

isel - IMC4-M

Handbuch zur Programmierung der Steuerung IMC4-M

www.isel.com

isel[®]

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

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

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

Stand: 09/2005

Inhaltsverzeichnis

1. Einleitung	5
1.1. Vorwort	5
1.2. Anschluß der Steuerung an einen PC	6
1.3. Umrechnungsfaktoren	7
1.4. Das Koordinatensystem und Bezugspunkte	8
1.5. Relative und Absolute Koordinatenangaben	9
 2. Der DNC-Mode und seine Befehle	 10
2.1. Befehlsaufbau der DNC-Befehle	10
2.2. Die Befehle der IMC4-M im DNC-Mode	11
2.2.1. Initialisierung, Achsenanzahl setzen	11
2.2.2. Referenzfahrt	12
2.2.3. Referenzgeschwindigkeit festlegen	13
2.2.4. Ausführen einer relativen Bewegung	14
2.2.5. Ausführen einer absoluten Bewegung	15
2.2.6. Abfragen der Istposition	16
2.2.7. Nullpunkt setzen	17
2.2.8. 3d Interpolation Ein-/Ausschalten	18
2.2.9. Ebenenwahl für Kreisinterpolation	19
2.2.10. Einstellen der Kreisrichtung für die Kreisinterpolation	20
2.2.11. Kreisinterpolation	21
2.2.12. Starten einer angehaltenen Bewegung	22
2.2.13. Lesen von Ports	23
2.2.14. Schreiben von Ports	24
2.2.15. Testmodus Ein-/Ausschalten	25
2.2.16. Abfragen der Statusdaten der Steuerung	26
2.2.17. Abfragen der Versionsdaten der Steuerung	27
2.2.18. Initialisierung von Parametern	28
2.2.19. Diagnose	29
2.2.20. Selbsttest durchführen	31
2.2.21. Kontroll und SteuerCodes	32
2.2.22. CNC-Programm löschen	33
2.2.23. Bewegung bis Portereignis	34
2.3. Die Berechnung der Kreisparameter	35
2.3.1. Die Parameter für die Kreisinterpolation	35
2.3.2. Die Berechnung der Bogenlänge	35
2.3.3. Die Berechnung des Interpolationsparameters	36
2.3.4. Startpunkt des Kreisbogens	36
2.3.5. Richtungen im Startpunkt der Kreisinterpolation	36
2.3.6. Berechnungsbeispiel zur Kreisinterpolation	37
 3. Der CNC-Mode und seine Befehle	 38
3.1. Befehlsaufbau der CNC-Befehle	38
3.2. Die Befehle der IMC4-M im CNC-Mode	39
3.2.1. CNC-Datenfeld speichern	39
3.2.2. Referenzfahrt im CNC-Mode	40
3.2.3. Relative Bewegung im CNC-Mode	41
3.2.4. Absolute Bewegung im CNC-Mode	42
3.2.5. Nullpunkt setzen im CNC-Mode	43
3.2.6. 3d Interpolation Ein-/Ausschalten im CNC-Mode	44
3.2.7. Ebenenwahl für Kreisinterpolation im CNC-Mode	45
3.2.8. Einstellen der Kreisrichtung für die Kreisinterpolation im CNC-Mode	46
3.2.9. Kreisinterpolation im CNC-Mode	47
3.2.10. Schleifen, Verzweigungen im CNC-Mode	48
3.2.11. Zeitverzögerungen im CNC-Mode	49

3.2.12. Port setzen im CNC-Mode	50
3.2.13. Port lesen und verzweigen im CNC-Mode	51
3.2.14. Datenfeldende im CNC-Mode	52
4. Die Fehlermeldungen der IMC4-M	53
5. Standardportbelegung für isel-Anlagen mit IMC4-M Steuerung	55
6. Steckerbelegung IMC4-M	56

1. Einleitung

1.1. Vorwort

Unsere Firma **isel**-automation ist seit vielen Jahren bekannt für Antriebssysteme und CNC-Steuerungen mit Schrittmotoren. Weltweit haben sich Schrittmotoren im kleinen und mittleren Leistungsbereich durchgesetzt. In Antriebssystemen mit mittleren Genauigkeits- und Dynamikanforderungen sind sie quasi nicht mehr wegzudenken. Schrittmotorsteuerungen arbeiten im Allgemeinen nicht mit geschlossenen Regelkreisen, so dass teure Sensoriken und Auswerteelektroniken entfallen. Dadurch ergibt sich neben einfachem Aufbau und Inbetriebnahme auch ein günstiges Preis-Leistungs-Verhältnis.

Seit vielen Jahren werden Systeme auf Basis der IMC4-Steuerungen von uns angeboten und finden in vielen Bereichen der Produktion, Automatisierung, Forschung und Weiterbildung ihren Einsatz. In Fortführung dieser Produktlinie wurde die Steuerung IMC4-M entwickelt. Die IMC4-M ist eine Schrittmotorsteuerung auf Mikroprozessorbasis zur Ansteuerung von bis zu 4 Achsen über Takt und Richtungssignale. Zur Datenübertragung wird die IMC4-M über eine serielle Schnittstelle angekoppelt. Die verwendeten Befehle im CNC- und DNC-Mode der Steuerung sind kompatibel zur früheren IMC4. Diese Beschreibung soll einen Einblick in die realisierten CNC- und DNC-Befehle der Steuerung geben.

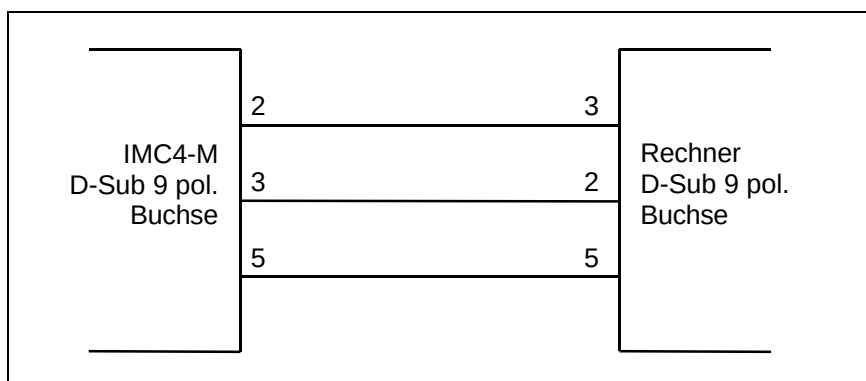
1.2. Anschluß der Steuerung an einen PC

Zur Datenübertragung zwischen IMC4 und Steuerrechner wird eine serielle Schnittstelle nach RS232 eingesetzt. Die Verbindung ist über eine 3-Draht-Leitung realisiert.

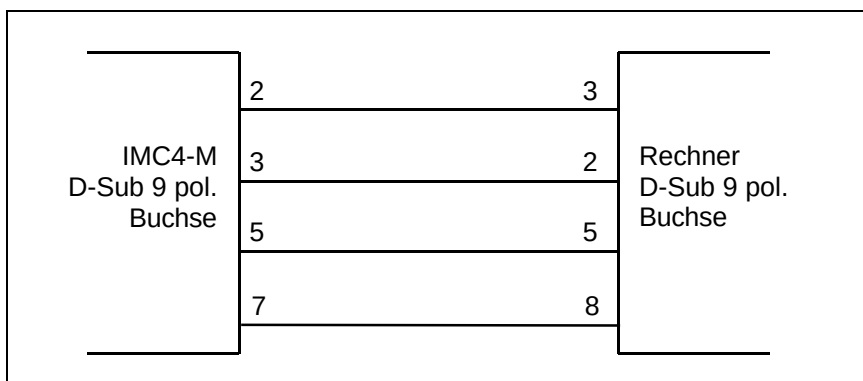
Als Datenübertragungsparameter sind auf der IMC4 folgende Parameter festgelegt:

Baudrate:	19200 (9600 möglich durch entsprechende Jumpereinstellung)
Daten-Bit:	8
Stop-Bit:	1
Parität:	keine

Als Verbindungsleitung zum PC findet eine 3-Drahtverbindung Anwendung:



In Verbindung mit isel-Schaltschränken und isel-Terminals kommt folgendes Kabel zum Einsatz:



Die zusätzliche Leitung wird benutzt, um den isel-Bedienoberflächen (ProNC, Remote) die direkte Überwachung des Starttasters am Bedienterminal zu ermöglichen

1.3. Umrechnungsfaktoren

Die Übergabe der Verfahrswege an die Steuerung erfolgt in Schritten. Im Gegensatz zur Übergabe von metrischen Maßen (mm) ermöglicht diese Art der Darstellung eine schnellere Dekodierung und Ausführung, da die Steuerung intern alle Positionen in Schritten darstellt. Die Umrechnung der Verfahrswege in Schrittmotorschritte muß vom Steuerechner übernommen werden. Für diese Umrechnung werden folgende Informationen über die angeschlossene Mechanik benötigt:

Spindelsteigung: - Steigung der in der Mechanik eingebauten Spindel
 - gibt den Weg an, um den sich bei einer Umdrehung der Spindel der Schlitten bewegt
 - üblich sind Steigungen von 2.5, 4, 5 und 10 mm

Schritte/Umdrehung: - Anzahl der Inkremente pro Umdrehung des Motors

Getriebefaktor: - wird nur bei Schrittmotorantrieben mit Getriebe benötigt

Die Umrechnung von mm in Schritte kann mit folgender Formel erfolgen:

$$\text{Schritte} = \frac{\text{Verfahrweg (mm)}}{\text{Spindelsteigung (mm)}} * \text{Schritte / Umdrehung} * \text{Getriebefaktor}$$

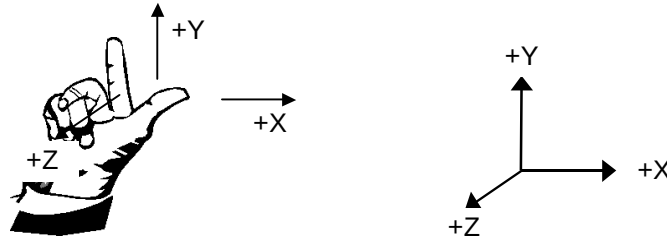
Für Geschwindigkeiten erwartet die Steuerung eine Angabe in Schritten (Inkremente) pro Sekunde. Die Umrechnung von in der Mechanik üblichen Geschwindigkeitsangaben (mm/s oder m/min) kann mit folgenden Formeln erfolgen:

$$V(\text{Schritte / s}) = \frac{V(\text{mm / s}) * \text{Schritt / Umdrehung} * \text{Getriebefaktor}}{\text{Spindelsteigung (mm)}}$$

$$V(\text{Schritte / s}) = \frac{V(\text{mm / min}) * \text{Schritt / Umdrehung} * \text{Getriebefaktor}}{\text{Spindelsteigung (mm)}} * \frac{60}{100}$$

1.4. Das Koordinatensystem und Bezugspunkte

Die Festlegung von Koordinatensystemen und Bezugspunkten ist eine notwendige Voraussetzung für die Programmierung von Bearbeitungsbewegungen innerhalb des Arbeitsraumes von Werkzeugmaschinen. In Übereinstimmung mit der DIN 66217 wird ein rechtshändiges, rechtwinkliges Koordinatensystem mit den Achsen X, Y und Z verwendet. Die Z-Achse ist identisch mit der Achse der Bearbeitungsspindel. Die positive Richtung der Z-Achse verläuft vom Werkstück zum Werkzeug.



Zur Erzeugung der Bearbeitungsbewegungen kann entweder das Werkzeug oder das Werkstück bewegt werden. Das Koordinatensystem bezieht sich aber unabhängig davon immer auf das Werkstück. Somit ist für die Programmierung unerheblich ob sich Werkzeug oder Werkstück bewegen. Der Programmierer geht immer davon aus, daß sich das Werkzeug relativ zum Koordinatensystem des stillstehenden Werkstücks bewegt.

Neben dem Koordinatensystem spielen Bezugspunkte bei der Programmierung von Werkzeugmaschinen eine wesentliche Rolle. Die wichtigsten seien hier genannt:

Maschinennullpunkt: - liegt im Ursprung des Anlagenkoordinatensystems unveränderlich fest
 - ist durch die Anlagenkonstruktion vorgegeben und kann nicht verändert werden

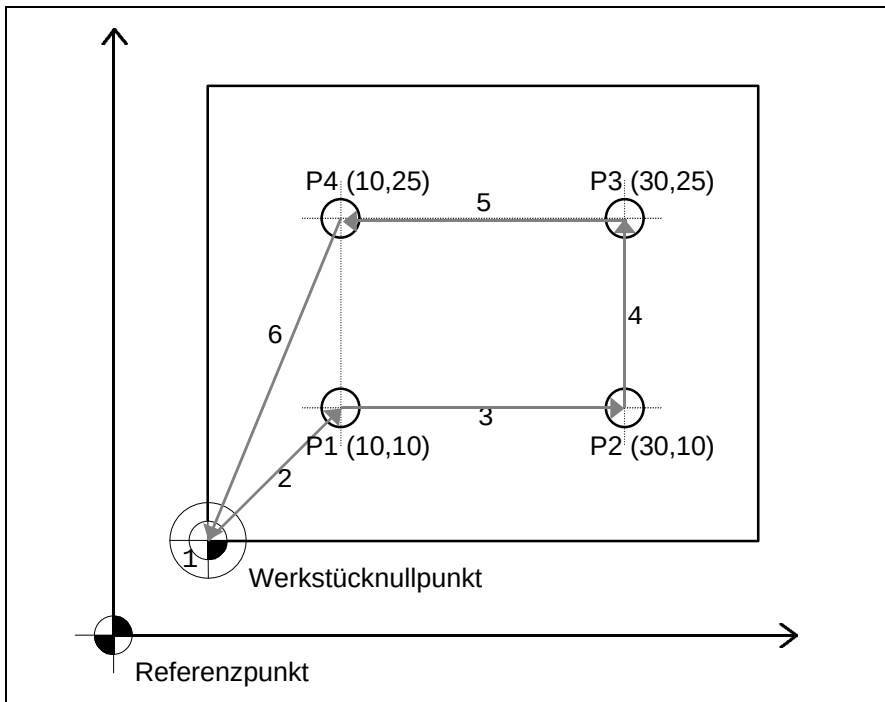
Referenzpunkt: - durch Endschalter auffindbarer, festgelegter Punkt, der meist in einer äußeren Ecke des Arbeitsraumes liegt und oft identisch mit dem Maschinennullpunkt ist
 - kann auch auf einen festen Abstand zum Maschinennullpunkt eingestellt werden und bleibt dann unverändert
 - kann meist mit der Genauigkeit von einem Weginkrement angefahren werden

Werkstücknullpunkt: - Ursprung des Werkstückkoordinatensystems
 - kann vom Programmierer frei gewählt werden und ist Bezugspunkt für die Programmierung

Die Festlegungen Ihre Anlage betreffend entnehmen Sie bitte den entsprechenden Bedienanleitungen und/oder Hardware- und Mechanikbeschreibungen.

