig Zahlen in ein Programm eingesetzt werden.

5.10 Pausen-Kennzeichen

Jeder Befehl kann mit einem Pausen-Kennzeichen versehen werden. Im Modus MOD1 ist er ohne Bedeutung, während im Modus MOD2 das Programm nach jedem gekennzeichneten Befehl anhält, siehe Kap. 7.

6 Der Programm-Editor

Der Einstieg in den Editor erfolgt durch das Menu PROGRAMMING, siehe Abschnitt 4.2. Der Benutzer muss nun die Nummer der zur Bearbeitung bestimmten Datei eingeben. Es erscheint die Linie 0 des Programmes :

```
POSA X 3 12.234 _  
10          0 p11
```

Die erste Linie zeigt das Symbol des Befehls und den Wert der eventuellen Argumente. Die untere Linie enthält die Befehl-Kennzahl (unter dem Symbol) und die Linien- und Dateinummern.

Wenn die Datei noch nicht geöffnet wurde, erscheint der NOP-Befehl. Eine Datei ist erst offen, wenn mindestens ein Befehl vorhanden ist.

6.1 Lesen eines Programmes

Pfeiltasten:	Anzeige der vorhergehenden ↑ oder der folgenden ↓ Zeile
Taste PROG/LINE:	direkte Anzeige der in der Tastatur eingegebenen Zeile
Taste ESC :	Zurück zum Editor, Wahl einer neuen Datei ; ein erneutes Drücken führt zum Grund-Menu zurück.

6.2 Aendern eines Linieninhaltes

Das Einfügen einer neuen Linie entspricht dem Aendern einer Linie, die einen NOP-Befehl beinhaltet. Es genügt deshalb, das Modifizieren einer Linie zu erläutern.

Taste ENTER:	Verschieben des Cursors gegen rechts zum neuen Argument. Die untere Linie bezeichnet die Art des neuen Argumentes. Nach Eingabe des letzten Argumentes (oder Befehl wenn kein Argument) wird die Linie gespeichert und die Folgelinie erscheint.
Taste CLR :	verschiebt den Cursor gegen links zum vorhergehenden Argument. Ist der Cursor auf dem Befehlsymbol, hat die Taste keine Wirkung.
Taste F5 :	Befindet sich der Cursor auf dem Befehlsymbol oder auf einem symbolischen Argument, können alle möglichen Werte oder Befehle der angewählten Position durch Drücken der F5-Taste auf dem Bildschirm aufgerufen werden. Steht der Cursor auf einem Zahl-Argument, hat die Taste keine Wirkung.

Um einen Befehl oder ein numerisches Argument einzugeben, ist es nicht unbedingt nötig, mit der F5-Taste alle Möglichkeiten durchzusuchen. Durch Eingabe der entsprechenden Kennzahl kann ein Befehl direkt eingegeben werden. Die Kennzahlen sind mit der Zeit leicht zu merken : Die Positionierungen, Klasse 10, die Daten-Manipulationen, Klasse 50, die Verzögerungen, Klasse 70 etc. Der Gebrauch der Taste F5 ist dann nur auf wenige Drucke beschränkt. Es sei erwähnt, dass bei der Eingabe einer Kennzahl, das Symbol nach der zweiten Ziffer automatisch erscheint (ohne ENTER). Um die Taste F5 so zu gebrauchen, muss mit dem Cursor zurückgefahren werden.

Beispiel: Einfügen des Befehls POSR in eine leere Linie :

- Der Befehl ist in der Klasse 10 ; 10 eingeben (ohne ENTER) ; das Symbol POSA wird

angezeigt ; der Cursor steht auf dem Argument 'Achse'.

- Mit CLR den Cursor zurückfahren, dann F5 drücken und die erscheinenden Symbole beobachten. POSR erscheint mit der Kennzahl 14. ENTER drücken.
- Die Anzeige fordert zur Wahl der Achse auf. Entweder mit F5 oder durch Eingabe von 0 für X, 1 für Y, 2 für Z, 3 für U.
- Mit ENTER das folgende Argumente, die vorgewählte Geschwindigkeit (SEL.SPEED) eingeben. Dies ist ein numerischer Wert. F5 ist dabei wirkunglos. Das folgende Argument ist die Verschiebung (DISP'MENT). Es wird in technischen Einheit eingegeben.
- Das letzte Argument ist der Ausführungsmodus (EXEC MODE) ; mit F5 oder direkte Eingabe :
 - 0 : keine Ausführung
 - 1 : Ausführung nur auf der gegebenen Achse
 - 2 : Ausführung auf allen Achsen
 - 3 : Generation eines Vektors
- Mit ENTER wird die Linie gespeichert und es folgt die nächste Linie.

Es kann auch nur ein einzelnes Argument in der Zeile verändert werden. Es muss durch den Cursor angewählt werden. Durch Drücken der ENTER-Taste (bis zum Verschwinden des Cursors) wird der neue Wert gespeichert.

6.3 Einfügen und löschen einer Linie

Mit der F3-Taste kann eine Linie in der angezeigten Position eingefügt werden.

Beispiel: Die Linie 12 wird angezeigt und enthält den Befehl WAIT. Nach Drücken der Taste F3 zeigt die Linie 12 en NOP und der Befehl WAIT ist in der Linie 13.

Mit der Tast F4 kann eine Linie gelöscht weden. Die nachfolgende Linie erhält ihre Nummer und wird angezeigt.

6.4 Setzen eines Pausen-Kennzeichen

Der Pausen-Kennzeichen kann zu jeder Zeit durch Drücken der Taste F1 an einem Befehl gelöscht oder gesetzt werden. Die entsprechende Leuchtdiode zeigt sein Zustand an. Auch wenn nur dieser Zustand geändert werden soll, muss man die Linie mit einer oder mehreren Betätigungen von ENTER verlassen.

7 Programm-Ausführung

Die Programm-Ausführung wird durch die Tasten START, STOP und PAUSE, durch die in der CTRL-Konfiguration ausgewählten Eingänge und durch den Ausführungsmodi MOD1 / MOD2 bestimmt.

7.1 Die Ausführungsmodi nach Tasten MOD1, MOD2

MOD1:

Normaler Ausführungsmodus. Die Pausezeichen im Befehl werden nicht berücksichtigt. Die START-Taste leuchtet.

MOD2:

Das Pausezeichen stoppt das Programm **vor** der Ausführung des gekennzeichneten Befehls. Die simultanen Abläufe gehen weiter, wenn kein gekennzeichneteter Befehl erreicht ist.

Während der Pause blinkt die START-Taste. Bei Drücken von START fährt das Programm fort bis zum nächsten gekennzeichneteten Befehl. Dieser Modus bewährt sich besonders mit dem TRACE-Dienstprogramm.

Dieser Ausführungsmodus bietet ausserdem den Vorteil, die Bewegungsgeschwindigkeit zu begrenzen.

In diesem Modus sind während der Bahnberechnung die Radian in der Reihenfolge der Ausführung der Umrisses angezeigt. Die Taste START blinkt und lädt zur Fortsetzung der Berechnungen ein.

