Handbuch

Venus-1

für

Corvus

Über diese Dokumentation

In diesem Handbuch ist die Interpretersprache Venus-1 der Steuerung Corvus beschrieben.

Für den schnellen Überblick sind die Kommandos in Funktionsgruppen eingeteilt, eine Indextabelle und die alphabetische Kommandoliste geben zusätzliche Hilfestellung.

In der Einführung werden die internen Mechanismen der Befehlsverarbeitung erläutert. Das Studium dieses Kapitels ist dringend zu empfehlen.

Soweit dies für das Verständnis der Zusammenhänge notwendig ist, wird auch auf hardwarespezifische Besonderheiten der Steuerung eingegangen. Detailliertere Informationen dazu finden sich in der Betriebsanleitung.

Inhaltsverzeichnis

Geschwindigkeit und Beschleunigung

Positionierkommandos

Abbruchkommandos

Arbeitsbereich und Bezugspunkt

Statusabfragen

Input / Output Funktionen

Sonderfunktionen

Joystick / Handrad

Systemkommandos

Einführung in Venus-1

Venus-1 ist eine Interpretersprache

Venus-1 Kommandos bestehen aus ASCII-Zeichen, die von der Steuerung interpretiert und ausgeführt werden.

Eine Software Entwicklungsumgebung zur Erzeugung der Steuerprogramme wird nicht benötigt.

Die Kommandos können von einem beliebigen Host und unabhängig von der Programmiersprache erzeugt werden; Voraussetzung ist der Zugriff auf die RS-232 Schnittstelle bzw. Ethernet Schnittstelle.

Im einfachsten Fall werden die Kommandos direkt von einem ASCII -Terminal an die Steuerung übertragen.

Historie und Kompatibilität

Venus-1 für Corvus ist eine Weiterentwicklung der bewährten Kommandosprache, die bisher für die Steuerungen mc-compact, smc-compact, MC-2000 und MC-3000 verwendet wurde.
Der grundsätzliche Befehlssyntax wurde beibehalten.
Alle Basisfunktionen sind kompatibel zu der früheren Version.

Befehlssyntax für die Parametrisierung

Die Parametrisierung erfolgt mit folgender Syntax:

[Parameter] _ [Achsenindex] _ [Kommando] _

_ = Leerzeichen, (Space) oder (SP)

Parameter

Der Parameter übergibt einen Wert ohne Einheit.
Sind für ein Kommando mehrere Parameter vorgeschrieben, müssen diese durch ein Leerzeichen (SP) voneinander getrennt werden.

Für Parameter sind folgende Zahlen und Zeichen erlaubt:

Buchstaben	keine
Zahlen	0 - 9
Zeichen	+ -.

-1 Parameter

Bei verschiedenen get-Kommandos kann der Parameter -1 vorangestellt werden, damit wird mit einem Kommando die Einstellung aller Achsen ausgelesen.

Beispiel:
Der ***2 getpitch*** liefert die Einstellung der Spindelsteigung von Achse-2.
Mit -1 getpitch wird die Einstellung aller Achsen ausgelesen.

Achsenindex

Mit dem Achsenindex wird die Zielachse für den Parameter adressiert. Die Nummerierung erfolgt analog zur der Bezeichnung am Motoranschluss.

Achsbezeichnung	Achsenindex
Axis-1	1
Axis-2	2
Axis-3	3

Kommandos

Für die Parametrisierung werden Kommandos verwendet die mit get_ und set_ bezeichnet sind. Es wird zwischen Gross- und Kleinschreibung unterschieden.

Für Kommandos sind folgende Zeichen erlaubt:

Buchstaben	a-z, A-Z
Zahlen	keine
Zeichen	keine

Befehlssyntax bei Positionierbefehlen

[Axis-1] _ [Axis-2] _ [Axis-3] _ [Kommando] _

Axis-1, Axis-2, Axis-3

Für die Positionierung werden die absoluten oder relativen Zielkoordinaten der Achsen an die Steuerung übergeben. Die Werte werden durch ein Leerzeichen voneinander getrennt.