1.5. Relative und Absolute Koordinatenangaben

Verfahrwege können in relativen (Kettenmaß) oder absoluten Koordinaten programmiert werden. Bei der Angabe von relativen Koordinaten ist der Bezugspunkt die aktuelle Werkzeugposition. Die Koordinaten stellen dann den Abstand des nächsten anzufahrenden Punkts von der aktuellen Position dar. Im Gegensatz dazu dient bei der Angabe von absoluten Koordinaten der Werkstücknullpunkt als Bezugspunkt. Die Koordinaten stellen dann den Abstand des nächsten anzufahrenden Punkts vom Werkstücknullpunkt dar. Folgendes Beispiel soll dies verdeutlichen:



Es sollen 4 Bohrungen in ein Werkstück an den Punkten P1, P2, P3 und P4 eingebracht werden. Dabei sollen die Punkte in der Reihenfolge P1 --> P2 --> P3 --> P4 angefahren werden.

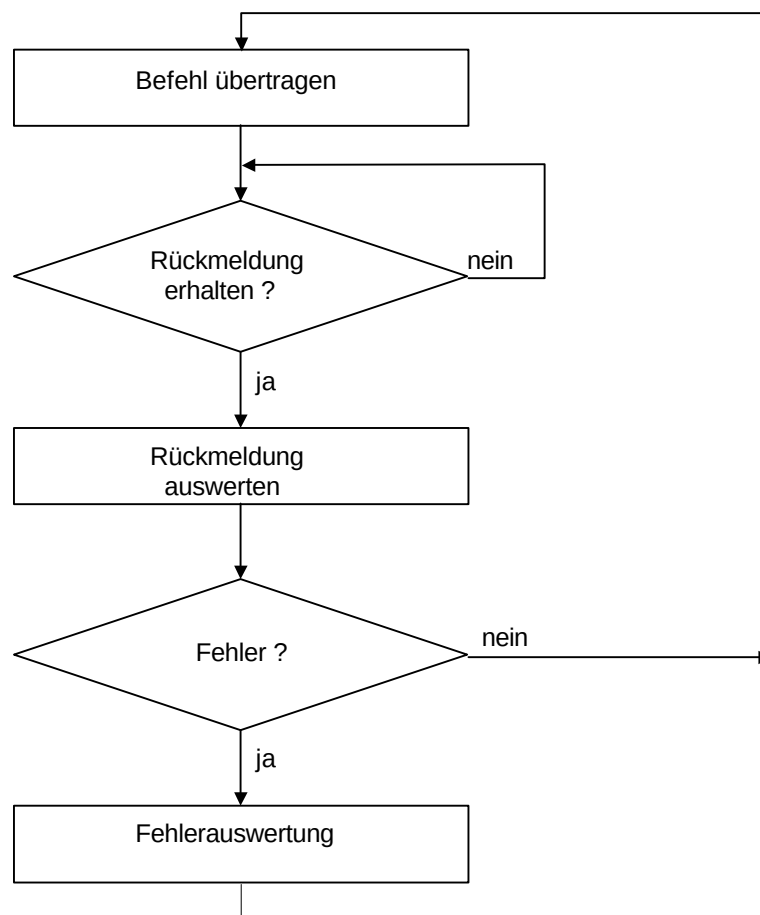
Für die Abarbeitung ergeben sich folgende relativen und absoluten Koordinaten:

	absolut		relativ	
1	0,	0	0,	0
2	10,	10	10,	10
3	30,	10	20,	0
4	30,	25	0,	15
5	10,	25	-20,	0
6	0,	0	-10,	-25

2. Der DNC-Mode und seine Befehle

2.1. Befehlsaufbau der DNC-Befehle

Im DNC-Modus betrieben, werden die von einem Steuerrechner übergebenen Datensätze, bzw. Befehle direkt ausgewertet und ausgeführt. Hierzu ist zu Beginn der Datenkommunikation eine sogenannte Initialisierung notwendig. Sie besteht aus dem Dateneröffnungszeichen @, der Gerätenummer (0=Standard) und der Anzahl der zu verfahrenen Achsen. Anschließend werden der Steuerung die Programmschritte einzeln übergeben und von ihr direkt ausgeführt. Zur Überprüfung der Datenübertragung bzw. Meldung von aufgetretenen Fehlern werden über die Schnittstelle entsprechende ASCII-Zeichen an den Steuerrechner zurückgesendet. Dieses sogenannte Software-Handshake-Verfahren wird folgendermaßen realisiert:



Zunächst wird ein Befehl an die Steuerung übertragen. Der Befehl wird von der Steuerung dekodiert und abgearbeitet, anschließend generiert die Steuerung ein entsprechendes Quittierungs- oder Fehlerzeichen.

Diese Rückmeldung wird vom Steuerrechner ausgewertet. Ist ein Fehler aufgetreten muß eine entsprechende Fehlerauswertung und -beseitigung durchgeführt werden. Anschließend kann der nächste Befehl auf gleiche Art und Weise an die Steuerung übertragen werden.

Im Folgenden wird der Befehlsumfang des DNC-Modes der Steuerung IMC4-M beschrieben.

2.2. Die Befehle der IMC4-M im DNC-Mode

2.2.1. Initialisierung, Achsenanzahl setzen

Befehl: Achsenanzahl setzen

Zweck: Durch Übergeben der Achsenanzahl wird die Steuerung neu initialisiert.

Aufbau: @<GN><Achsen><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
<Achsen>	= Achsenangabe, s. u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @07, @08

Erläuterung: Die Steuerung wird durch "@0" adressiert, der anschließende Zahlenwert beinhaltet die Achsenkonfiguration. Dabei wird intern jede Achse durch ein Bit eines Binärwertes repräsentiert und somit ergeben sich folgende Werte:

1 --> X Achse
3 --> X+Y Achse
7 --> X+Y+Z Achse
8 --> A Achse

Beschränkung: Nicht zulässig sind die Kombinationen "@00", "@02", "@04", "@06", sowie "@09".
ACHTUNG: Die A-Achse muß immer separat initialisiert werden.

2.2.2. Referenzfahrt

Befehl: Referenzfahrt

Zweck: Die Steuerung verfährt alle angegebenen Achsen an ihre Nullpunkte (Referenzpunkte). Die Referenzpunkte der Achsen sind bei isel-Systemen immer in einer sinnvollen Defaultanordnung festgelegt, können aber durch entsprechende Initialisierungsbefehle angepaßt werden.

Aufbau: @<GN>R<Achsen><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
R oder r	= Befehl Referenzfahrt
<Achsen>	= Achsenangabe, s. u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0R7, @0r7, @0R8

Erläuterung: Die Steuerung wird durch "@0" adressiert. "R" gibt an, daß eine Referenzfahrt ausgeführt werden soll. Der anschließende Zahlenwert definiert die Achsen, die eine Referenzfahrt durchführen sollen. Dabei wird intern jede Achse durch ein Bit eines Binärwertes repräsentiert und somit ergeben sich folgende Werte:

1 --> X Achse
2 --> Y Achse
3 --> X+Y Achse
4 --> Z Achse
5 --> X+Z Achse
6 --> Y+Z Achse
7 --> X+Y+Z Achse
8 --> A Achse

Die Reihenfolge der Ausführung ist dabei wie folgt festgelegt:
Z-Achse --> Y-Achse --> X-Achse --> A-Achse

Nach erfolgter Referenzfahrt sendet die Steuerung ihr Quittierungszeichen und wartet auf die nächsten Befehle. Die Steuerung kann erst wieder Befehle verarbeiten, nachdem die Referenzfahrt durch die Mechanik ausgeführt worden ist.

Beschränkung: Der Befehl kann erst verwendet werden, nachdem eine Initialisierung der Steuerung durch den Befehl "Achsenanzahl setzen" erfolgt ist und beschränkt sich auf die dort vorgegebene Achsenkonfiguration. Bei falscher Achsangabe erfolgt die Fehlerrückmeldung "3". Befindet sich die Steuerung im 3D-Modus, schaltet der Befehl wieder in die 2,5 dim. Ausführung zurück.

ACHTUNG: Die A-Achse muß immer separat referenziert werden.

ACHTUNG: Bei nicht angeschlossenem Referenzschalter, wird die entsprechende Achse permanent angesteuert. Durch Betätigen des Stop-Tasters besteht jedoch die Möglichkeit den Referenzschalter zu simulieren.

2.2.3. Referenzgeschwindigkeit festlegen

Befehl: Referenzgeschwindigkeit setzen

Zweck: Der Befehl definiert, getrennt für jede Achse die Geschwindigkeit, mit der eine Referenzfahrt ausgeführt wird.

Aufbau:	@<GN>d<Gx><CR>	(x)
	@<GN>d<Gx>,<Gy><CR>	(x-y)
	@<GN>d<Gx>,<Gy>,<Gz><CR>	(x-y-z)
	@<GN>d<Gx>,<Gy>,<Gz>,<Ga><CR>	(x-y-z-a)

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
d	= Befehl Referenzgeschwindigkeit setzen
<Gx>	= Referenzgeschwindigkeit x
<Gy>	= Referenzgeschwindigkeit y
<Gz>	= Referenzgeschwindigkeit z
<Ga>	= Referenzgeschwindigkeit a
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0d2500, @0d2400,3000, @0d1000,3000,2000, @0d1000,3000,2000,2000

Erläuterung: Werden der Steuerung keine Informationen zur Referenzgeschwindigkeit übergeben, erfolgt die Ausführung mit einem Default-Wert. Ein geänderter Wert bleibt beim Ausschalten nicht erhalten.

Beschränkung: Die angegebenen Geschwindigkeiten müssen innerhalb des gültigen Wertebereiches für Geschwindigkeiten liegen.
Eine zu hoch gewählte Referenzgeschwindigkeit kann in Verbindung mit einer großen Spindelsteigung, durch die vorhandene Massenträgheit, zu einer Beschädigung der Referenzschalter führen. Die Steuerung benötigt eine Schalthysterese des angeschlossenen Nullagenschalters. Dies ist bei Anschluß von elektronischen Nullsensoren zu beachten!

2.2.4. Ausführen einer relativen Bewegung

Befehl: Bewegung relativ

Zweck: Die Steuerung generiert entsprechend der übergebenen Schrittzahl und Schrittgeschwindigkeit eine relative Bewegung. Die Verfahrbewegung wird sofort ausgeführt.

Aufbau: @<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
@<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
A oder a	= Befehl Bewegung relativ
<Sx>	= Schrittzahl x
<Gx>	= Geschwindigkeit
<Sy>	= Schrittzahl y
<Gy>	= Geschwindigkeit
<Sz>, <Sz1>	= Schrittzahl z
<Sz2>	= Schrittzahl z, 2. Bewegung bei 2.5d, 3 Achsen
<Gz>, <Gz1>	= Geschwindigkeit
<Gz2>	= Geschwindigkeit z, 2. Bewegung bei 2.5d, 3 Achsen
<Sa>	= Schrittzahl a, bei 4 Achsen
<Ga>	= Geschwindigkeit, bei 4 Achsen
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0A 5000,900 (nur x-Achse)
@0A 50,900,20,9000 (x und y-Achse)
@0A 30,800,10,900,4,90,-4,30 (x,y und z-Achse, bei 3 Achsen)
@0A 30,800,10,900,4,90,-4,30 (x,y,z und a-Achse, bei 4 Achsen)

Erläuterung: Die Steuerung wird durch "@0" adressiert; "A" oder "a" gibt an, daß eine relative Bewegung erfolgen soll. Die Steuerung erwartet nun für jede Achse ein Zahlenpaar bestehend aus Schrittzahl und Geschwindigkeit. Die Bewegung erfolgt im Relativmaß, d.h. bezogen auf die letzte Position. Die Anzahl der Angaben muß mit der Achsenzahl übereinstimmen, d.h. ein Parameterpaar bei x-Betrieb, zwei Parameterpaare bei xy-Betrieb, vier Parameterpaare für xyz-Betrieb und vier Parameterpaare für xyza-Betrieb. Die einzelnen Zahlen müssen durch Kommata getrennt werden. Für die z-Achse werden bei 3 Achsen und 2,5d zwei Zahlenpaare erwartet, da für Bearbeitungsanwendungen sehr häufig die Situation "Fahren, Werkzeug absenken und anschließend anheben" vorkommt. Im 2.5d-Interpolationsbetrieb werden zuerst die Bewegungen der x- und y-Achse verfahren (linear interpoliert), anschließend wird die z-Achse zuerst um die in z1-angegebenen und dann um die in z2-angegebenen Werte verfahren. Besteht die Notwendigkeit nur eine Achse zu bewegen, so sind dennoch für alle initialisierten Achsen Werte zu übertragen. Dabei ist für die Schrittzahlen der nicht bewegten Achsen 0 anzugeben. Die Steuerung meldet sich nach erfolgter Ausführung mit dem Handshake-Charakter ("0"). Die Steuerung kann erst wieder nach Ausführung des Befehles neue Befehle verarbeiten.