7.2 Tastenfunktionen START, PAUSE, STOP

Diese Funktionen sind durch Eingänge doppelt geführt, die in der Konfiguration CTRL eingestellt sind.

START

Wenn die Glühleuchten der START und STOP-Tasten aus sind, so bewirkt ein Drücken von START den Beginn des Programms, das als "STARTPROGRAMM" durch das Menu VECT gekennzeichnet ist.

Wenn die STOP-Glühleuchte an ist, so wird das Programm gestartet, das als "POWER ON PROGRAM" gekennzeichnet ist.

PAUSE

Die MAN-Taste – oder ein Impuls an der entsprechenden Eingang - unterbricht das laufende Programm am Ende des momentanen Befehls. Eine Bewegung wird sofort auf 0-Geschwindigkeit mit der eingestellten Rampe gestoppt. Die tatsächliche Position bleibt erhalten. Die START-Drucktaste blinkt und die MAN-Taste leuchtet.

STOP

Das erste Drücken von STOP, während das Programm abläuft, bewirkt einen sofortigen Stop der Ausführung aller Abläufe. Die momentanen Bewegungen werden auf 0-Geschwindigkeit mit der eingestellten Rampe gestoppt. Die Ausgänge und der DAC werden zurückgesetzt. Die tatsächlichen Positionen der Achsen bleiben erhalten.

Bei einem zweiten Drücken leuchtet das STOP-Glühlicht. Somit wird gemeldet, dass beim nächsten Drücken von Start das Programm ausgeführt wird, das in VECT als POWER ON PROGRAMM eingestellt ist.

Nach Einschalten der Steuerung wird der Modus MOD1 festgesetzt und das POWER ON PROGRAMM automatisch ausgeführt. (Wenn ein Power On Programm nicht erwünscht ist, muss 100 im VECTOR-Datenfeld eingegeben werden).

Die Ausführung kann durch Drücken von START fortgesetzt werden.

Anmerkung:

Die meisten Dienstfunktionen sind verfügbar, während ein Programm läuft. Dennoch sollte der Editor mit Vorsicht benutzt werden : Eine Zeilen-Einfügung oder -Löschung bewegt den Teil des Benutzerspeichers über die momentane Zeile. Katastrophale Fehler können daraus resultieren.

7.3 Fehlerbearbeitung

Drei Fehlersituationen werden bearbeitet:

- Fehler, die innerhalb des Motorantriebs entstehen.
- Überbewegung, die durch Softwarebegrenzungen festgestellt werden (nur im Bahnmodus; im Positionierungsmodus ist die Bewegung von vornherein begrenzt)
- Ueberschreitung der Schleppfehlergrenze (Alarm zone)

Wenn eine Fehlersituation auftaucht werden alle Bewegungen sofort gestoppt. Die digitalen Ausgänge bestehen jedoch ungewechselt. Der Bildschirm zeigt eine der folgenden Meldungen:

AXIS a FAULT
STOP druecken

STROKE a TOO LARGE
STOP druecken

AXIS a ALARM ZONE
STOP druecken

" a " steht für die Achsen X, Y, Z und U.

Drücken von STOP setzt alle Ausgänge und DAC zurück. Bei der ersten Fehlersituation zeigt der Bildschirm:

AXIS a FAULT
JOGGING +/- → ESC

Die fehlerhafte Achse kann mit den JOG-Tasten langsam bewegt werden. Dies ist nützlich, wenn der Antrieb mit festverdrahteten Begrenzungsschaltern ausgestattet ist. Durch die ESC-Taste kehrt die Steuerung zum Normalzustand zurück. Wenn der Fehler bestehen bleibt, muß die Steuerung ausgeschaltet werden.

In jedem Fall führt die Steuerung nach einer Rückkehr aus einer Fehlersituation das "POWER ON PROGRAMM" durch.

WICHTIG :

Die STOP-Taste, sowie der als Extern-Stop eingestellte Eingang, dürfen nicht im Sinne der Sicherheitsvorschriften als Notstop betrachtet werden.

8 Erzeugung von Vektoren und Umrissen

8.1 Einführung : Möglichkeiten und Definition des Raumes

Die Steuerungen der Reihe E600 können Vektoren in jedem zweidimensionalen, durch die Achsen X, Y, Z und U erzeugten Raum erstellen. Die Bewegungen auf willkürlichen Kurven erreicht man, indem die Kurve in kurze, aufeinanderfolgende Geraden aufgeteilt wird. Die Basissprache PINX-E ermöglicht das Definieren der unterschiedlichsten Kurven im kartesischen, polaren und zylindrischen Koordinatensystem.

UNIPROG+ vereinfacht die Aufgabe des Programmierers durch seinen Befehlsatz zur linearen, kreis- und schraubenförmigen Interpolation in **kartesischen Koordinaten**.

Man muss die Erzeugung eines Vektors und die Erstellung eines Umrisses mit Hilfe von geradlinigen und/oder kreisförmigen Segmenten mit gleichbleibender Geschwindigkeit unterscheiden. Ein einzelner Vektor wird mit dem Ausführungsmodus 3 bei den Positionierbefehlen erstellt. Im Gegensatz dazu wird die Geometrie eines Umrisses mit Pseudobefehlen ausserhalb des Hauptprogrammes definiert.

Während der Erzeugung eines Vektors oder der Ausführung einer Bahn wird die Geschwindigkeit in der **Kurvenkoordinate** gesteuert. Sie stellt also eine virtuelle Achse dar (die Kurvenkoordinate misst den Weg entlang eines Vektors oder einer Kurve).

Eine Kurve mit Eckpunkten hat zwangsläufig Unterbrechungen der Achsengeschwindigkeit zur Folge ; diese Unterbrechungen können von einem Schrittmotor nur bei langsamen Bewegungen bewältigt werden.

Vektoren können in 6 verschiedenen kartesischen Räumen definiert werden :

0	1	2	3	4	5
XY	XZ	XU	YZ	YU	ZU

8.2 Erzeugung eines Vektors

30 DPATH <Ebene> <Werkzeug links-rechts>

Der Befehl DPATH (**D**efine **P**ATH) bestimmt die Ebene, in der der Umriss ausgeführt wird.

- Das Argument links-rechts (L-R) zeigt die Seite für den Start der Korrektur der Werkzeugbahn.
- Dieser Befehl muss sich unbedingt auf der ersten Zeile einer Umrissdatei befinden.

40 ORGP <Achse> <Absolute Position>

Der Befehl ORGP (**O**RiGin of **P**ath) bestimmt das Bezugssystem für den Umriss und muss auf den Befehl DPATH folgen.

Wenn das Bezugssystem für den Umriss sich auf dem Startpunkt befindet, ist der Befehl ORGP überflüssig.