Die Anzahl der Achsenkoordinaten, die mit dem Befehl übergeben werden müssen, ist abhängig von der Einstellung ***setdim***.

setdim	Achsen
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

Werden zu wenige Koordinaten angegeben, wird das Kommando nicht ausgeführt, bei zu vielen Werten verbleiben überschüssige Elemente auf dem Stack.

Bei der Angabe der Koordinaten sind folgende Zahlen und Zeichen erlaubt:

Buchstaben	keine
Zahlen	0 - 9
Zeichen	+ -.

Befehlsabschlusszeichen beim Senden

Im **Host Modus** werden die Kommandos mit einem Leerzeichen ASCII abgeschlossen:
[Parameter] _ [Achsenindex] _ [Kommando] _

Im **Terminal Modus** erfolgt der Befehlsabschluss durch CR (carriage return).
Parameter] _ [Achsenindex] _ [Kommando] CR

Befehlsabschlusszeichen beim Empfangen

Daten, die von der Steuerung zurückgeliefert werden, sind immer mit ASCII (CR) und (LF) abgeschlossen.

[1.Parameter] _ [2.Parameter] _ [n.Parameter] CR LF

Bei verschiedenen get-Kommandos werden die Parameter in mehreren Zeilen zurückgeliefert. Auch in diesen Fällen ist jede Zeile mit (CR) und (LF) abgeschlossen.
Wie viele Zeilen eine Anfrage zurückmeldet, ist in der Kommandobeschreibung angegeben.

Wichtige ASCII Zeichen bei der Programmierung

ASCII Code	Zeichen	Dez	HEX
CR	Ctrl-M	13	0xD
LF	Ctrl-J	10	0xA
SP		32	0x20
ETX	Ctrl-C	3	0x3

Befehlsverarbeitung

Die von einem Host übertragenen ASCII Daten durchlaufen folgende Funktionsgruppen:

- **Dateneingangsspeicher**
- **Scanner und Stack**
- **Interpreter**

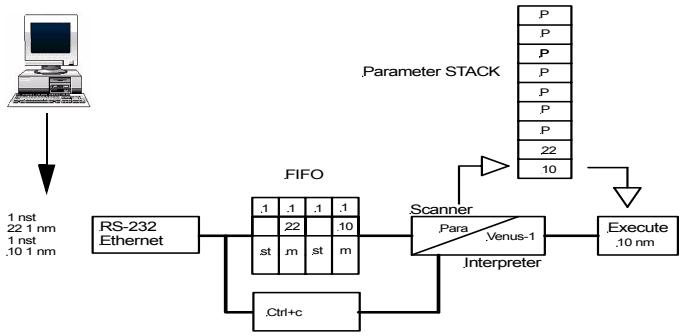
Dateneingangsspeicher

Die von der Schnittstelle übertragenen Zeichen werden zunächst in diesen Eingangspuffer übertragen. Der Speicher besitzt eine FIFO Struktur (First_In_First_Out).

Der Dateneingangsspeicher kann bis zu 256 Zeichen aufnehmen. Bei der Übertragung der Daten wird keine Datenflusskontrolle durchgeführt, das heisst, ein Überlauf des FIFO wird nicht erkannt.
Bei der Programmierung ist deshalb darauf zu achten, dass nicht zu viele Daten zu schnell an die Steuerung übertragen werden.
Der Eingangspuffer ist nach dem Abschalten der Steuerung gelöscht.

Scanner -> Stack -> Interpreter

Die Daten im Eingangspuffer der Schnittstelle werden vom Scanner sequentiell ausgelesen, dabei werden Parameter und Venus-1 Kommandos getrennt. Die Parameter werden in einen Stapelspeicher (Stack) gelegt, der bis zu 99 Werte aufnehmen kann. Kommandos werden dem Interpreter übergeben, sobald dieser den vorherigen Befehl abgearbeitet hat. Der Interpreter holt sich die dem Kommando zugeordneten Parameter vom Stack und führt den Befehl aus.