Beschränkung: Der Befehl kann erst verwendet werden, nachdem die Achsenanzahl gesetzt worden ist. Darüberhinaus prüft die Steuerung nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verläßt.
Im 2.5d-Interpolationsbetrieb wird die Geschwindigkeitsangabe der Achse mit dem längsten Weg als Bahngeschwindigkeit übernommen und die Geschwindigkeit der anderen Achsen entsprechend dem Wegeverhältnis angepaßt. Dem gegenüber wird im 3d-Interpolationsbetrieb die Geschwindigkeitsangabe der x-Achse als Vorgabewert für die Bahngeschwindigkeit herangezogen.

2.2.5. Ausführen einer absoluten Bewegung

Befehl: Bewegung zur absoluten Position

Zweck: Die Steuerung fährt mit den angegebenen Geschwindigkeiten an die angegebene Position. Die Verfahrbewegung wird sofort ausgeführt.

Aufbau: @<GN>M<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
@<GN>M<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
M	= Befehl Bewegung absolut
<Sx>	= Position x
<Gx>	= Geschwindigkeit
<Sy>	= Position y
<Gy>	= Geschwindigkeit
<Sz>, <Sz1>	= Position z
<Sz2>	= Position z, 2. Bewegung immer = 0
<Gz>, <Gz1>	= Geschwindigkeit
<Gz2>	= Geschwindigkeit
<Sa>	= Position a, bei 4 Achsen
<Ga>	= Geschwindigkeit, bei 4 Achsen
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0M 5000,900 (nur x-Achse)
@0M 50,900,20,9000 (x und y-Achse)
@0M 30,800,10,900,4,90,0,30 (x,y und z-Achse, bei 3 Achsen)
@0M 30,800,10,900,4,90,4,30 (x,y,z und a-Achse, bei 4 Achsen)

Erläuterung: Die Steuerung wird durch "@0" adressiert. "M" gibt an, daß eine Absolut-Position folgt. Aus Kompatibilitätsgründen zum relativen Positionierbefehl werden auch hier bei 3 Achsen für die z-Achse zwei Zahlenpaare erwartet. Die zweite Positionsangabe der z-Position muß dann jedoch Null sein und wird ignoriert. Die Steuerung meldet sich nach erfolgter Ausführung mit dem Handshake-Charakter. Die Steuerung kann erst wieder nach Ausführung des Befehles neue Befehle verarbeiten.

Beschränkung: Der Befehl kann erst verwendet werden, nachdem die Achsenanzahl gesetzt worden ist. Die Steuerung prüft nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verläßt.

2.2.6. Abfragen der Istposition

Befehl: Positionsabfrage

Zweck: Die Steuerung gibt die momentane Istposition aller Achsen an den übergeordneten Rechner zurück.

Aufbau: @<GN>P<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
P	= Befehl Positionsabfrage
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0P

Erläuterung: Die Steuerung wird durch "@0" adressiert. "P" gibt an, daß eine Positionsabfrage erfolgt. Die Steuerung bestätigt dies mit dem Handshake-Charakter und gibt anschließend im hexadezimalen Format die Positionswerte aller Achsen aus (bei bis zu 3 Achsen 18 hexadezimale Ziffern, bei 4 Achsen 24 hexadezimale Ziffern).

Der Aufbau der rückgemeldeten Position ist folgendermaßen:

z. Bsp.: 00010002000FFFFFE für 3 Achsen

Position x = 000100, hexadezimal im 2er-Komplement, entspricht 256 dezimal.
Position y = 02000F, hexadezimal im 2er-Komplement, entspricht 131087 dezimal.
Position z = FFFFFE, hexadezimal im 2er-Komplement, entspricht -2 dezimal.

z. Bsp.: 000B00044000000FFE003040 für 4 Achsen

Position x = 000B00, hexadezimal im 2er-Komplement, entspricht 2816 dezimal.
Position y = 044000, hexadezimal im 2er-Komplement, entspricht 278528 dezimal.
Position z = 000FFE, hexadezimal im 2er-Komplement, entspricht 4094 dezimal.
Position a = 003040, hexadezimal im 2er-Komplement, entspricht 12352 dezimal.

Beschränkung: Der Befehl kann nur verwendet werden, wenn keine Verfahrbewegung stattfindet (wenn sich die Anlage im Stop-Zustand befindet). Die Steuerung kann nicht prüfen, ob die Istposition der aktuellen Position der Mechanik entspricht, da kein Regelkreis vorhanden ist.

ACHTUNG: Es werden bei bis zu 3 Achsen immer die Positionen für drei Achsen durch die Funktion zurückgegeben, unabhängig von der Anzahl der definierten Achsen. Bei 4 Achsen werden immer Positionen für 4 Achsen zurückgegeben.

2.2.7. Nullpunkt setzen

Befehl: Nullpunkt am aktuellen Punkt setzen

Zweck: Die Steuerung speichert die momentane Position als virtuellen Nullpunkt für die angegebene(n) Achse(n). Die nächsten "Verfahre absolut"-Anweisungen berücksichtigen diesen virtuellen Nullpunkt als neuen Bezugspunkt.

Aufbau: @<GN>n<Achsen><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
n	= Befehl Nullpunkt setzen
<Achsen>	= Achsenangabe, s. u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0n7, @0n1, @0n8

Erläuterung: Die Steuerung wird durch "@0" adressiert. "n" gibt an, daß eine Nullpunktverschiebung vorgenommen werden soll. Nach dem Befehl werden der Steuerung die Achsen mitgeteilt, für die eine Nullpunktverschiebung durchgeführt werden soll. Dabei wird intern jede Achse durch ein Bit eines Binärwertes repräsentiert und somit ergeben sich folgende Werte:

1 -->	X Achse
2 -->	Y Achse
3 -->	X+Y Achse
4 -->	Z Achse
5 -->	X+Z Achse
6 -->	Y+Z Achse
7 -->	X+Y+Z Achse
8 -->	A Achse

Die Steuerung meldet sich nach erfolgter Ausführung mit einer Rückmeldung.

Beschränkung: Der virtuelle Nullpunkt hat nur für den Befehl "Verfahre absolut" eine Bedeutung. Relativpositionierung wird vom virtuellen Nullpunkt nicht beeinflusst, da hier ein relativer Fahrvektor angegeben wird.

ACHTUNG: Die Nullpunktverschiebung für die A-Achse muß immer separat durchgeführt werden.

2.2.8. 3d Interpolation Ein-/Ausschalten

Befehl: 3D-Linearinterpolation Ein-/Ausschalten

Zweck: Die Steuerung erweitert die Interpolation von 2.5D auf 3D. Durch Verwenden des Befehles kann die 3D-Interpolation gezielt aus- und eingeschaltet werden.

Aufbau: @<GN>z<Status><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
z	= Befehl 3d Intrpolation
<Status>	= 0 --> Ausschalten, 1 --> Einschalten
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0z1, @0z0

Erläuterung: Die Steuerung wird mit dem Dateneröffnungsteil "@0" auf einen neuen Befehl vorbereitet. "z1" ändert die Interpolation von 2D- auf 3D-Betrieb. Die Anweisung wirkt modal, d. h. alle relativen und absoluten Bewegungen werden dreidimensional ausgeführt. Die Angabe von z2-Parametern bei 3 Achsen in diesen Verfahrbewegungen wird ignoriert. Die Geschwindigkeitsangabe der Interpolation muß bei der x-Angabe erfolgen. Bei 4 Achsen wird die 4.Achse entsprechend nachgeführt.

2.2.9. Ebenenwahl für Kreisinterpolation

Befehl: Ebenenwahl

Zweck: Einstellen der Interpolationsebene für die Kreisinterpolation. Kreise sind nur innerhalb einer Ebene definiert. Die Defaultebene für die Kreisinterpolation ist die XY-Ebene. Durch den "Ebenenwahl"-Befehl besteht hier jedoch die Möglichkeit jede andere Ebenenkonfiguration als Kreisebene zu definieren.

Aufbau: @<GN>e<Ebene><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
e	= Befehl Kreisebene setzen
<Ebene>	= Ebenenangabe, s. u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0e1, @0e0

Erläuterung: Die Steuerung wird durch "@0" adressiert. "e" gibt an, daß die Ebene für die Kreisinterpolation eingestellt werden soll. Der anschließende Zahlenwert definiert die Ebene in folgender Weise:

0 --> XY Ebene
1 --> XZ Ebene
2 --> YZ Ebene

Beschränkung: Dieser Befehl ist modal wirksam, d.h. eine Ebenenwahl für die Kreisinterpolation bleibt erhalten bis sie von einer erneuten Ebenenwahl überschrieben wird.

2.2.10. Einstellen der Kreisrichtung für die Kreisinterpolation

Befehl: Kreisrichtung einstellen

Zweck: Einstellen der Kreisrichtung für die Kreisinterpolation. Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle ausgelöst. Der erste Befehl legt die Kreisrichtung fest, im Zweiten (s. 2.2.11.) werden die Interpolationsparameter übergeben.

Aufbau: @<GN>f<Richtung><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
f	= Befehl Kreisrichtung einstellen
<Richtung>	= 0 --> CW (Uhrzeigersinn), -1 --> CCW (gegen Uhrzeigersinn)
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0f-1, @0f0

Erläuterung: Die Steuerung wird durch "@0" adressiert. "f" gibt an, daß die Richtung für die Kreisinterpolation eingestellt werden soll. Der anschließende Zahlenwert definiert die Richtung in folgender Weise:

0 --> CW (Kreisinterpolation im Uhrzeigersinn)
-1 --> CCW (Kreisinterpolation entgegen dem Uhrzeigersinn)

Beschränkung: Die Richtung für die Kreisinterpolation ist prinzipiell vor jeder Kreisbewegung zu programmieren.

2.2.11. Kreisinterpolation

Befehl: Kreisinterpolation

Zweck: Bearbeiten von Kreisen und Kreisbögen mit konstanter Bahngeschwindigkeit. Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle ausgelöst. Der erste Befehl legt die Kreisrichtung fest (s. 2.2.10.), im Zweiten werden die Interpolationsparameter übergeben.

Aufbau: @<GN>y,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
	= Bogenlänge in Schritten
<V>	= Geschwindigkeit
<D>	= Interpolationsparameter
<Xs>	= Startpunkt x
<Ys>	= Startpunkt y
<Rx>	= Richtung x
<Ry>	= Richtung y
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0y400,1500,119,-141,141,-1,-1

Erläuterung: Die Steuerung wird durch "@0" adressiert. "y" gibt an, daß eine Kreisinterpolation ausgeführt werden soll. Die Bogenlänge gibt die Länge des Bogens in Schritten zwischen dem Start- und Endpunkt der Kreisinterpolation an. Als Geschwindigkeitsangaben sind alle ganzzahligen Werte innerhalb des gültigen Wertebereiches für Geschwindigkeiten zulässig. Der Interpolationsparameter muß übergeben werden, da die Steuerung aufgrund ihrer Speicherkapazität nicht in der Lage ist diesen Parameter selbst zu berechnen. Die Parameter Xs und Ys geben den Startpunkt des Bogens relativ zum Kreismittelpunkt an. Rx und Ry geben an, in welchem Quadranten des Kreises die Interpolation startet. Die Steuerung meldet sich nach erfolgter Ausführung mit dem Handshake-Charakter ("0"). Die Steuerung kann erst wieder nach Ausführung des Befehles neue Befehle verarbeiten.

ACHTUNG: Zur Berechnung der Parameter lesen Sie bitte das Kapitel "Berechnung der Parameter für die Kreisinterpolation".

Beschränkung: Der Befehl kann erst verwendet werden, nachdem die Achsenanzahl gesetzt worden ist. Darüberhinaus prüft die Steuerung nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verläßt.

2.2.12. Starten einer angehaltenen Bewegung

Befehl: Start

Zweck: Eine angehaltene Bewegung soll fortgeführt werden.

Aufbau: @<GN>S<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
S oder s	= Befehl Start
<CR>	= Carriage Return als Befehlsabschluß

Erläuterung: Die Steuerung wird durch "@0" adressiert. "S" gibt an, daß eine angehaltene Bewegung gestartet und somit der Rest der eigentlichen Bewegung zur Ausführung gebracht werden soll. Die Steuerung meldet sich nach erfolgter Ausführung mit dem Handshake-Charakter ("0") oder mit einer Fehlermeldung, falls kein Bewegungsrest im Speicher vorhanden ist. Die Steuerung kann erst wieder nach Ausführung der restlichen Bewegung neue Befehle verarbeiten.

Beschränkung: Der Befehl ist nur sinnvoll, wenn eine Bewegung angehalten wurde und sich die Steuerung im Stopmode befindet. In diesem Mode kann über den Befehl „Diagnose“ (siehe 2.2.19. Diagnose) der aktuelle Status des Starttasters abgefragt und bei dessen Betätigung vom Bedienprogramm ein Startbefehl erzeugt werden. Im DNC-Mode ist eine Bedienung über die Tasten der Steuerung nur auf diese Weise möglich, da die Steuerung nicht direkt auf die Starttaste reagiert.