8.3 Geometrische Definition einer Kontur

Ein Umriss ist eine geometrische Kette aus linearen Teilstücken, Segmenten von Kreisen und Schraubenlinien, die wie eine einzige Bewegung ausgeführt wird. Die Definition der Geometrie eines Umrisses befindet sich in einer Datei ausserhalb des eigentlichen Programmes. Eine Datei mit geometrischen Definitionen kann mehrere Umrisse enthalten. Das Interpolationssystem braucht eine Interpretation Die geometrische Definition muss vorberechnet – oder interpretiert – werden. Diese Interpretation erstellt Tabellen im Speicher, die vom Programm bei seinem Ablauf gebraucht werden. Der programmierer muss also die Interpretation des Umrisses verlangen. Diese Frage wird im Abschnitt 8.4 behandelt.

Ein Umriss ist in ihrem eigenen Bezugssystem definiert und steht in keinem Zusammenhang mit den Bezugssystemen aus Kapitel 3. Ein Umriss beginnt immer im Ursprung des Bezugssystems. Während der Ausführung beginnt sie an der aktuellen Achsenposition. Der Umriss kann zur Zeit des Programmablaufs an unterschiedlichen Orten 'gezeichnet' werden. Die geometrische Datei kann also als eine Art Unterprogramm betrachtet werden.

Die geometrische Definition eines Umrisses beginnt mit der Definition des Raumes. Der Befehl "Define PATH" -DPATH- (siehe Absch. 8.2) wird hier zu einem Pseudo-Befehl. Falls verschiedene Umrisse in einer einzigen Datei vorhanden sind, dient der Pseudo-Befehl DPATH zur Trennung derselben. (Der Befehl END darf nur am Ende des letzten Umrisses der Datei verwendet werden).

8.3.1 Definition eines linearen Teilstückes

32	LINA	<Achse>	<Koordinate>	<mode-e>
33	LINR	<Achse>	<Komponente>	<mode-e>

Die Pseudo-Befehle LINA und LINR haben ein Verhalten, das analog zu den Befehlen POSA und POSR ist. Die Begriffe der absoluten Koordinate und der relativen Bewegung beziehen sich auf den Raum im Umriss, (Abb. 8-1) und nicht auf das Bezugssystem des Kapitels 3. Ein Pseudo-Befehl ist für jede Komponente des Vektors erforderlich, sie können in einer beliebigen Reihenfolge angegeben werden. Der letzte Pseudo-Befehl muss den Modus 2 aufweisen.

Die Koordinaten oder die Komponenten sind natürlich in der Einheit angegeben, die durch SCALEK definiert ist. Der Modus für alle Komponenten ist 0 mit der Ausnahme der zuletzt eingeschriebenen, die den Modus 2 aufweisen muss.

Wenn eine der Koordinaten oder eine der Komponenten des Vektors nicht angegeben ist, wird der letzte Aufruf der Achse berücksichtigt. Diese Besonderheit gestattet, Eingabearbeit zu sparen und einfach tabellierte Funktionen zu programmieren, wie das nachfolgende Beispiel zeigt. Diese Vereinfachung funktioniert nicht mehr nach der Programmierung eines Kreisbogens oder eines Befehl POINT.

Die Befehle LINA und LINR müssen immer den Befehlen POINT folgen, siehe "POINT".

```

DPATH   XY   L
LINA    X    30
LINA    Y    22   ;seg. 1
LINA    X    52   ;seg. 2
LINA    X    60
LINA    Y    32   ;seg. 3
LINA    Y    52   ;seg. 4
.....
.....

```

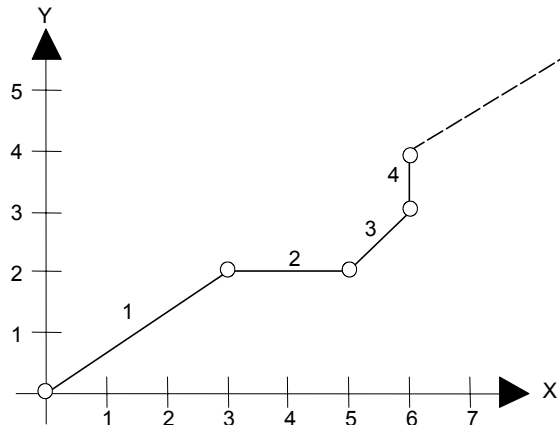


Abb. 8-1 : Umriss mit Durchmesser Null

Wenn eine Werkzeugkorrektur erforderlich ist, muss der Befehl POINT mit einem Bogen vom Radius Null (RAD = 0) anstelle der Befehle LINA oder LINR verwendet werden.

```

DPATH   XY   L
POINT   X    30
POINT   Y    22   ;seg. 1
POINT   X    52   ;seg. 2
POINT   X    60
POINT   Y    32   ;seg. 3
POINT   Y    52   ;seg. 4
.....
.....

```

42 POINT <Achse> <Absolute Position des Eckpunktes> <mode-e>

Der Befehl POINT bietet die Möglichkeit, die Koordinaten des Eckpunktes zu markieren, in dem der Radius der Ausrundung sich einschreiben wird, der von dem Befehl RAD gegeben wird.

Die Koordinaten der Eckpunkte werden im Bezugssystem des Umrisses angegeben, das heisst, in Bezug auf einen beliebigen Punkt der Umrisebene. Im Falle einer Figur, die eine zentrale Symmetrie aufweist, werden die Koordinaten natürlich im Verhältnis zum Zentrum angegeben.

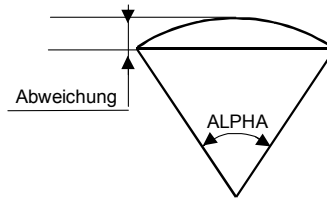
Der Befehl POINT zeigt das gleiche Verhalten wie der Befehl LINA/LINR: wenn eine Koordinate nicht spezifiziert ist, wird der letzte Aufruf der Achse berücksichtigt. Diese Vereinfachung funktioniert aber nur wenn die Folge von POINT-Befehlen nicht durch CIRA/CIRR oder LINA/LINR unterbrochen wird.

Um einen Umriss zu beenden, der aus den Befehlen POINT besteht, muss die Datei mit den Befehlen LINA oder LINR beendet werden. Die 2 Ebenenkoordinaten müssen aufgerufen werden, um den Umriss zu beenden.

8.3.2 Definition eines Kreisbogens

35 CDEF <Maximale Abweichung>

Der Befehl CDEF bestimmt den Winkel der Segmentation, um die maximal erlaubte Abweichung auf einem elementaren Segment zu erhalten.



34 RAD <Modus> <Radius>

Der Modus des Befehls RAD in Verbindung mit den Befehlen CIRR und CIRA zeigt den Quadranten an, an welcher Stelle der Kreis ausgeführt werden soll und mit welchem Radius. Auf jeden Fall wird dieser Befehl überflüssig, wenn der Kreis einem anderen Kreis oder einer Geraden folgt. Das Programm UNIPROG+ kann automatisch den Radius und den Modus generieren, um die Bedingung "Umriss ohne Eckpunkte" zu respektieren.

Der Befehl RAD verbunden mit den Befehlen POINT ermöglicht das Generieren des Radius der Ausrundung. In diesem Falle ist der Modus überflüssig.

Wenn der Radius identisch ist, braucht man ihn für die darauf folgenden Kreise nicht anzugeben.

Ein Kreisbogen wird durch die folgenden Elemente angegeben :

- Die Komponenten der Bogensehne oder den Koordinaten des Bogenendes
- Der Kreisradius
- Der Modus, der eine der 4 möglichen Lösungen auswählt, siehe Abb. 8-2

Der Radius und der Modus sind auf dem ersten Kreisbogen obligatorisch. Für die Folgenden sind sie fakultativ im Falle, dass ein Umriss keine Eckpunkte enthält. Nämlich kann den Radius automatisch bestimmt werden.

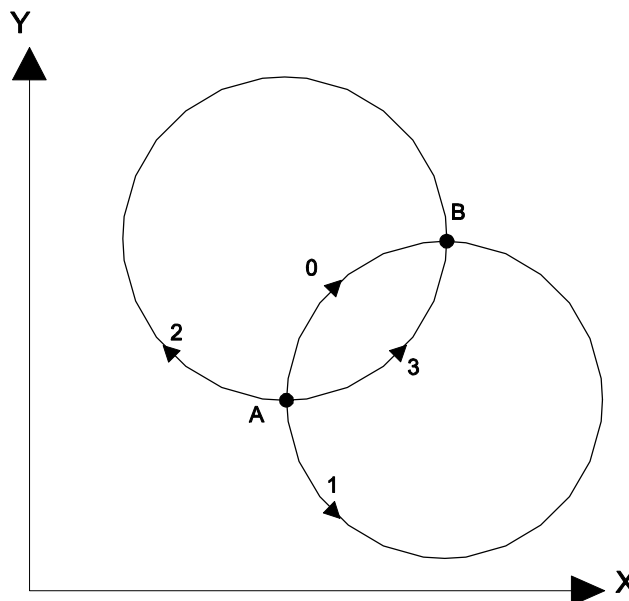


Abb. 8-2 : Kreisbogen-Modi

36	CIRA	<Achse><Koordinate>	<mode-e>
37	CIRR	<Achse><Komponente>	<mode-e>

Die Pseudo-Befehle CIRA und CIRR verhalten sich wie LINA und LINR. Wenn mehrere Kreisbögen im gleichen Umriss den gleichen Radius und gleichen Modus besitzen, kann der Pseudo-Befehl RAD nur einmal eingegeben werden. CDEF und RAD müssen vor CIRA/CIRR kommen.

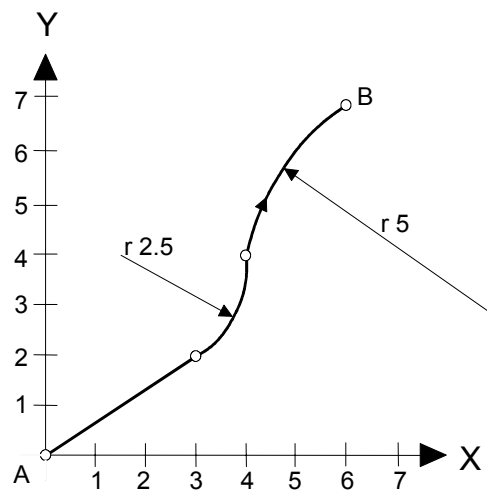


Abb. 8-3 : Beispiel eines Umrisses

Als Beispiel geben wir zwei Methoden an, wie der Umriss nach Abb. 8-3 kodiert werden kann.

DPATH	XY	L	DPATH	XY	L	
LINA	X	30	LINR	X	30	
LINA	Y	22	LINR	Y	22	
RAD	3	2.5	RAD	3	2.5	; nicht obligatorisch
CIRA	X	40	CIRR	X	10	
CIRA	Y	42	CIRR	Y	22	
RAD	0	5	RAD	0	5	; nicht obligatorisch
CIRA	X	60	CIRR	X	20	
CIRA	Y	72	CIRR	Y	32	
END			END			

8.4 Bearbeitung der Umrissdateien

Die Bahn muss vor ihrer Ausführung noch mit Hilfe des Befehls PCOMP (Path COMPUTation) bearbeitet werden. PCOMP erstellt für jeden Umriss eine Tabelle im Speicher (C-MOS). Da die Rechnerleistung begrenzt ist und diese Operation meistens recht lange dauert, ist es nötig, sie während einer Pause des Rechners auszuführen. Zwei Fällen treten auf :

- Der Speicherplatz ist gross genug, um alle Bahnen aufzunehmen. Man setzt also alle PCOMP-Befehle in das Initialisierungsprogramm. Die Bearbeitung erfolgt nur einmal : wenn der Strom eingeschaltet wird oder nach einem Halt durch die Taste STOP.
- Der Speicherplatz ist nicht genügend gross ; man muss das Programm so gestalten, dass die Umrisse während der unbenützten Rechnerzeit bearbeitet werden können, wobei der Bearbeitung genügend Zeit zugestanden werden muss.

Der Befehl PCOMP erstellt die Tabellen im Speicher rotierend, d.h. sobald der Speicher voll ist, beginnt PCOMP wieder am Anfang des Speichers. Bei dieser Arbeitsweise muss natürlich darauf geachtet werden, dass eine Bahn ausgeführt wird, bevor sie von neuen Berechnungen der Bearbeitung überschrieben wird.

Syntaxe des Befehls PCOMP :

31 PCOMP <Datei >

Das Argument ist die Nummer der Datei, in der sich die zu bearbeitenden Umrisse befinden. Falls die Datei mehrere mit DPATH getrennte Umrisse enthält, geht die Bearbeitung bis ans Ende der Datei oder bis zu einer END-Anweisung weiter.

47 TOOLP <Werkzeugnummer> <Nummer der Umrissdatei>

Der Befehl TOOLP bestimmt ein neues Bezugssystem und lädt die Umrissdatei, die zu diesem Werkzeug gehört. Die Komponenten des Verschiebungs-Vektors werden in der Datei '0' gespeichert.

Die Tabelle 'TOOL' gestattet die Überprüfung oder die Modifikation dieser Ursprungspunkte sowie des dazugehörigen Werkzeugdurchmessers.

Diese Ursprungspunkte können auch direkt vom Jogging angelernt und bestätigt werden.

Das Argument 'Werkzeugnummer' wählt die Gruppe der 4 Komponenten, die zu einer Werkzeugnummer gehören.

Wenn das Programm den Befehl TOOLP ausführt, blinkt die rote STOP-LED, um den Berechnungsvorgang für die Umrissdatei anzuzeigen.

Die Umrissdatei wird nicht neu berechnet, wenn sie nicht modifiziert wurde.

Der Umriss benutzt die Werkzeugnummer, um die Werkzeugkorrektur zu bestimmen.
Der Radius wird in der Tabelle 'TOOL' gespeichert.