2.2.13. Lesen von Ports

Befehl: Port lesen

Zweck: Der Befehl ermöglicht den aktuellen Zustand von logischen oder physikalischen Eingangsports über die serielle Schnittstelle zu ermitteln.

Aufbau: @<GN>b<Portnr><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
b	= Befehl Port lesen
<Portnr>	= Portnummer s.u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0b0, @0b1

Erläuterung: Die Steuerung wird durch "@0" adressiert. "b" gibt an, daß der Status eines Eingabeports ermittelt werden soll. Anschließend wird die Portnummer übermittelt und der Befehl mit Carriage Return abgeschlossen. Die Steuerung antwortet mit dem Software-Handshake "0" gefolgt von zwei Zeichen, die einen Hexadezimalwert angeben, der dem aktuellen Status des Eingangsports entspricht. Für die Steuerung IMC4 sind folgende Ports mit entsprechenden Funktionalitäten definiert:

Port	Zustand	Funktion
0	00 - FF	User E/A (auch noch mit 65531 möglich)
		Bit0 Eingang Stoptaster
		Bit1 Eingang Starttaster
		Bit2 Anwender Eingang 1
		Bit3 Eingang PowerOK
		Bit4 Eingang Notausschalter
		Bit5 Eingang Eintaster
		Bit6 Eingang Haubenschalter
		Bit7 Eingang Schlüsselschalter
1	00	Haube ist geöffnet
	01	Haube ist geschlossen
2	00	Spindel ist ausgeschaltet
	01	Spindel ist eingeschaltet
3	00	Motorströme sind ausgeschaltet
	01	Motorströme sind eingeschaltet
5	00 - FF	User E/A I2C-Porterweiterung (I2C-Adresse=0)
6	00 - 1F	User E/A
		Bit0 Anwender Eingang 1
		Bit1 Anwender Eingang 2
		Bit2 Anwender Eingang 3
		Bit3 Anwender Eingang 4
		Bit4 Anwender Eingang 5

Beschränkung: Eine Rückgabe der Portzustände erfolgt nur, wenn die Steuerung mit dem Software-Handshake "0" antwortet. Der Befehl kann nur verwendet werden, wenn keine Bewegung abgearbeitet wird.

2.2.14. Schreiben von Ports

Befehl: Port schreiben

Zweck: Der Befehl erlaubt es, logische oder physikalische Ausgangsports mit definierten Werten über die serielle Schnittstelle zu beschreiben.

Aufbau: @<GN>B<Portnr>,<Wert><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
B	= Befehl Port schreiben
<Portnr>	= Portnummer s.u.
<Wert>	= neuer Portwert
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0B1,1

Erläuterung: Die Steuerung wird durch "@0" adressiert. "B" gibt an, daß der Wert eines Ausgabeports gesetzt werden soll. Anschließend wird die Portnummer und der neue Portwert getrennt durch Komma übermittelt und der Befehl mit Carriage Return abgeschlossen. Die Steuerung antwortet mit dem Software-Handshake "0", falls die Ausführung erfolgreich war, oder mit einer Fehlermeldung, falls falsche Portnummern und/oder Werte übergeben wurden. Für die Steuerung IMC4 sind folgende Ports mit entsprechenden Funktionalitäten definiert:

Port	Wert	Funktion
0	0 - 255	User E/A (auch noch mit 65529 möglich) Bit3 Anwender Ausgang 1 Bit4 Anwender Ausgang 2 Bit5 Ausgang Bremse Bit6 Anwender Ausgang 3
1	0	Haube darf nicht geöffnet werden
	1	Haube darf geöffnet werden
2	0	Spindel ausschalten
	1	Spindel einschalten
3	0	Motorströme ausschalten
	1	Motorströme einschalten
4	0 - 255	analoger Ausgang 0 – 10V
5	0 - 255	User E/A I2C-Porterweiterung (I2C-Adresse=0)
6	0 - 255	User E/A Bit4 Anwender Ausgang 5 Bit5 Anwender Ausgang 6 Bit6 Anwender Ausgang 7 Bit7 Anwender Ausgang 8

Beschränkung: Ein Überschreiben der Portwerte erfolgt nur, wenn die Steuerung mit dem Software-Handshake "0" antwortet. Der Befehl kann nur verwendet werden, wenn keine Bewegung abgearbeitet wird. Die reservierten Bits des Port 0 können vom Anwender nicht überschrieben werden.

2.2.15. Testmodus Ein-/Ausschalten

Befehl: Testmodus Ein-/Ausschalten

Zweck: Durch Verwenden des Befehles kann der Testmodus gezielt aus- und eingeschaltet werden.

Aufbau: @<GN>T<Status><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
T	= Befehl Testmodus Ein-/Ausschalten
<Status>	= 0 --> Ausschalten, 1 --> Einschalten
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0T1, @0T0

Erläuterung: Die Steuerung wird mit dem Dateneröffnungsteil "@0" auf einen neuen Befehl vorbereitet. "T1" schaltet den Testmodus ein "T0" schaltet den Testmodus aus. Die Steuerung meldet sich nach erfolgter Ausführung mit dem Handshake-Charakter ("0"). Im Testmodus behandelt die Steuerung die Referenzfahrt und die Endschalter anders als im normalen Betrieb. Wenn im Testmodus ein Befehl Referenzfahrt empfangen wird, führt die Steuerung keine Referenzfahrt im eigentlichen Sinne aus sondern setzt den aktuellen Punkt als Referenzpunkt. Die Endschalter werden weiterhin überwacht können aber überfahren werden. Dies ist sehr nützlich wenn eine Achse nach dem Einschalten der Anlage in einem Endschalter steht und freigefahren werden muss.

Beschränkung: Der Befehl kann nur verwendet werden, wenn keine Bewegung abgearbeitet wird.

2.2.16. Abfragen der Statusdaten der Steuerung

Befehl: Statusdaten abfragen

Zweck: Abfrage wichtiger Statusdaten der Steuerung zur Darstellung des aktuellen Zustandes und zur Fehlersuche und -diagnose.

Aufbau: @<GN>H<CR>
 @<GN>h<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
H	= Befehl Statusdaten der Bedienelemente und Eingänge abfragen
h	= Befehl Statusdaten der Endschaltereingänge abfragen
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0H, @0h

Erläuterung: Die Steuerung wird mit dem Dateneröffnungsteil "@0" auf einen neuen Befehl vorbereitet. "H" oder "h" veranlasst die Steuerung Informationen über den aktuellen Status im Klartextformat zurückzusenden. Am Ende dieser Informationen antwortet die Steuerung mit dem Handshake-Charakter ("0"). Die Informationen werden im ASCII-Format bereits zeilenweise formatiert ausgegeben, so daß sie z.B. in einem Terminalfenster direkt auf dem Bildschirm eines Steuerrechners dargestellt werden können. Diese Informationen umfassen den Zustand der Endschalter oder der Bedienelemente einer Anlage.

Beschränkung: Für den Aufruf der Funktion muß ein genügend großer Empfangspuffer (min. 512 Byte) auf Seite des Steuerrechners zur Verfügung stehen, damit keine Informationen verlorengehen.

2.2.17. Abfragen der Versionsdaten der Steuerung

Befehl: Versionsdaten abfragen

Zweck: Abfrage wichtiger Versionsdaten der Steuerung.

Aufbau: @<GN>V<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
V	= Befehl Versionsdaten abfragen
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0V

Erläuterung: Die Steuerung wird mit dem Dateneröffnungsteil "@0" auf einen neuen Befehl vorbereitet. "V" veranlasst die Steuerung Informationen über Version der Steuerung im Klartextformat zurückzusenden. Am Ende dieser Informationen antwortet die Steuerung mit dem Handshake-Charakter ("0"). Die Informationen werden im ASCII-Format bereits zeilenweise formatiert ausgegeben, so daß sie z.B. in einem Terminalfenster direkt auf dem Bildschirm eines Steuerrechners dargestellt werden können.

Beschränkung: Für den Aufruf der Funktion muß ein genügend großer Empfangspuffer (min. 512 Byte) auf Seite des Steuerrechners zur Verfügung stehen, damit keine Informationen verlorengehen.

2.2.18. Initialisierung von Parametern

Befehl: Parameter initialisieren

Zweck: Initialisierung von Achs- und Referenzrichtungen.

Aufbau: @<GN>I<Code><Wert><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
I	= Befehl Initialisierung
<Code>	= ASCII-Zeichen zu Unterscheidung verschiedener Parameter s.u.
<Wert>	= neuer Wert für den Parameter
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0ID3, @0IR1, @0IS1

Erläuterung: Die Steuerung wird mit dem Dateneröffnungsteil "@0" auf einen neuen Befehl vorbereitet. "I" teilt der Steuerung mit, daß eine Initialisierung vorgenommen werden soll. Anschließend folgen eine Kennung für den Parameter sowie der neue Wert und Carriage Return als Befehlsende. Folgende Parameter können inialisiert werden:

Code	Wert	Funktion
D	1	Richtung X-Achse negiert
	2	Richtung Y-Achse negiert
	4	Richtung Z-Achse negiert
	8	Richtung A-Achse negiert
		(durch Addition der Werte sind alle Kombinationen möglich)
R	1	Referenzrichtung X-Achse negiert
	2	Referenzrichtung Y-Achse negiert
	4	Referenzrichtung Z-Achse negiert
	8	Referenzrichtung A-Achse negiert
		(durch Addition der Werte sind alle Kombinationen möglich)
L	1	X-Achse ist Drehachse
	2	Y-Achse ist Drehachse
	4	Z-Achse ist Drehachse
	8	A-Achse ist Drehachse
		(durch Addition der Werte sind alle Kombinationen möglich)
S	0-255	Setzen des internen Hardwarestatusbytes
		Bitbelegung: Bit0: Enable/Disable Haubenöffnung
		Bit1-7: nicht benutzt

Beschränkung: Das interne Hardwarestatusbyte sollte vom Anwender nicht benutzt werden und ist hier nur der Vollständigkeit halber aufgeführt. Die Benutzung bleibt den Softwareprodukten der Fa. isel-automation KG vorbehalten.

2.2.19. Diagnose

Befehl: Diagnose

Zweck: Abfragen von Diagnosedaten der Steuerung.

Aufbau: @<GN>D<Code1><Code2><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
D	= Befehl Diagnose
<Code1>	= ASCII-Zeichen zu Unterscheidung verschiedener Parameter s.u.
<Code2>	= ASCII-Zeichen zu Unterscheidung verschiedener Parameter s.u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0DRp, @0DRn, @0DS0

Erläuterung: Die Steuerung wird mit dem Dateneröffnungsteil "@0" auf einen neuen Befehl vorbereitet. "D" teilt der Steuerung mit, daß eine Diagnose vorgenommen werden soll. Anschließend folgen zwei Kennungen für den Parameter und Carriage Return als Befehlsende. Die Steuerung antwortet mit dem Software-Handshake "0" gefolgt von zwei Zeichen, die einen Hexadezimalwert angeben, der dem aktuellen Wert des Parameters entspricht. Folgende Parameter können abgefragt werden:

Code1	Code2	Funktion
R	p	positive Endschalter abfragen
R	n	negative Endschalter abfragen
P	0	Eingangsport abfragen
P	1	Endschalterport abfragen
P	2	UserEingangsport abfragen
O	0	Ausgangsport abfragen
O	1	UserAusgangsport abfragen
S	0	Hardwarestatuseingangsport abfragen
H	0	Hardwarestatusausgangsport abfragen
S	L	Achskonfiguration abfragen

Bitbelegung Endschalter:	Bit0:	X-Achse
	Bit1:	Y-Achse
	Bit2:	Z-Achse
	Bit3:	A-Achse

Bitbelegung Eingangsport:	Bit0:	Anwender Eingang 1
	Bit1:	Eingang Stoptaste
	Bit2:	Anwender Eingang 2
	Bit3:	Eingang PowerOK
	Bit4:	Eingang Notausschalter
	Bit5:	Eingang Eintaster
	Bit6:	Eingang Haubenschalter
	Bit7:	Eingang Schlüsselschalter

Bitbelegung Endschalterport:	Bit0:	positiver Endschalter A-Achse
	Bit1:	negativer Endschalter A-Achse
	Bit2:	positiver Endschalter Z-Achse
	Bit3:	negativer Endschalter Z-Achse
	Bit4:	positiver Endschalter Y-Achse
	Bit5:	negativer Endschalter Y-Achse
	Bit6:	positiver Endschalter X-Achse
	Bit7:	negativer Endschalter X-Achse

Bitbelegung UserEingangsport:

Bit0: Anwender Eingang 3
Bit1: Anwender Eingang 4
Bit2: Anwender Eingang 5
Bit3: Eingang Starttaster

Bitbelegung Ausgangsport:

Bit0: Ausgang Enable Endstufen
Bit1: Ausgang SoftwareOK
Bit2: Ausgang Spindel
Bit3: Anwender Ausgang 1
Bit4: Anwender Ausgang 2
Bit5: Ausgang Bremse
Bit6: Anwender Ausgang 3
Bit7: Ausgang Enable Haube

Bitbelegung UserAusgangsport:

Bit4: Anwender Ausgang 5
Bit5: Anwender Ausgang 6
Bit6: Anwender Ausgang 7
Bit7: Anwender Ausgang 8

Bitbelegung Hardwarestatuseingangsport:

Bit0: Status Spindel
Bit1: Status Starttaste
Bit2: Status Stoptaste
Bit3: Status NotAus
Bit4: Status Schlüsselschalter
Bit5: Status Haube
Bit6: nicht benutzt
Bit7: Status Endstufen/Stromversorgung

Bitbelegung Hardwarestatusausgangsport:

Bit0: Enable/Disable Haubenöffnung
Bit1-7: nicht benutzt

Bitbelegung Achskonfiguration:

Bit0: X-Achse 1=enabled, 0=disabled
Bit1: Y-Achse 1=enabled, 0=disabled
Bit2: Z-Achse 1=enabled, 0=disabled
Bit3: A-Achse 1=enabled, 0=disabled
Bit4: X-Achse 1=Drehachse, 0=Linear
Bit5: Y-Achse 1=Drehachse, 0=Linear
Bit6: Z-Achse 1=Drehachse, 0=Linear
Bit7: A-Achse 1=Drehachse, 0=Linear

Beschränkung: Die Diagnosefunktionen sollten vom Anwender nicht benutzt werden und sind hier nur der Vollständigkeit halber aufgeführt. Die Benutzung bleibt den Softwareprodukten der Fa. isel-automation KG vorbehalten.