8.5 Längen-Begrenzung bei der linearen Interpolation

Der Modul S der Strecke (Länge in der Ebene oder im Raum) in Anwender-Einheiten (mm, Zoll,..) bei der Positionierung im Ausführungs-Modus 3 soll nicht überschreiten :

$$S < 67'000'000 / KMUL_{max}$$

Bestimmung von $KMUL_{max}$:

- Für jede durch dem Befehl DPATH ausgewählte Achse wird der Faktor berechnet

$$KMUL(axe) = SCALEK(axe) * DIV(axe)$$

- Der Faktor $KMUL_{max}$ ist der grösste aller Faktoren $KMUL(axe)$:

$$KMUL_{max} = \max(KMUL(axe))$$

Beispiel:

Achsenauswahl : DPATH XY

$$SCALEK(X) = 1000 \text{ Schritt/mm}$$

$$DIV(X) = 300$$

$$KMUL(X) = 300'000$$

$$SCALEK(Y) = 1000 \text{ Schritt/mm}$$

$$DIV(Y) = 200$$

$$KMUL(Y) = 200'000$$

$$KMUL_{max} = 300'000$$

$$S < 223 \text{ mm}$$

8.6 Begrenzung der gesamten Anzahl Segmente

- verfügbarer RAM-Speicher : 30'000 bytes
- Platzbedarf einer Bahn: $11 + S \cdot (3 + 6 \cdot d)$ bytes
S = Anzahl Segmente
d = Anzahl der betroffenen Achsen

Beispiel: Contour rectangulaire fermé avec arrondis sur les 4 angles, résolution circulaire 5 degrés, espace XY.

- $S = 4 + 4 \cdot 18 = 76$ Segmente (4 Seiten und 4 Kreisviertel zu 18 Segmente)
- Platzbedarf : $11 + 76 \cdot (3 + 6 \cdot 2) = 1151$ bytes.

8.7 Ausführung von Umrissen

48 PATH <Geschwind.>

Diese Anweisung war früher der Befehl 38. Das Argument "Dateinummer" ist verschwunden, die Datei wird durch den letzten Befehl TOOLP geladen.

46 CORR <Geschw.>

Der Befehl CORR führt eine geradlinige Bewegung zur Startposition des Umrisses unter Berücksichtigung des Werkzeugradius und der Richtung aus. Die Verfahrgeschwindigkeit ist der Parameter des Befehls.

82 WAITP <Achse> <Geschwindigkeit> <Position> <Mode-w> 66 ENDP

Eine Bahn wird normalerweise mit konstanter Tangentialgeschwindigkeit gefahren. Muss aber die Verfahrgeschwindigkeit an gewählten Punkten der Bahn geändert werden, dann kommt der Befehl WAITP zur Anwendung. Die erste Anweisung WAITP muss **unmittelbar nach PATH** geschrieben werden. Mehrere WAITP können hintereinander vorkommen. Nach der letzten WAITP muss ENDP als Schlussanweisung geschrieben werden, die das Bahnende abwartet.

Wirkungsweise von WAITP:

Der vorgesetzte Befehl PATH ist so behaftet, dass er das Bahnende **nicht** abwartet, sondern nur dessen Ausführung startet.

Mit Mode-w = 0, Abwarten, solange $Istposition < <Position>$,

mit Mode-w = 1, Abwarten, solange $Istposition > <Position>$,

wobei *Istposition* die laufende Position der bezeichneten Achse ist und die Werte algebraisch (mit Vorzeichen) verglichen werden, siehe .

- Die Positionen sind im Werkzeug Bezugssystem gemessen. Der Umriss-Ursprung wird nicht berücksichtigt.
- Die Reaktionszeit dieser Anweisung hängt von der Anzahl der simultanen Abläufen ab, die

in der Steuerung aktiv sind.

- Wenn die programmierte Position nicht erreicht werden kann, muss das Programm mit der STOP-Taste abgebrochen werden.

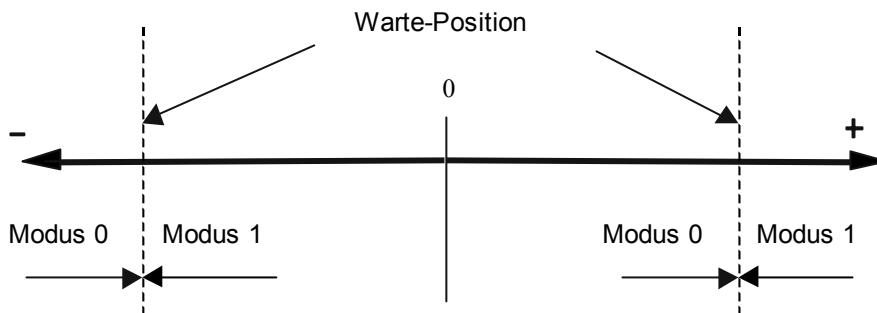
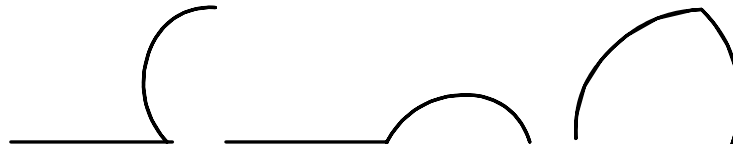


Abb. 8-4 : Modus gemäss Bewegungsrichtung und Warte-Position

8.8 Situationen, wobei die Werkzeugkorrektur nicht funktioniert

UNIPROG+ löst nicht alle Fälle der Werkzeugkorrektur, besonders, wenn Unstetigkeiten oder Eckpunkte vorhanden sind.

Um diese Fälle lösen zu können, muss ein geradliniges Segment als Tangente zum Bogen kreiert werden (minimale Länge von 0.02 mm).



8.9 Anzeige der Bahnrechnungs-Fehler

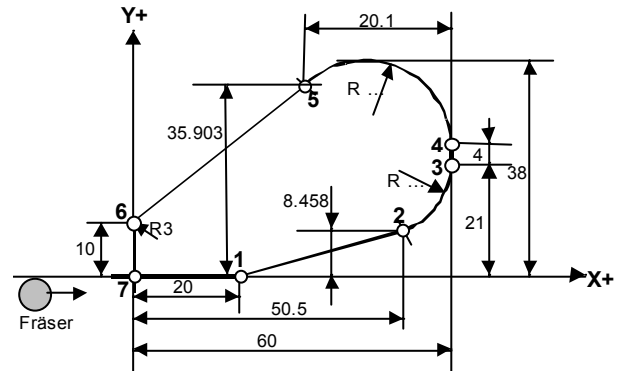
Während des Berechnungsvorgangs können numerierte Fehler auftreten, wenn fehlerhafte Daten erscheinen. Diese Fehler halten die Berechnung an.

- Fehler 0 :** Der Radius ist negativ.
- Fehler 1 :** Der Umriss ist unmöglich. Die Koordinaten für den Endpunkt des Bogens (CIRR CIRA) sind ausserhalb des Radius. Der Radius ist negativ.
- Fehler 2 :** Division durch 0 bei dem Befehl POINT. Der Radius ist negativ.
- Fehler 3 :** Division durch 0 beim automatischen Generieren eines Bogens als Tangente zu einem anderen Bogen.
- Fehler 4 :** Ungenauigkeit bei der Berechnung des Winkels.
- Fehler 5 :** Ungenauigkeit bei der Berechnung des Zentrums.
- Fehler 6 :** Die Festsetzung des Startpunktes ist unmöglich. Das erste Segment (LIN) muss mehr als 0.01 mm messen.
- Fehler 7 :** Die Befehle LINR oder LINA dürfen nicht vor dem Befehl POINT sein.
- Fehler 8 :** 2 geradlinige Segmente hintereinander (LINR, LINA). Dieser Fall wird nur akzeptiert, wenn der Werkzeugradius Null ist.
Die Anwendung des Befehls POINT mit einem Radius Null kann diesen Fall lösen.
- Fehler 9 :** Division durch 0, der Umriss ist unmöglich, die Koordinaten des Bogenendpunktes sind ausserhalb des Radius.
- Fehler 10 :** Es fehlt der Befehl RAD. Die Bestimmung des Radius ist unmöglich, wenn der Umriss mit einem Kreis beginnt.
- Fehler 11 :** Die Koordinaten für die Eingänge des Bogens sind mit denen des Ausgangs gleich.
- Fehler 12 :** Das Generieren der Ausrundung ist unmöglich, die 3 Koordinaten, die den Winkel des Polygons bilden, sind gerade, als Folge ist der Radius unendlich.
- Fehler 13 :** Eine Umrissdatei ist nicht vorhanden.
- Fehler 14 :** Es fehlt der Befehl DPATH in der ersten Zeile der Umrissdatei.
- Fehler 15 :** Im Augenblick der Ausführung erfolgt das Abspeichern des Umrisses (BUFFER) ausserhalb der Speicherzone. Eine fehlerhafte Manipulation hat die erste Zeile der Umrissdatei verändert.
- Fehler 17 :** Die Befehle POINT dürfen nicht auf die Befehle LINA oder LINR folgen.
-
- RAYON < 1 :** Die Bahn zeigt wenigstens einen Radius, der kleiner als 1 mm ist. In der Folge muss die Geschwindigkeit für die Ausführung des Umrisses angepasst werden, um den Radius zu fahren.
- SEGMENTATION NULLE :** Der Bogen ist genügend kurz, um nicht mehr von den geraden Segmenten angenähert zu werden. Er generiert also die Sehne anstatt des Bogens.

8.10 Beispiel eines Umrisses

Werkzeugtabelle

ORIGIN X	TOOL 0	?
ORIGIN Y	TOOL 0	?
ORIGIN Z	TOOL 0	?
ORIGIN U	TOOL 0	?
DIAMETRE	TOOL 0	?
ORIGIN X	TOOL 1	13
ORIGIN Y	TOOL 1	54
ORIGIN Z	TOOL 1	?
ORIGIN U	TOOL 1	?
DIAMETRE	TOOL 1	8



File 1

```
0 01 47 TOOLP 1 23
```

;lädt das Werkzeug 1 und die Umrissdatei 23

;Ursprung X 13 et Y 54

;Werkz.-Durchmesser 8 mm aus Tabelle TOOL

;setzt Werkzeug auf korrigierte Bahn, Vorschub 2

;Führt den durch TOOLP geladenen Umriss 23, Vorschub 1 aus

```
1 01 46 CORR 2
```

;Werkzeug-Rückzug, Vorschub 0

```
2 01 48 PATH 1
```

```
3 01 * 10 POSA X 0 50.0000 0
```

```
4 01 10 POSA Y 0 50.0000 2
```

File 23 Umriss-Unterprogramm

```
0 23 30 DPATH XY L
```

;bestimmt die Ebene XY und Fall 'Material links'

```
1 23 40 ORGP X -1.0000
```

;Umrissbeginn mit 1 mm Schutzabstand in X

```
2 23 35 CDEF 0.01
```

;max. zulässige Abweichung in Kreisbogen 0.01 mm

```
3 23 42 POINT X 20.0000 0
```

```
4 23 34 RAD ? 0.0000
```

;scharfe Ecke, Modus nicht erforderlich

```
5 23 42 POINT Y 0.0000 2
```

;1. Punkt, scharf

```
6 23 32 LINA X 50.5000 0
```

;2. Punkt, mit dem nachfolgenden Bogen berührend

```
7 23 32 LINA Y 8.4580 2
```

```
8 23 36 CIRA X 60.0000 0
```

;3. Punkt, Bogen-Endpunkt

```
9 23 36 CIRA Y 21.0000 2
```

; Radius und Mittelpunkt selbsttätig bestimmt

```
10 23 33 LINR X 0.0000 0
```

;4. Punkt, mit dem nachfolgenden Bogen berührend

```
11 23 32 LINR Y 4.0000 2
```

```
12 23 37 CIRR X -20.100 0
```

```
13 23 36 CIRA Y 35.9030 2
```

;5. Punkt, Bogen-Endpunkt

; Radius und Mittelpunkt selbsttätig bestimmt

```
14 23 42 POINT X 0.0000 0
```

```
15 23 34 RAD ? 3.0000
```

;Abrundungsradius 3 mm, Modus überflüssig

```
16 23 42 POINT Y 10.0000 2
```

;6. Punkt, Abrundungs-Eckpunkt

```
17 23 32 LINA X 0.0000 0
```

```
18 23 32 LINA Y -2.0000 2
```

;7. Punkt, Umrissende, 2 mm Schutzabstand in Y

8.11 Zusammenfassung der Befehle und Pseudo-Befehle für die Erzeugung von Umrissen

Code	Instruction	1 ^{er} arg.	2 ^{ème} arg.	3 ^{ème} arg.	4 ^{ème} arg.	Description	Page
35	CDEF	max. Abweichung				Genauigkeit bei Kreisbogen	48
36	CIRA	Achse	Koord.	Mode-e		Absolute Kreisbogen	50
37	CIRR	Achse	Komp.	Mode-e		Relative Kreisbogen	50
46	CORR	Geschw.				positioniert zum Umrissbeginn	53
30	DPATH	Raum	Werkz. L/R			legt die Ebene fest	46
32	LINA	Achse	Koord.	Mode-e		absolutes Segment	47
33	LINR	Achse	Komp.	Mode-e		relatives Segment	47
40	ORGP	Achse	abs. Pos.			legt den Umriss-Ursprung fest	46
48	PATH	Geschw.				führt den Umriss aus	53
31	PCOMP	Datei				bearbeitet einen Umriss	51
42	POINT	Achse	Ecken-Pos.	Mode-e		absolutes Segment mit Rundung	48
34	RAD	Mode-r	Radius			legt den Radius u. den Kreisbogenmodus fest	49
47	TOOLP	Werkz. Nr.	Datei			legt das Werkz. fest u. bearbeitet einen Umriss	51

9 Anschluss der E-600 Steuerungen

Der hier gegebene Anschluss ist unter dem UNIPROG+ Betriebssystem gültig.

9.1 Einführung

Die Steuerungslogik besteht aus der Tastatur, der INTEL-CPU-Platine und der NEC-Kinematik-Platine, auf die die Module angeschlossen werden.