2.2.20. Selbsttest durchführen

Befehl: Selbsttest

Zweck: Diese Anweisung löst einen Selbsttest der Steuerung aus. Diese Test umfaßt die Bewegung der Achsen sowie den Schnittstellentest und die Ausgabe von Versionsinformationen.

Aufbau: @<GN>?<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
?	= Befehl Selbsttest
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0?

Erläuterung: Die Steuerung wird durch "@0" adressiert. "?" gibt an, daß ein Selbsttest durchgeführt werden soll. Der Befehl wird mit Carriage Return abgeschlossen. Die Steuerung gibt anschließend Versionsinformationen aus, testet die Bewegung der Motoren und führt einen Schnittstellentest aus. Zum Testen der Schnittstelle gibt die Steuerung zunächst den ASCII-Zeichensatz aus. Wird über die Schnittstelle ein Zeichen empfangen so wechselt die Steuerung in den Echo-Mode und gibt alle empfangenen Zeichen an den Steuerrechner zurück.

Beschränkung: Für den Aufruf der Funktion muß ein genügend großer Empfangspuffer auf Seite des Steuerrechners zur Verfügung stehen, damit keine Informationen verlorengehen. Der Aufruf dieser Funktion ist nur innerhalb eines Terminalprogramms oder einer Terminalfunktion sinnvoll. Der Selbsttest kann nur durch Ausschalten der Steuerung oder Softwarereset (char(254)) beendet werden. Anschließend ist eine Neuinitialisierung der Anlage notwendig.

Der Selbsttest kann auch ausgelöst werden, wenn beim Einschalten der Steuerung die Starttaste betätigt und erst nach Beginn des Selbsttest losgelassen wird.

2.2.21. Kontroll und Steuercodes

Kontroll- und Steuercodes ermöglichen den direkten Eingriff in den Funktionsablauf der Steuerung über die serielle Schnittstelle. Dabei werden die jeweils gesendeten Kommandos ohne Verzögerung direkt in der Empfangsroutine der Steuerung ausgewertet und anschließend ausgeführt. Für folgende Funktionalitäten stehen spezielle Steuercodes zu Verfügung:

Funktion: Softwarestop char(253)

Eine Positionierbewegung im DNC-Modus (relativ oder absolut), kann durch einen Stopbefehl angehalten werden, ohne das Schrittverluste auftreten. Ein danach ausgeführter Startbefehl (durch Übergabe von "@0S", siehe 2.2.12. Starten einer angehaltenen Bewegung) beendet den unterbrochenen Funktionsablauf. Außerdem kann nach einem Stopbefehl mit Hilfe des Befehles "Positionsabfrage" die aktuelle erreichte Position rückgelesen werden. Diese Funktionalität kann auch durch Betätigen des Stoptasters erreicht werden. Wurde eine Bewegung erfolgreich angehalten erzeugt die Steuerung eine zusätzliche Rückmeldung "F".

Die Funktion wird durch Übergabe eines char(253) über die serielle RS232-Schnittstelle aufgerufen.

Funktion: Softwarereset char(254)

Die Steuerung unterbricht sofort alle Aktivitäten und führt intern einen Softwarereset durch. Anschließend muß die Anlage wieder neu initialisiert und eine Referenzfahrt durchgeführt werden.

Die Funktion wird durch Übergabe eines char(254) über die serielle RS232-Schnittstelle aufgerufen.

Funktion: Softwarebreak char(255)

Eine Positionierbewegung im DNC-Modus (relativ oder absolut), kann durch einen Breakbefehl beendet werden. Dies bedeutet, daß der Rest der Bewegung vergessen wird.

Die Funktion wird durch Übergabe eines char(255) über die serielle RS232-Schnittstelle aufgerufen.

siehe auch: 4. Die Fehlermeldungen der IMC4-M

2.2.22. CNC-Programm löschen

Befehl: CNC-Programm im FlashPROM löschen

Zweck: Diese Anweisung löscht ein im FlashPROM abgespeichertes CNC-Programm.

Aufbau: @<GN>k<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
k	= Befehl FlashPROM löschen
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0k

Erläuterung: Die Steuerung wird durch „@0“ adressiert. „k“ gibt an, dass der FlashPROM-Speicher der Steuerung gelöscht werden soll. Der Befehl wird mit Carriage Return abgeschlossen. Wurde der Befehl von der Steuerung richtig verstanden erfolgt das Löschen der FlashPROM-Speicher, dies dauert etwa 2 Sekunden. Die Steuerung bestätigt das Löschen anschließend mit einer Rückmeldung „0“.

2.2.23. Bewegung bis Portereignis

Befehl: Bewegung bis Portereignis

Zweck: Bewegung relativ linear bis Portereignis oder Bewegungsende

Aufbau: @<GN>Z<Portnr>,<Maske>,<Wert>,<G>,<Sx>,<Sy>,<Sz>,<Sa><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
Z	= Befehl Bewegung bis Port
<Portnr>	= Portadresse
<Maske>	= Maske zur Maskierung der Bits
<Wert>	= Sollwert als Bedingung für Bewegungsende
<G>	= Geschwindigkeit
<Sx>	= Schrittzahl X
<Sy>	= Schrittzahl Y
<Sz>	= Schrittzahl Z
<Sa>	= Schrittzahl A
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0Z0,4,4,600,0,0,3000,0

Erläuterung: Die Steuerung wird durch "@0" adressiert; Z gibt an, daß eine relative Bewegung erfolgen soll, die durch ein Portereignis beendet werden kann. Die Bewegung wird mit der Schrittfrequenz 600Hz ausgeführt. Die Bewegung wird beendet, wenn die vorgebenen Schrittzahlen erreicht sind, oder wenn am Eingansportport 0 das Bit2 auf 1 gesetzt wird.

Kommentar: Bitte beachten Sie, daß die Anzahl der Parameter abhängig von der Anzahl der initialisierten Achsen ist. Somit ergibt sich für verschiedene Achsenanzahlen folgender Befehlsaufbau:

X-Achse:	@<GN>Z<A>,<M>,<W>,<G>,<X><CR>
XY-Achse:	@<GN>Z<A>,<M>,<W>,<G>,<X>,<Y><CR>
XYZ-Achse:	@<GN>Z<A>,<M>,<W>,<G>,<X>,<Y>,<Z><CR>
XYZA-Achse:	@<GN>Z<A>,<M>,<W>,<G>,<X>,<Y>,<Z>,<A><CR>

2.3. Die Berechnung der Kreisparameter

2.3.1. Die Parameter für die Kreisinterpolation

Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle (s. 2.2.10, 2.2.11) ausgelöst. Der erste Befehl legt die Kreisrichtung fest, im Zweiten werden die Interpolationsparameter übergeben.

Kreisrichtung: @<GN>f<Richtung><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
f	= Befehl Kreisrichtung einstellen
<Richtung>	= 0 --> CW (Uhrzeigersinn), -1 --> CCW (gegen Uhrzeigersinn)
<CR>	= Carriage Return als Befehlsabschluß

Kreisinterpolation: @<GN>y,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry><CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
	= Bogenlänge in Schritten
<V>	= Geschwindigkeit
<D>	= Interpolationsparameter
<Xs>	= Startpunkt x
<Ys>	= Startpunkt y
<Rx>	= Richtung x
<Ry>	= Richtung y
<CR>	= Carriage Return als Befehlsabschluß

Steuerungsintern wird ein speziell angepasster Differenzalgorithmus nach Bresenham benutzt um Kreisbögen zu erzeugen. Diese Art Algorithmus wird sehr oft in Mikroprozessoranwendungen benutzt, da man hier mit geringem Rechenaufwand hohe Ausführungsgeschwindigkeiten erreicht. Hier soll nun die Bedeutung und Berechnung der Parameter für die Kreisinterpolation erläutert werden. Ein entsprechendes Beispiel finden Sie unter Punkt 2.3.6. dieser Beschreibung.

2.3.2. Die Berechnung der Bogenlänge

Die Bogenlänge gibt die Länge des Kreisbogens in Schritten zwischen dem Startpunkt und dem Endpunkt des Bogens an und wird steuerungsintern als Laufvariable für den Differenzalgorithmus benutzt. Die Berechnung der Bogenlänge in Schritten kann auf unterschiedlichen Wegen erfolgen und soll im Folgenden erläutert werden.

- Einfache Näherungsformel

Für einfache Kreisanwendungen, die z.B. nur Viertel-, Halb-, oder Vollkreise enthalten kann die Bogenlänge in Schritte mit folgender Formel berechnet werden:

B - Bogenlänge in Schritten
R - Radius des Bogens in Schritten
A - Anfangswinkel im Bogenmaß
E - Endwinkel im Bogenmaß

$$B = 4 * R * \frac{E - A}{\pi}$$

Das Ergebnis ist auf den nächsten ganzzahligen Wert zu runden. Um für die Abarbeitung auf der Steuerung die Ungenauigkeit zu eliminieren sollte die nächste Positionierung als absolute Bewegung programmiert werden.

- Berechnung mittels Softwareroutine

Eine genaue Berechnung der Bogenlänge in Schritten kann mittels einfacher Softwareroutinen erreicht werden.

2.3.3. Die Berechnung des Interpolationsparameters

Der Interpolationsparameter dient der Steuerung als Startwert für das Differenzregister des von der Steuerung verwendeten Algorithmus zur Kreiserzeugung. Die Berechnung des Parameters erfolgt auf PC-Seite mittels entsprechender Softwareroutinen. Dies entlastet die Steuerung von unnötigem Rechenaufwand und erhöht somit die Abarbeitungsgeschwindigkeit.

2.3.4. Startpunkt des Kreisbogens

Der Startpunkt des Kreisbogens stellt den Abstand in X und Y in Schritten vom Kreismittelpunkt in relativen Koordinaten dar (d.h. der Kreismittelpunkt wird für die Berechnung als gedachter Nullpunkt angenommen). Die Berechnung kann mittels der entsprechenden Kreisfunktionen erfolgen.

Xs - X-Koordinate des Startpunktes relativ zum Mittelpunkt
 Ys - Y Koordinate des Startpunktes relativ zum Mittelpunkt
 R - Radius in Schritten
 A - Anfangswinkel im Bogenmaß

$$Xs = R * \cos(A)$$

$$Ys = R * \sin(A)$$

2.3.5. Richtungen im Startpunkt der Kreisinterpolation

Zur Ausführung des Interpolationsalgorithmus benötigt die Steuerung eine Information darüber, in welchem Quadranten der Bogen beginnt und welche Vorzeichen steuerungsintern für bestimmte Berechnungen benutzt werden sollen. Diese Informationen werden der Steuerung in Form der Parameter Rx und Ry zur Verfügung gestellt. Dabei gelten folgende Festlegungen:

Kreisbögen entgegen dem Uhrzeigersinn (CCW)

Kreisbögen im Uhrzeigersinn (CW)

90 Grad	
II. Quadrant	I. Quadrant
Rx = -1 Ry = -1	Rx = -1 Ry = +1
180 Grad	0 Grad
Rx = +1 Ry = -1	Rx = +1 Ry = +1
III. Quadrant	IV. Quadrant
270 Grad	

90 Grad	
II. Quadrant	I. Quadrant
Rx = +1 Ry = +1	Rx = +1 Ry = -1
180 Grad	0 Grad
Rx = -1 Ry = +1	Rx = -1 Ry = -1
III. Quadrant	IV. Quadrant
270 Grad	

2.3.6. Berechnungsbeispiel zur Kreisinterpolation

Zur Vertiefung der Berechnung der Parameter des Kreisbefehls soll hier ein kurzes Beispiel berechnet werden.

Es soll ein Kreisbogen entgegen dem Uhrzeigersinn mit dem Radius von 200 Schritten mit einer Geschwindigkeit von 1500 Schritten/s verfahren werden. Der Anfangswinkel sei 135 Grad, der Endwinkel 225 Grad. Beachten Sie bitte, daß für die Berechnung alle Wegangaben in Schritten und alle Winkelangaben im Bogenmaß zur Verfügung stehen müssen.

geg:	Radius	R = 200	ges:	Bogenlänge	B
	Startwinkel	A = $135 \cdot \pi / 180 = 2.3562$		StartpunktX	Xs
	Endwinkel	E = $225 \cdot \pi / 180 = 3.9267$		StartpunktY	Ys
	Speed	V = 1500		RichtungX	Rx
	Richtung	CCW		RichtungY	Ry
				InterpolationsparameterD	

Bogenlänge B (siehe 2.3.2.):

$$\begin{aligned}
 B &= 4 \cdot R \cdot (E - A) / \pi \\
 B &= 4 \cdot 200 \cdot (3.9267 - 2.3562) / \pi \\
 B &= 4 \cdot 200 \cdot 0.4999 = 399.9245 \\
 \underline{B} &= \underline{400}
 \end{aligned}$$

Startpunkt Xs und Ys (siehe 2.3.4):

$$\begin{aligned}
 Xs &= R \cdot \cos(A) = 200 \cdot \cos(2.3562) = -141.4221 \\
 \underline{Xs} &= \underline{-141} \\
 Ys &= R \cdot \sin(A) = 200 \cdot \sin(2.3562) = 141.4205 \\
 \underline{Ys} &= \underline{141}
 \end{aligned}$$

Richtung Rx und Ry (siehe 2.3.5):

$$\begin{aligned}
 &\text{Startwinkel 135 Grad Drehrichtung CCW} \\
 \underline{Rx} &= \underline{-1} \quad \underline{Ry} = \underline{-1}
 \end{aligned}$$

Interpolationsparameter D (siehe 2.3.3.):

$$D = (Rx \cdot Ry \cdot R + Rx \cdot Ry \cdot \text{Summe}(R-1) - Rx \cdot \text{Summe}(Xs + (Rx - Ry)/2) + Ry \cdot \text{Summe}(Ys + (Rx + Ry)/2)) / 2$$

$$\text{Summe}(R-1) = \text{Summe}(199) = 199 \cdot (199+1) = \underline{39800}$$

$$\text{Summe}(Xs + (Rx - Ry)/2) = \text{Summe}(-141 + (-1 - (-1))/2) = \text{Summe}(-141) = 141 \cdot (-141+1) = \underline{-19740}$$

$$\text{Summe}(Ys + (Rx + Ry)/2) = \text{Summe}(141 + (-1 - (-1))/2) = \text{Summe}(141) = 141 \cdot (141+1) = \underline{20022}$$

$$\begin{aligned}
 D &= ((-1) \cdot (-1) \cdot 200 + (-1) \cdot (-1) \cdot 39800 - (-1) \cdot (-19740) + (-1) \cdot 20022) / 2 \\
 D &= (200 + 39800 - 19740 - 20022) / 2 = \underline{119}
 \end{aligned}$$

Die Befehle würden lauten:

```

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

```

3. Der CNC-Mode und seine Befehle

3.1. Befehlsaufbau der CNC-Befehle

Im CNC-Modus betrieben, speichert die Steuerung alle übersendeten Befehle im internen Datenspeicher. Zur Aktivierung ist nach der Standardinitialisierung der Befehl "CNC-Datenfeld speichern" zu übertragen. Anschließend wird das Datenfeld übergeben und mit dem Befehl "Datenfeldende" abgeschlossen.

Das Programm kann nun ohne weitere Kommunikation mit dem Steuerrechner durch einen externen Start-Befehl (Betätigen der Starttaste) aktiviert werden.

Als Speichermedium kommen auf der IMC4-M FlashPROM's (nichtflüchtige elektrisch schreib- und löschbare Speicher) zum Einsatz. Diese Speicher werden, ähnlich einem EPROM, im System durch bestimmte Programmierzyklen mit den entsprechenden Informationen beschrieben. Das Löschen entspricht dem Beschreiben mit einem Defaultwert. Vor einem Wiederbeschreiben der Chips müssen diese prinzipiell gelöscht werden. Dazu kann der Befehl "CNC Programm Löschen" ("@0k") verwendet werden.

Nach dem Löschen können die Speicher mit einem neuen Programm überschrieben werden. Ist auf den Speichern bereits ein Programm oder ein Teil eines Programmes abgelegt führt der Befehl "CNC-Datenfeld speichern" zu einer Fehlermeldung.

Im Folgenden werden die speicherbaren Befehle der Steuerung IMC4-M aufgelistet und kurz erläutert. Eine Detaillierung kann für einige Befehle unter dem entsprechenden Befehl des DNC-Modus nachgeschlagen werden, da die Bedeutung und Anzahl der Parameter denen des DNC-Modus entsprechen.

Ist während der Übertragung und Speicherung eines CNC-Datenfeldes ein Fehler aufgetreten wird das bis dahin abgespeicherte CNC-Programm als ungültig markiert und kann nicht abgearbeitet werden. Der Fehler im Programm muß dann entsprechend beseitigt und das FlashPROM gelöscht werden, ehe das Datenfeld erneut zu Abspeicherung übertragen werden kann.

3.2. Die Befehle der IMC4-M im CNC-Mode

3.2.1. CNC-Datenfeld speichern

Befehl: CNC-Datenfeld speichern

Zweck: Diese Anweisung dient als Initialisierung für die Übertragung von speicherbaren Befehlen und ist zu Beginn des CNC-Modus zwingend erforderlich.

Aufbau: @<GN>i<CR>

@	= Dateneröffnungszeichen
<GN>	= Gerätenummer, Standard=0
i	= Befehl CNC-Datenfeld speichern
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: @0i

Erläuterung: Die Steuerung wird durch "@0" adressiert. "i" gibt an, daß ein CNC-Datenfeld gespeichert werden soll. Der Befehl wird mit Carriage Return abgeschlossen. Die Steuerung akzeptiert anschließend bis zum Befehl "Datenfeldende" oder bis zum Auftreten eines Fehlers nur noch CNC-Befehle. Der Befehl wird mit einer entsprechenden Rückmeldung quittiert. Alle nachfolgenden speicherbaren Befehle werden im FlashPROM abgespeichert.

Beschränkung: Der Befehl kann nur verwendet werden, wenn die Steuerung vorher initialisiert wurde und keine Bewegung abgearbeitet wird. Ist bereits ein Programm auf den Speichern abgelegt, so führt dies zu einer Fehlermeldung.

3.2.2. Referenzfahrt im CNC-Mode

Befehl: Referenzfahrt

Zweck: Die Steuerung speichert eine Bewegung aller angegebenen Achsen an ihre Nullpunkte (Referenzpunkte). Die Referenzpunkte der Achsen sind bei isel-Systemen immer in einer sinnvollen Defaultanordnung festgelegt, können aber durch entsprechende Initialisierungsbefehle angepaßt werden.

Aufbau: 7<Achsen><CR>

7 = Befehlscode Referenzfahrt
<Achsen> = Achsenangabe, s. u.
<CR> = Carriage Return als Befehlsabschluß

Anwendung: 77, 78

Erläuterung: "7" gibt an, daß eine Referenzfahrt ausgeführt werden soll. Der anschließende Zahlenwert definiert die Achsen, die eine Referenzfahrt durchführen sollen. Dabei wird intern jede Achse durch ein Bit eines Binärwertes repräsentiert und somit ergeben sich folgende Werte:

1 --> X Achse
2 --> Y Achse
3 --> X+Y Achse
4 --> Z Achse
5 --> X+Z Achse
6 --> Y+Z Achse
7 --> X+Y+Z Achse
8 --> A Achse

Die Reihenfolge der Ausführung ist dabei wie folgt festgelegt:
Z-Achse --> Y-Achse --> X-Achse --> A-Achse

Nach erfolgter Referenzfahrt wird der nächste CNC-Befehl aus dem Speicher gelesen und abgearbeitet.

Beschränkung: Der Befehl beschränkt sich auf die initialisierte Achsenkonfiguration. Bei falscher Achsangabe erfolgt die Fehlerrückmeldung "3". Befindet sich die Steuerung im 3D-Modus, schaltet der Befehl wieder in die 2,5 dim. Ausführung zurück.

ACHTUNG: Die A-Achse muß immer separat referenziert werden.

ACHTUNG: Bei nicht angeschlossenem Referenzschalter, wird die entsprechende Achse permanent angesteuert. Durch Betätigen des Stop-Tasters besteht jedoch die Möglichkeit die Referenzfahrt zu beenden.

3.2.3. Relative Bewegung im CNC-Mode

Befehl: Bewegung relativ

Zweck: Die Steuerung speichert entsprechend der übergebenen Schrittzahl und Schrittgeschwindigkeit eine relative Bewegung.

Aufbau: 0<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>
0<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>

0 = Befehlscode Bewegung relativ
<Sx> = Schrittzahl x
<Gx> = Geschwindigkeit
<Sy> = Schrittzahl y
<Gy> = Geschwindigkeit
<Sz>, <Sz1> = Schrittzahl z
<Sz2> = Schrittzahl z, 2. Bewegung bei 2.5d, 3 Achsen
<Gz>, <Gz1> = Geschwindigkeit
<Gz2> = Geschwindigkeit z, 2. Bewegung bei 2.5d, 3 Achsen
<Sa> = Schrittzahl a, bei 4 Achsen
<Ga> = Geschwindigkeit, bei 4 Achsen
<CR> = Carriage Return als Befehlsabschluß

Anwendung: 05000,900 (nur x-Achse)
050,900,20,9000 (x und y-Achse)
030,800,10,900,4,90,-4,30 (x,y und z-Achse, bei 3 Achsen)
030,800,10,900,4,90,-4,30 (x,y,z und a-Achse, bei 4 Achsen)

Erläuterung: "0" gibt an, daß eine relative Bewegung erfolgen soll. Die Steuerung erwartet nun für jede Achse ein Zahlenpaar bestehend aus Schrittzahl und Geschwindigkeit. Die Angabe der Entfernungen erfolgt im Relativmaß, d.h. bezogen auf die letzte Position. Die Anzahl der Angaben muß mit der Achsenzahl übereinstimmen, d.h. ein Parameterpaar bei x-Betrieb, zwei Parameterpaare bei xy-Betrieb, vier Parameterpaare für xyz-Betrieb und vier Parameterpaare für xyza-Betrieb. Die einzelnen Zahlen müssen durch Kommata getrennt werden. Für die z-Achse werden bei 3 Achsen und 2,5d zwei Zahlenpaare erwartet, da für Bearbeitungsanwendungen sehr häufig die Situation "Fahren, Werkzeug absenken und anschließend anheben" vorkommt. Im 2.5d-Interpolationsbetrieb werden zuerst die Bewegungen der x- und y-Achse verfahren (linear interpoliert), anschließend wird die z-Achse zuerst um die in z1-angegebenen und dann um die in z2-angegebenen Werte verfahren. Besteht die Notwendigkeit nur eine Achse zu bewegen, so sind dennoch für alle initialisierten Achsen Werte zu übertragen. Dabei ist für die Schrittzahlen der nicht bewegten Achsen 0 anzugeben. Die Steuerung meldet sich nach erfolgter Speicherung mit dem Handshake-Charakter ("0").

Beschränkung: Die Steuerung prüft nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verläßt.
Im 2.5d-Interpolationsbetrieb wird die Geschwindigkeitsangabe der Achse mit dem längsten Weg als Bahngeschwindigkeit übernommen und die Geschwindigkeit der anderen Achsen entsprechend dem Wegeverhältnis angepaßt. Dem gegenüber wird im 3d-Interpolationsbetrieb die Geschwindigkeitsangabe der x-Achse als Vorgabewert für die Bahngeschwindigkeit herangezogen.

3.2.4. Absolute Bewegung im CNC-Mode

Befehl: Bewegung zur absoluten Position

Zweck: Die Steuerung speichert entsprechend den angegebenen Geschwindigkeiten und Positionen eine absolute Bewegung.

Aufbau: `m<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2><CR>`
`m<Sx>,<Gx>,<Sy>,<Gy>,<Sz>,<Gz>,<Sa>,<Ga><CR>`

<code>m</code>	= Befehlscode Bewegung absolut
<code><Sx></code>	= Position x
<code><Gx></code>	= Geschwindigkeit
<code><Sy></code>	= Position y
<code><Gy></code>	= Geschwindigkeit
<code><Sz>,<Sz1></code>	= Position z
<code><Sz2></code>	= Position z, 2. Bewegung immer = 0
<code><Gz>,<Gz1></code>	= Geschwindigkeit
<code><Gz2></code>	= Geschwindigkeit
<code><Sa></code>	= Position a, bei 4 Achsen
<code><Ga></code>	= Geschwindigkeit, bei 4 Achsen
<code><CR></code>	= Carriage Return als Befehlsabschluß

Anwendung:	<code>m5000,900</code>	(nur x-Achse)
	<code>m50,900,20,9000</code>	(x und y-Achse)
	<code>m30,800,10,900,4,90,0,30</code>	(x,y und z-Achse, bei 3 Achsen)
	<code>m30,800,10,900,4,90,4,30</code>	(x,y,z und a-Achse, bei 4 Achsen)

Erläuterung: "m" gibt an, daß eine Absolut-Position folgt. Aus Kompatibilitätsgründen zum relativen Positionierbefehl werden auch hier bei 3 Achsen für die z-Achse zwei Zahlenpaare erwartet. Die zweite Positionsangabe der z-Position muß dann jedoch Null sein und wird ignoriert. Die Steuerung meldet sich nach erfolgter Speicherung mit dem Handshake-Charakter.

Beschränkung: Die Steuerung prüft nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verläßt.

3.2.5. Nullpunkt setzen im CNC-Mode

Befehl: Nullpunkt am aktuellen Punkt setzen

Zweck: Die Steuerung speichert einen Befehl, um die momentane Position während der Abarbeitung des CNC-Programms als virtuellen Nullpunkt für die angegebene(n) Achse(n) zu setzen. Die nachfolgenden "Verfahre absolut"-Anweisungen beziehen sich dann auf diesen virtuellen Nullpunkt.