Intern auf dieser NEC-Leiterplatte sind die Achsen-Stecker durch 0, 1, 2 und 3 hardware-seitig bezeichnet. Das Betriebssystem macht die Zuweisung X=0, Y=1, Z=2, U=3.

Je nach verwendetem Modul können diese Achsenstecker unterschiedliche Arte aufweisen.

Eine Verwechslung dieser Flachkabel-Stecker kann schwere Schaden verursachen.

Der Anschluss der meist verwendeten Leistungsmodule wird hiernächst kurz beschrieben. Ausführliche Angabe sind in den Unterlagen der einzelnen Module zu finden.

Es wird nur die äussere Stiftbelegung angegeben. Man kann aber auch die Zuweisung der Module an die Achsen durch Vertauschung der inneren Flachkabeln anpassen.

9.2 E-600-base, Stromversorgung, Logik und Frontplatte

Drei Sicherungen sind dem Benutzer verfügbar:

- im Netzeingang: 5 Amp 220 V (Netzkabel-Einführung)
- unter der Plastikabdeckung: 6,3A 70 V und 5A 24 V

9.2.1 I/O-Stecker

Die im E600-Base eingebauten Eingänge und Ausgänge, die analogen Signale und die Spannungsversorgung sind an einem Stecker mit der Bezeichnung "I/O" vorhanden.

Stift	Signal und Funktion
A	0 V, gemeinsam für die Ausgänge ²⁾
B	UNIPROG+ OUT(4), hochschaltend, 24 V 0.5 A
C	UNIPROG+ OUT(5), hochschaltend, 24 V 0.5 A
D	UNIPROG+ OUT(6), hochschaltend, 24 V 0.5 A
E	UNIPROG+ OUT(7), hochschaltend, 24 V 0.5 A
F	Analog. Ground für ADC oder WATCHDOG ¹⁾³⁾
G	DAC out, 0..10 V
H	UNIPROG+ OUT(0), hochschaltend, 24 V 0.5 A
J	+ 5 Volt, max. 50mA
K	ADC in
L	IN(2), Endschalter oder Eingang, hochschaltend
M	IN(6), Endschalter oder Eingang, hochschaltend
N	IN(3), Endschalter oder Eingang, hochschaltend
P	IN(7), Endschalter oder Eingang, hochschaltend
R	Analog. Ground für DAC ¹⁾
S	UNIPROG+ OUT(1), hochschaltend, 24 V 0.5 A
T	UNIPROG+ OUT(2), hochschaltend, 24 V 0.5 A
U	UNIPROG+ OUT(3), hochschaltend, 24 V 0.5 A
V	+ 24 V, unreguliert

Tabelle 9-1 : E600, I/O-Stecker, 19-Wege-Burndy

- 1) Die "analogue ground"-Leitungen haben eine begrenzte Strombelastbarkeit und sind deshalb nur als Nullleitung der Anlogschaltungen beabsichtigt. (DAC und ADC).
- 2) Jeder Ausgang kann 400 mA leisten aber der Gesamtstrom darf 4 A nicht überschreiten.
- 3) Von Seriennummer: 707 der I/O-Platine an aufwärts wird diese Leitung zum Ausgang der Software-Überwachung 'WATCHDOG'. Dieser 24V DC, 500 mA –Ausgang ist eingeschaltet, solange das Betriebssystem normal verläuft. Bei gewissen Störungen schaltet er aus.

9.2.2 I/O EXT -Stecker.

An diesem Stecker werden die Eingangsmodule E-500-I1, I2, I3 und das Ausgangsmodul E-500-ODC1 über den I/O-Erweiterungsbus angeschlossen.

9.2.3 RS 232 -Stecker

Durch diesen Schalter werden die folgenden Verbindungen hergestellt:

- PC-Rechner und UNICOM- oder APEX A-600 -Software-Werkzeuge
- Seriell-Drucker oder DOS-Kommando "COPY" bei der Funktion "PRT"

9.3 Verträglichkeit mit E-500

Im Fall, wobei eine E500 eine E600 ersetzt, müssen beim I/O-Stecker die Stifte H, S, T, U weggenommen werden.

Im umgekehrten Fall (wobei eine E600 eine E500 ersetzt) muss die Brücke zwischen den Stiften H und K weggenommen und die Zuweisung der Eingängen überprüft werden.

9.4 E-600-1, Modul, BERGER-Treiber für 5-Phasen-Motoren

Intern werden die Module mit J2, J3, J4 oder J5 durch das Bandkabel verbunden.

Stift	Funktion
A	Wicklung W1A gelb (Original BERGER Kabelfarben)
B	Wicklung W1E weiss
C	Wicklung W2A blau
D	Wicklung W2E rot
E	Wicklung W3A orange
F	Wicklung W3E grün
G	Wicklung W4A grau
H	Wicklung W4E schwarz
J	Wicklung W5A braun
K	Wicklung W5E violett
L	Eingang, hochschaltend, 24 V, IN(i)
M	Eingang, hochschaltend, 24 V, IN(i+4)
N	Näherungsschalerversorgung ,24V
P	Näherungsschalerversorgung , 0V
R,S,T,U	N.C.
V	Schutzerde

Tabelle 9-2 : E600-1, Burndy 19-Wegeschalter

i= hardware-seitige Achsenbezeichnung, siehe Abschn. 9.5

9.5 E-600-3 -Module, 2-Phasen-EIP-Motortreiber

Bipolar-Antrieb, 1600 Mikro-Schritte pro Umdrehung. Sog. "slow/fast decay"-Stromregelung.

Das Flachkabel ist am internen Stecker J2, J3, J4 oder J5 angeschlossen.

Stift	Funktion
A	Wicklungsphase B
B	Wicklungsphase B
C	Wicklungsphase A
D	Wicklungsphase A
E	Eingang, hochschaltend, 24 V, IN(i)
F	Eingang, hochschaltend, 24 V, IN(i+4)
G	Näherungsschalerversorgung 24V
H	Näherungsschalerversorgung 0V

Tabelle 9-3 : E600-3, BURNDY-8-Wege-Stecker

i = Achsennummer, beim Einstecken des Bandkabels gewählt

interner Stecker	J2	J3	J4	J5	J6	J7
i	0	1	2	3	0	1
(Schritt-) Achse	0	1	2	3		
(Servo-) Chanel					0	1

Tabelle 9-4 : Achsenbezeichnung

9.5.1 Stromeinstellung

Durch den Drehschalter wird der dem Motor angemessene Spitzenstrom (peak) eingestellt.

Der Nennwert entspricht dem aktiven 'BOOST'-Signal-Zustand. Wenn unaktiv, ist der Strom auf 60% des Vollwertes reduziert.

Stellung	Strom	Stellung	Strom
0	2.0 A	5	5.3 A
1	2.7 A	6	6.0 A
2	3.3 A	7	6.7 A
3	4.0 A	8	7.3 A
4	4.6 A	9	8.0 A

Tabelle 9-5 : E600-3, Stromeinstellung

10 Anschluss der Steuerung E-600-ND

Dieses Kapitel wird in einer künftigen Auflage dieser Uebersetzung erscheinen.