Aufbau: n<Achsen><CR>

n	= Befehlscode Nullpunkt setzen
<Achsen>	= Achsenangabe, s. u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: n7, n1, n8

Erläuterung: "n" gibt an, daß eine Nullpunktverschiebung vorgenommen werden soll. Nach dem Befehlscode werden der Steuerung die Achsen mitgeteilt, für die eine Nullpunktverschiebung durchgeführt werden soll. Dabei wird intern jede Achse durch ein Bit eines Binärwertes repräsentiert und somit ergeben sich folgende Werte:

1 -->	X Achse
2 -->	Y Achse
3 -->	X+Y Achse
4 -->	Z Achse
5 -->	X+Z Achse
6 -->	Y+Z Achse
7 -->	X+Y+Z Achse
8 -->	A Achse

Die Steuerung meldet sich nach erfolgter Speicherung mit einer Rückmeldung.

Beschränkung: Der virtuelle Nullpunkt hat nur für den Befehl "Verfahre absolut" eine Bedeutung. Relativpositionierung wird vom virtuellen Nullpunkt nicht beeinflusst, da hier ein relativer Fahrvektor angegeben wird.

ACHTUNG: Die Nullpunktverschiebung für die A-Achse muß immer separat durchgeführt werden.

3.2.6. 3d Interpolation Ein-/Ausschalten im CNC-Mode

Befehl: 3D-Linearinterpolation Ein-/Ausschalten

Zweck: Die Steuerung speichert den Befehl, um die 2.5d Interpolation des Betriebssystems auf eine 3-dimensionale Interpolation zu erweitern. Durch Verwenden des Befehles kann diese Interpolation gezielt aus- und eingeschaltet werden.

Aufbau: z<Status><CR>

z	= Befehlscode 3d Interpolation
<Status>	= 0 --> Ausschalten, 1 --> Einschalten
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: z1, z0

Erläuterung: "z1" ändert die Interpolation von 2d- auf 3d-Betrieb. Die Anweisung wirkt modal, d. h. alle relativen und absoluten Bewegungen werden dreidimensional ausgeführt. Die Angabe von z2-Parametern bei 3 Achsen in diesen Verfahrbewegungen wird ignoriert. Die Geschwindigkeitsangabe der Interpolation muß bei der x-Angabe erfolgen. Bei 4 Achsen wird die 4.Achse entsprechend nachgeführt.

3.2.7. Ebenenwahl für Kreisinterpolation im CNC-Mode

Befehl: Ebenenwahl

Zweck: Speichern der Interpolationsebene für die Kreisinterpolation. Kreise sind nur innerhalb einer Ebene definiert. Die Defaultebene für die Kreisinterpolation ist die XY-Ebene. Durch den "Ebenenwahl"-Befehl besteht hier jedoch die Möglichkeit jede andere Ebenenkonfiguration als Kreisebene zu definieren.

Aufbau: e<Ebene><CR>

e	= Befehlscode Kreisebene setzen
<Ebene>	= Ebenenangabe, s. u.
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: e1, e0

Erläuterung: "e" gibt an, daß die Ebene für die Kreisinterpolation eingestellt werden soll. Der anschließende Zahlenwert definiert die Ebene in folgender Weise:

0 -->	XY Ebene
1 -->	XZ Ebene
2 -->	YZ Ebene

Beschränkung: Dieser Befehl ist modal wirksam, d.h. eine Ebenenwahl für die Kreisinterpolation bleibt erhalten bis sie von einer erneuten Ebenenwahl überschrieben wird.

3.2.8. Einstellen der Kreisrichtung für die Kreisinterpolation im CNC-Mode

Befehl: Kreisrichtung einstellen

Zweck: Speichern der Kreisrichtung für die Kreisinterpolation. Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle programmiert. Der erste Befehl legt die Kreisrichtung fest, im Zweiten (s. 3.2.9.) werden die Interpolationsparameter übergeben.

Aufbau: f<Richtung><CR>

f	= Befehlscode Kreisrichtung einstellen
<Richtung>	= 0 --> CW (Uhrzeigersinn), -1 --> CCW (gegen Uhrzeigersinn)
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: f-1, f0

Erläuterung: "f" gibt an, daß die Richtung für die Kreisinterpolation eingestellt werden soll. Der anschließende Zahlenwert definiert die Richtung in folgender Weise:

0	--> CW (Kreisinterpolation im Uhrzeigersinn)
-1	--> CCW (Kreisinterpolation entgegen dem Uhrzeigersinn)

Beschränkung: Die Richtung für die Kreisinterpolation ist prinzipiell vor jeder Kreisbewegung zu programmieren.

3.2.9. Kreisinterpolation im CNC-Mode

Befehl: Kreisinterpolation

Zweck: Speichern von Bewegungsbefehlen für Kreise und Kreisbögen mit konstanter Bahngeschwindigkeit. Die Kreisinterpolation wird durch zwei aufeinanderfolgende Befehle ausgelöst. Der erste Befehl legt die Kreisrichtung fest (s. 3.2.8.), im Zweiten werden die Interpolationsparameter übergeben.

Aufbau: `y<B>,<V>,<D>,<Xs>,<Ys>,<Rx>,<Ry><CR>`

y	= Befehlscode Kreisinterpolation
	= Bogenlänge in Schritten
<V>	= Geschwindigkeit
<D>	= Interpolationsparameter
<Xs>	= Startpunkt x
<Ys>	= Startpunkt y
<Rx>	= Richtung x
<Ry>	= Richtung y
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: `y400,1500,119,-141,141,-1,-1`

Erläuterung: "y" gibt an, daß eine Kreisinterpolation gespeichert werden soll. Die Bogenlänge gibt die Länge des Bogens in Schritten zwischen dem Start- und Endpunkt der Kreisinterpolation an. Als Geschwindigkeitsangaben sind alle ganzzahligen Werte innerhalb des gültigen Wertebereiches für Geschwindigkeiten zulässig. Der Interpolationsparameter muß übergeben werden, da die Steuerung aufgrund ihrer Speicherkapazität nicht in der Lage ist diesen Parameter selbst zu berechnen. Die Parameter Xs und Ys geben den Startpunkt des Bogens relativ zum Kreismittelpunkt an. Rx und Ry geben an, in welchem Quadranten des Kreises die Interpolation startet. Die Steuerung meldet sich nach erfolgter Ausführung mit dem Handshake-Charakter ("0").

ACHTUNG: Zur Berechnung der Parameter lesen Sie bitte das Kapitel "Berechnung der Parameter für die Kreisinterpolation".

Beschränkung: Die Steuerung prüft nicht, ob die Bewegung den zulässigen Bereich der angeschlossenen Mechanik verläßt.

3.2.10. Schleifen, Verzweigungen im CNC-Mode

Befehl: Schleife, Verzweigung

Zweck: Speichern von Schleifen und Verzweigungen. Schleifen dienen dazu, gleichartige Bewegungsabläufe zusammenzufassen. Hierdurch wird der zur Verfügung stehende Speicherplatz der Steuerung besser genutzt. Durch Verzweigungen kann nach einer logischen Entscheidung zu einem bestimmten Satz innerhalb des Programms gesprungen werden.

Aufbau: 3<Anzahl>,<Offset><CR>

3	= Befehlscode Schleife, Verzweigung
<Anzahl>	= Schleifenzahl
Schleife:	$0 < \text{Schleifenzahl} < 32768$
Verzweigung:	immer 0
<Offset>	= Sprungziel
Schleife:	$-1 \geq \text{Sprungziel} \geq -32768$
Verzweigung:	$-32768 \leq \text{Sprungziel} \leq 32767$
<CR>	= Carriage Return als Befehlsabschluß

Anwendung:	3 25,-1	Wiederhole letzten Befehl 25 mal
	3 0,-5	Verzweige immer 5 Schritte zurück
	3 0,5	Überspringe die nächsten 4 Befehle
	3 6,-5	Wiederhole die letzten 5 Befehle 6 mal

Erläuterung: Trifft die Steuerung innerhalb des CNC-Programmablaufes auf den Befehl "Schleife/Verzweigung" wird zunächst durch Prüfen der Schleifenzahl entschieden, ob es sich um einen Schleifen- oder um einen Verzweigungsbefehl handelt. Bei einem Schleifenbefehl, wird ein Schleifenzähler eingerichtet, vorbesetzt und der Befehlszähler um den angegebenen Offset korrigiert. Die Befehle bis zum Schleifenzähler werden nun jeweils wiederholt und der Schleifenzähler dekrementiert, bis dieser Null erreicht hat. Anschließend wird mit der Ausführung des ersten Befehls nach der Schleife fortgefahren. Schleifen können mit einer Schachtelungstiefe von 15 ineinander verschachtelt sein. Die notwendigen Zähler werden dann auf einem entsprechenden Schleifenstack verwaltet. Bei einer Verzweigung wird der Offset als relatives Sprungziel innerhalb des CNC-Programmes verstanden und der Befehlszähler entsprechend um den Offset korrigiert.

Beschränkung: Es darf nicht vor den Anfang oder hinter das Ende des Datenfeldes verzweigt werden. Vorwärtsschleifen sind unzulässig. Eine Schleife wiederholt immer die letzten n-Befehle. Es muß mindestens ein Befehl wiederholt werden. Schleifen dürfen geschachtelt sein, die maximale Schachtelungstiefe beträgt 15. Eine Schleife darf nicht durch eine Verzweigung verlassen werden.

3.2.11. Zeitverzögerungen im CNC-Mode

Befehl: Zeitverzögerung

Zweck: Speichern von Zeitverzögerungen.

Aufbau: 5<Zeit><CR>

5	= Befehlscode Zeitverzögerung
<Zeit>	= Zeit in 1/10 sec
<CR>	= Carriage Return als Befehlsabschluß

Anwendung: 3 50 Verzögerung 5 Sekunden

Erläuterung: Trifft die Steuerung innerhalb des CNC-Programmablaufes auf den Befehl "Zeitverzögerung". So erfolgt die Ausführung des nächste Befehls im CNC-Programm erst nach Ablauf der Verögerungszeit. Die Zeitangabe erfolgt dabei in 1/10 Sekunden.

Beschränkung: Eine Zeitverzögerung kann nicht durch Betätigen des Stop-Tasters der Steuerung abgebrochen werden.

3.2.12. Port setzen im CNC-Mode

Befehl: Ausgangsport setzen

Zweck: Definiertes Ein- / Ausschalten von vorhandenen Ausgangsports.

Aufbau: p<Portnr>,<Bitnr>,<Wert><CR>

p = Befehlscode Port setzen
 <Portnr> = Portnummer
 <Bitnr> = Bitnummer, 0 - 7 --> bitweise, 128 --> byteweise
 <Wert> = neuer Wert
 <CR> = Carriage Return als Befehlsabschluß

Anwendung: p0,128,1 Port 0, byteweise auf 1 setzen
 p0,0,1 Port 0 , Bit 0 auf 1 setzen

Erläuterung: "p" gibt an, daß der Wert eines Ausgabeports gesetzt werden soll. Anschließend wird die Portnummer, die Bitnummer und der neue Portwert getrennt durch Komma übermittelt und der Befehl mit Carriage Return abgeschlossen. Die Steuerung antwortet mit dem Software-Handshake "0", falls die Speicherung erfolgreich war, oder mit einer Fehlermeldung, falls falsche Portnummern und/oder Werte übergeben wurden. Für die Steuerung IMC4 sind folgende Ports mit entsprechenden Funktionalitäten definiert:

Port	Bit	Wert	Funktion
0	0 - 7	0 - 255	User E/A (auch noch mit 65529 möglich)
			Bit3 Anwender Ausgang 1
			Bit4 Anwender Ausgang 2
			Bit5 Ausgang Bremse
			Bit6 Anwender Ausgang 3
1	0	0	Haube darf nicht geöffnet werden
	0	1	Haube darf geöffnet werden
2	0	0	Spindel ausschalten
	0	1	Spindel einschalten
3	0	0	Motorströme ausschalten
	0	1	Motorströme einschalten
4	0 - 7	0 - 255	analoger Ausgang 0 – 10V
5	0 - 7	0 - 255	User E/A I2C-Porterweiterung (I2C-Adresse=0)
6	0 - 7	0 - 255	User E/A
			Bit4 Anwender Ausgang 5
			Bit5 Anwender Ausgang 6
			Bit6 Anwender Ausgang 7
			Bit7 Anwender Ausgang 8

Beschränkung: Das Setzen der Portausgänge wird innerhalb der Steuerung dem Programmablauf entsprechend durchgeführt. Somit ist ein Setzen bzw. Löschen von Ausgängen während einer Befehlsbearbeitung z.B. einer Positionierbewegung nicht möglich.

3.2.13. Port lesen und verzweigen im CNC-Mode

Befehl: Eingangsport lesen

Zweck: Eingangsport lesen und im Programmablauf verzweigen. Durch die Verzweigung kann nach einem logischen Vergleich zu einem bestimmten Satz innerhalb des Programms gesprungen werden.