11 Zusammenfassung

11.1 Befehle

Code	Instruction	1 ^{er} arg.	2 ^{ème} arg.	3 ^{ème} arg.	4 ^{ème} arg.	Beschreibung	Seite
91	ADDD	Adresse				addiert dem Akkumulator	41
86	ANGLE	Geschw.	Wert	Mode-e		Relativ-Verschieb., Polarkoord.	35
22	BRIN0	Eingang	Adresse			springt, wenn Eingang = 0	36
23	BRIN1	Eingang	Adresse			springt, wenn Eingang = 1	36
24	BRM	Adresse				springt, wenn Akk. < 0	40
27	BRNZ	Adresse				springt, wenn Akk <> 0	40
25	BRP	Adresse				springt wenn Akk >= 0	40
26	BRZ	Adresse				springt wenn Akk = 0	40
61	CALL	Adresse				Unterprogramm-Aufruf	39
35	CDEF	max. Abweichung				Genauigkeit bei Kreisbogen	48
36	CIRA	Achse	Koord.	Mode-e		Absolute Kreisbogen	50
37	CIRR	Achse	Komp.	Mode-e		Relative Kreisbogen	50
18	CLOS	Achse	Geschw.			Referenz-Prüfung	32
46	CORR	Geschw.				positioniert zum Bahnbeginn	53
95	CPL	Ausgang				komplementiert den Ausgang	37
59	DECD	Adresse				dekrementiert, direkt (Adresse)	39
94	DIVD	Adresse				dividiert den Akkumulator	41
30	DPATH	Raum	Werkz. I-r			legt die Ebene fest	46
62	END					Programm- u. Unterp.-Ende	39
66	ENDP					Umrissende	53
65	ENDRP					Wiederhol.-Schlaufen-Ende	40
98	FDATA	Reellzahl				definiert eine Reellzahl	41
50	FLOAD	Reellzahl				lädt eine Relle in den Akk.	38
99	IDATA	Ganzzahl				definiert eine Ganzzahl	41
51	ILOAD	Ganzzahl				lädt eine Ganzzahl in den Akk.	38
58	INCD	Adresse				inkrementiert, direkt (Adresse)	39
60	JMP	Adresse				unbedingter Sprung	39
32	LINA	Achse	Coordon.	Mode-e		absolutes Segment	47
33	LINR	Achse	Compos.	Mode-e		relatives Segment	47
52	LOADD	Adresse				lädt den Akk., direkt	38
53	LOADI	Zeiger				lädt den Akk., indirekt	38
85	MOTOR	Motor-Nr.	Drehzahl			Motorumschaltung u. Drehzahl	38
93	MULD	Adresse				multipliziert den Akkumulator	41
90	NOP					No Operation	41
28	OFF	Ausgang				schaltet Ausgang aus	36
29	ON	Ausgang				schaltet Ausgang ein	36
88	ORGA	Déc. Angul.				Winkelverschiebung	35
40	ORGP	Achse	Pos. Abs.			legt den Umriss-Ursprung fest	46
48	PATH	Geschw.				führt den Umriss aus	53
31	PCOMP	Fichier				bearbeitet einen Umriss	51
84	PECK	Achse	Vorschub	End-Pos.	Mode-d	Bohrzyklus	33
42	POINT	Achse	Pos. Angul.	Mode-e		absolutes Segment mit Rundung	48

Code	Instruction	1 ^{er} arg.	2 ^{ème} arg.	3 ^{ème} arg.	4 ^{ème} arg.	Beschreibung	Seite
10	POSA	Achse	Geschw.	Koord.	Mode-e	unmittelb. abs. Positionierung	31
11	POSAD	Achse	Geschw.	Adresse	Mode-e	direkte abs. Positionierung	31
12	POSAI	Achse	Geschw.	Zeiger	Mode-e	indirekte abs. Positionierung	31
14	POSR	Achse	Geschw.	Versch.	Mode-e	unmittelb. rel. Positionierung	32
15	POSRD	Achse	Geschw.	Adresse	Mode-e	direkte rel. Positionierung	32
16	POSRI	Achse	Geschw.	Zeiger	Mode-e	indirekte rel. Positionierung	32
34	RAD	Mode-r	Rayon			legt den Radius u. den Kreisbogenmodus fest	49
87	RADIUS	Geschw.	Wert	Mode-e		Relativ-Verschieb., Polarkoord.	35
17	REF	Achse				Referenz-Aufnahme	32
63	REP	Wiederhol-Zahl				Wiederholschleufe, unmittelbar	40
64	REPD	Adresse				Wiederholschleufe, direkt	40
54	SAVE	Adresse				speichert in EEPROM ab	39
83	SET	Param-Nr.	Wert			Parameter-Festlegung	33
67	SIM1	Adresse				ruft den 1. SIM-Ablauf auf	40
68	SIM2	Adresse				ruft den 2. SIM-Ablauf auf	40
57	SPVEL	Drehzahl				lädt die Spindeldrehzahl	38
55	STORD	Adresse				speichert Akk. ab, direkt	39
56	STORI	Zeiger				speichert Akk. ab, indirekt	39
92	SUBD	Adresse				subtrahiert dem Akk.	41
19	TOOL	Werkz.-Nr.				Werkzeug-Auswahl	32
47	TOOLP	Werkz.-Nr.	Datei			legt das Werkz. fest u. bearbeitet einen Umriss	51
81	TPING	Achse	Steigung	End-Pos.		Gewindebohren	34
70	WAIT	Temps				Pause, unmittelbar	40
20	WAIT0	Eingang				wartet ab, solange Eing. <> 0	36
21	WAIT1	Eingang				wartet ab, solange Eing. <> 1	36
71	WAITD	Adresse				Pause, direkt	40
82	WAITP	Achse	Geschw.	Position	Mode-w	wartet auf Position (Bahn)	53

11.2 Ein- und Ausgänge

Input	Item	Output	Item
0	IN(0)	0	OUT(0)
1	IN(1)	1	OUT(1)
2	IN(2)	2	OUT(2)
3	IN(3)	3	OUT(3)
4	IN(4)	4	OUT(4)
5	IN(5)	5	OUT(5)
6	IN(6)	6	OUT(6)
7	IN(7)	7	OUT(7)
8	SIM0	8	SIM0
9	SIM1	9	SIM1
10	SIM2	0	SIM2
11	FLAG(1)	11	FLAG(1)
12	FLAG(2)	12	FLAG(2)
13	FLAG(3)	13	FLAG(3)
14	FLAG(4)	14	FLAG(4)
15	FLAG(5)	15	FLAG(5)
16..63	IN(16..63)	16..63	OUT(16..63)

11.3 Diverses

Achsen: X = 0 Y = 1 Z = 2 U = 3										
Kanäle: X = 0 Y = 1 touche F1										
Räume:	0	1	2	3	4	5				
	XY	XZ	XU	YZ	YU	ZU				
Pause-Flag: Taste und LED F1 Insert: Taste F3 Delete: Taste F4										