Aufbau: o<Portnr>,<Bitnr>,<Wert>,<Offset><CR>

o = Befehlscode Port setzen
 <Portnr> = Portnummer
 <Bitnr> = Bitnummer, 0 - 7 --> bitweise, 128 --> byteweise
 <Wert> = Vergleichswert
 <Offset> = Sprungziel -32768 <= Sprungziel <= 32767
 <CR> = Carriage Return als Befehlsabschluß

Anwendung: o0,128,1,-1 warten bis Port 0 <> 1
 o0,0,1,-1 warten bis Port0, Bit0 = 0
 o0,0,1,3 wenn Port0, Bit0 == 1, Befehlszähler += 3

Erläuterung: "o" gibt an, daß der Wert eines Eingabeports gelesen und der Programmablauf entsprechend des Wertes angepaßt werden soll. Anschließend wird die Portnummer, die Bitnummer, der Vergleichswert und der Befehlsoffset getrennt durch Komma übermittelt und der Befehl mit Carriage Return abgeschlossen. Die Steuerung antwortet mit dem Software-Handshake "0", falls die Speicherung erfolgreich war, oder mit einer Fehlermeldung, falls falsche Portnummern und/oder Werte übergeben wurden. Während des Programmablaufs wird das entsprechende Port abgefragt und bit- oder byteweise logisch mit dem vorgegebenen Wert verglichen. Ist der logische Vergleich wahr, wird um den Offset verzweigt, sonst wird der nächste Befehl im Programmablauf abgearbeitet. Für die Steuerung IMC4 sind folgende Ports mit entsprechenden Funktionalitäten definiert:

Port	Bit	Zustand	Funktion
0	0 - 7	00 - FF	User E/A (auch noch mit 65531 möglich)
			Bit0 Eingang Stoptaster
			Bit1 Eingang Starttaster
			Bit2 Anwender Eingang 1
			Bit3 Eingang PowerOK
			Bit4 Eingang Notausschalter
			Bit5 Eingang Eintaster
			Bit6 Eingang Haubenschalter
			Bit7 Eingang Schlüsselschalter
1	0	00	Haube ist geöffnet
	0	01	Haube ist geschlossen
2	0	00	Spindel ist ausgeschaltet
	0	01	Spindel ist eingeschaltet
3	0	00	Motorströme sind ausgeschaltet
	0	01	Motorströme sind eingeschaltet
5	0 - 7	00 - FF	User E/A I2C-Porterweiterung (I2C-Adresse=0)
6	0 - 7	00 - 1F	User E/A
			Bit0 Anwender Eingang 1
			Bit1 Anwender Eingang 2
			Bit2 Anwender Eingang 3
			Bit3 Anwender Eingang 4
			Bit4 Anwender Eingang 5

Beschränkung: Das Abfragen der Porteingänge wird innerhalb der Steuerung dem Programmablauf entsprechend durchgeführt. Somit ist eine Abfrage von Eingängen während einer Befehlsbearbeitung z.B. einer Positionierbewegung nicht möglich.

3.2.14. Datenfeldende im CNC-Mode

Befehl: Datenfeldende

Zweck: Der Befehl kennzeichnet das Ende eines CNC-Datenfeldes und dient dem Abschluß der Datenübertragung und Speicherung von speicherbaren Befehlen.

Aufbau: 9<CR>

 9 = Befehlscode Datenfeldende
 <CR> = Carriage Return als Befehlsabschluß

Anwendung: 9

Erläuterung: "9" gibt an, daß das Ende des übertragenen CNC-Datenfeldes erreicht ist. Der Befehl wird mit Carriage Return abgeschlossen. Die Steuerung antwortet mit dem Software-Handshake "0", falls die Speicherung erfolgreich war, oder mit einer Fehlermeldung. Neben der Kennzeichnung des Datenfeldes als gültiges CNC-Programm werden Statusinformationen (z.B. die aktuelle Referenzgeschwindigkeit) im FlashProm abgelegt. Anschließend befindet sich die Steuerung wieder im DNC-Mode und akzeptiert die entsprechenden Befehle.

Beschränkung: Ein CNC-Datenfeld muß mit dem Befehl Datenfeldende abgeschlossen werden, sonst ist das abgespeicherte CNC-Programm nicht gültig und kann nicht abgearbeitet werden.

4. Die Fehlermeldungen der IMC4-M

Nach jedem übertragenen Befehl antwortet die Steuerung mit einer entsprechenden Rückmeldung. Diese Codes werden als ASCII-Zeichen übertragen und können somit einfach ausgewertet werden. Anhand des übermittelten Zeichens können Fehlerquellen und -ursachen erkannt werden. Die einzelnen Fehlercodes sind im Folgenden beschrieben.

Code	Beschreibung
0	Handshake-Charakter - Kein Fehler, der Befehl wurde korrekt ausgeführt. - Der nächste Befehl kann übermittelt werden.
1	Fehler in übergebener Zahl - Die Steuerung hat eine Zahlenangabe empfangen, die nicht korrekt interpretiert werden konnte. - Der übergebene Zahlenwert ist außerhalb des zulässigen Bereiches oder der übergebene Zahlenwert enthält unzulässige Zeichen.
2	Endschalterfehler - Durch die Verfahrensbewegung wurde ein Endschalter angesprochen. Die aktuelle Bewegung wurde abgebrochen. Dies geschieht durch ein stoppen der Bewegung ohne Bremsrampe. Dadurch sind die Istpositionen der Steuerung nicht mehr korrekt, wahrscheinlich sind Schrittverluste aufgetreten. - Die Referenzfahrt einer Schrittmotorachse wurde nicht korrekt oder noch nicht ausgeführt. ACHTUNG: Nach einem Endschalterfehler muß die Steuerung neu initialisiert und eine Referenzfahrt ausgeführt werden.
3	unzulässige Achsangabe - Der Steuerung wurde eine Achsangabe für einen auszuführenden Befehl übermittelt, die eine nicht definierte Achse enthält. - Verwenden Sie in den Befehlen, die Achsangaben enthalten nur Kombinationen aus Achsen, die auch initialisiert sind.
4	keine Achsen definiert - Bevor der Steuerung Bewegungen oder allgemein Befehle, die eine von der Achsenzahl abhängige Anzahl von Parametern haben, übergeben werden, muß der Befehl "Achsenanzahl setzen" übergeben werden, um die internen Achsenparameter korrekt zu setzen.
5	Syntax-Fehler - Ein Befehl wurde fehlerhaft übermittelt. - Der verwendete Befehl existiert nicht oder kann von dieser Steuerung nicht abgearbeitet werden. - Überprüfen Sie, ob alle übertragenen Befehle korrekt sind.
6	Speicherende - Es wurde versucht mehr Befehle im CNC-Mode zu übertragen, als in der Steuerung gespeichert werden können.
7	unzulässige Parameterzahl - Die Steuerung hat mehr oder weniger Parameter für den Befehl erhalten, als benötigt werden. - Prüfen Sie, ob die Anzahl der Parameter für den Befehl in Verbindung mit der Anzahl der Achsen korrekt ist.
8	zu speichernder Befehl inkorrekt - Der Steuerung wurde ein Befehl übergeben, der als CNC-Befehl nicht verfügbar ist.
9	Anlagenfehler - Die Spannungsversorgung der Anlage ist noch nicht eingeschaltet. - Der Sicherheitskreis der Anlage ist nicht aktiv. - Die Endstufen und/oder der Sicherheitskreis konnten nicht eingeschaltet werden, weil die Haube noch geöffnet ist. - Eine Notaus-Situation ist eingetreten. ACHTUNG: Nach einer Notaus-Situation muß die Steuerung neu initialisiert und eine Referenzfahrt ausgeführt werden.
A	von dieser Steuerung nicht benutzt

B	von dieser Steuerung nicht benutzt
C	von dieser Steuerung nicht benutzt
D	unzulässige Geschwindigkeit - Die zulässigen Grenzen für Geschwindigkeitsangaben wurden nicht eingehalten. - Prüfen Sie, ob alle Geschwindigkeitsangaben korrekt sind.
E	von dieser Steuerung nicht benutzt
F	Benutzerstop - Der Benutzer hat die Halttaste an der Steuerung betätigt, die aktuelle Bewegung wurde angehalten. Die Befehlsausführung kann mit der Starttaste oder dem Startbefehl "@0s" wieder aufgenommen werden.
G	ungültiges Datenfeld - Der Steuerung wurde ein Startbefehl übermittelt, obwohl kein Bewegungsrest im Speicher vorhanden, d.h. obwohl zuvor keine Haltfunktion ausgeführt wurde. - Es wurde versucht ein CNC-Programm zu übertragen, obwohl im Speicher noch ein Programm oder Teile eines Programmes vorhanden sind.
H	Haubenfehler - Es wurde versucht einen Befehl auszuführen, der bei geöffneter Haube nicht zulässig ist.
=	von dieser Steuerung nicht benutzt

5. Standardportbelegung für isel-Anlagen mit IMC4-M Steuerung

Festlegung der Beschaltung der Ein-/Ausgänge für IMC4-M mit GeckoBoard

Signal / Bezeichnung	logischer Port* in ProNC / Remote	lokaler Port** auf IMC4-M	Stecker	Ort	Terminal- Befehl	Klemme auf IMC4-M
Standardbeschaltung						
Schaltpegel						IO-Interface
Spannzange: 1=öffnen / 0=schließen	A1.4	1	X30	GeckoBoard	@0B0,8	
Werkstück spannen: 1=ein / 0=aus	A1.5	1	X31	GeckoBoard	@0B0,16	
Kühlung: 1=ein / 0=aus	A1.7	1	X32	GeckoBoard	@0B0,64	
WZW-Haube: 1=öffnen / 0=schließen	A2.5	7	CN106.6	IMC4M	@0B6,16	Out5
Ausgang 2.6 (frei)	A2.6	7	CN106.2	IMC4M	@0B6,32	Out6
Ausgang 2.7 (frei)	A2.7	7	CN106.7	IMC4M	@0B6,64	Out7
Ausgang 2.8 (frei)	A2.8	7	CN106.3	IMC4M	@0B6,128	Out8
Spindel: 1=ein / 0=aus	X	3	X19	GeckoBoard	@0B2,1	
Stop-Taste am Bedienpanel: 0=betätigt	E1.1	1	X21	GeckoBoard	@0b0	
Start-Taste am Bedienpanel: 1=betätigt	E1.2	1	X34	GeckoBoard	@0b0	
Längenmesstaster: 0=betätigt	E1.3	1	X27	GeckoBoard	@0b0	
Power OK: 1=Betriebspann. Motorendstufen vorh.	E1.4	1	X29	GeckoBoard	@0b0	
Eingang 2.2 (frei)	E2.2	7	X28	GeckoBoard	@0b6	
WZW-Haube: 1=geschlossen	E2.3	7	CN106.8	IMC4M	@0b6	In1
Luftdruck: 1=OK, d.h. ausreichend	E2.4	7	CN106.6	IMC4M	@0b6	In2
Eingang 2.5 (frei)	E2.5	7	CN106.9	IMC4M	@0b6	In3
Start-Taste am Bedienpanel: 1=betätigt	E3.4***	16	X34	GeckoBoard		

* "logische Port" beginnen mit 1

** "lokaler Port auf IMC4-M" beginnen mit 1

*** konfigurationsabhängig

Hinweis: Diese Tabelle ist verbindlich ab KW39/2005 mit dem IMC4-M-Firmwarestand V3.2.0

WZW=Werkzeugwechsler

6. Steckerbelegung IMC4-M

CN1 SubD-Stecker-37polig

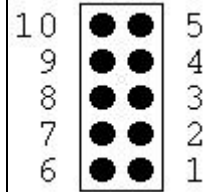
Pin SubD37 IMC4-M	Signal	Anschlussklemme auf Board
1	Takt X	
2	Takt Z	
3	nc (Takt B)	
4	Richtung Y	
5	Richtung A	
6	+24V	
7	Anwender Ausgang 1 (Spannzange auf/zu)	X30
8	Anwender Ausgang 3 (Kühlung ein/aus)	X32
9	Ausgang Enable Endstufen	
10	GND 24V	
11	Referenzschalter A (negativer Endschanter)	
12	Referenzschalter Z (negativer Endschanter)	
13	Referenzschalter Y (negativer Endschanter)	
14	Referenzschalter X (negativer Endschanter)	
15	nc (Referenzschalter B)	
16	GND analog (0-10V Umrichter)	X20
17	Eingang PowerOK	X29
18	Anwender Eingang 2	X28
19	Eingang Starttaster	X34
20	Takt Y	
21	Takt A	
22	Richtung X	
23	Richtung Z	
24	nc (Richtung B)	
25	+5V	
26	Anwender Ausgang 2	X31
27	Ausgang Spindel Start	X19
28	Ausgang Bremse	
29	positiver Endschanter A	
30	positiver Endschanter Z	
31	positiver Endschanter Y	
32	positiver Endschanter X	
33	nc (positiver Endschanter B)	
34	Ausgang analog 0-10V	X20
35	Eingang Stoptaster	X21
36	Anwender Eingang 1 (Längenmesstaster)	X27
37	GND 24V	

CN104 SubD-Stecker-9polig

Pin	Signal
1	
2	RXD
3	TXD
4	
5	Digital GND
6	
7	Starttaster
8	
9	

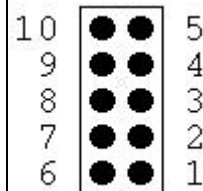
CN105 Stifteleiste 2X5polig

Pin	Signal
1	24V-Versorgungsspannung
2	Ausgang Haube Enable
3	Eingang Notausschalter
4	Eingang Schlüsselschalter
5	
6	Ausgang SoftwareOK
7	Eingang Eintaster
8	Eingang Haube geschlossen
9	
10	GND-Versorgungsspannung



CN106 Stifteleiste 2X5polig

Pin	Signal
1	24V-Versorgungsspannung
2	Anwender Ausgang 6
3	Anwender Ausgang 8
4	Anwender Eingang 4
5	
6	Anwender Ausgang 5
7	Anwender Ausgang 7
8	Anwender Eingang 3
9	Anwender Eingang 5
10	GND-Versorgungsspannung



CN201 Stifteleiste 2X2polig

Pin	Signal
1	5V-Spannungsversorgung
2	I2C-Bus Daten
3	I2C-Bus Clock
4	Digital GND