Sperrende und nicht sperrende Kommandos

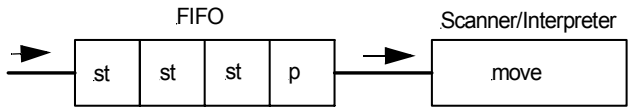
Während der Interpreter eine Positionierung ausführt, ist die gleichzeitige Verarbeitung bestimmter Kommandos möglich. Diese werden auch als **nicht sperrende Kommandos** bezeichnet.

Im Gegensatz dazu gibt es Kommandos, welche erst dann abgearbeitet werden können, wenn die Positionierung abgeschlossen ist.
Befindet sich solch ein Kommando im Dateneingangspuffer, wird die Ausführung aller dahinter liegenden Kommandos gesperrt bis das sperrende Kommando selbst abgearbeitet ist und so aus dem FIFO entfernt wurde.
Diese werden deshalb als sperrende **Kommandos** bezeichnet.

Beispiele von sperrenden und nicht sperrenden Kommandos

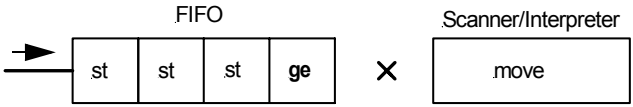
Der Interpreter verarbeitet mehrere Kommandos gleichzeitig:

Im unten stehenden Beispiel führt der Interpreter den Befehl ***move*** aus. Die im FIFO befindlichen Kommandos ***p*** und 3 x ***st*** werden danach ebenfalls verarbeitet.



Der Interpreter wurde mit dem Kommando *ge* gesperrt.

Der Interpreter führt hier einen ***move*** Kommando aus. Das Kommando ***ge*** blockiert den Interpreter, dadurch können die dahinter liegenden Befehle nicht abgearbeitet werden. Nachdem der Positionierbefehl ausgeführt wurde, kommt ***ge*** zur Ausführung, erst danach wird auch der Befehl ***st*** ausgeführt.



Liste der nicht sperrenden Kommandos

Die folgenden Kommandos bewirken keine Sperrung der Befehlsverarbeitung, diese werden auch verarbeitet während der Interpreter ein Positionierkommando ausgeführt.

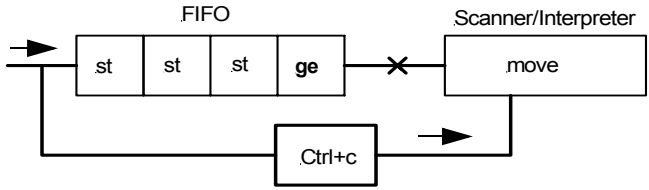
Venus-1 Kommando	Funktion
st	Statusrückmeldung
p	Aktuelle Position lesen
getin	Digitalen Eingang lesen
setout	Digitalen Ausgang schreiben
abort	Der aktuell ausgeführte Befehl wird abgebrochen. Achtung: Dieser Abbruchbefehl durchläuft im Gegensatz zu Ctrl+c das Daten-FIFO und kann somit durch ein anders Kommando blockiert werden.
Ctrl+c	Der aktuell ausgeführte Befehl abgebrochen. Dieser Befehl durchläuft nicht das FIFO und kann daher nicht durch ein sperrende Kommando verzögert werden.

Interpreter entsperren

Eine dauerhafte Blockierung des Interpreters durch ein sperrendes Kommandos ist nicht möglich, da letzt endlich immer alle Kommandos vom Interpreter abgearbeitet werden.
Um das Entsperren zu beschleunigen, kann der Abbruchbefehl ***Ctrl+c*** genutzt werden.
Damit wird immer der aktuell im Interpreter befindliche Befehl vorzeitig abgebrochen und so eine schnelleres Abarbeiten der im Dateneingangspuffer befindlichen blockierenden Kommandos erreicht.

Befehle abbrechen

Ein aktuell ausgeführter Befehl wird durch ***Ctrl+c*** sofort abgebrochen.
Wie dargestellt durchläuft dieser Befehl nicht den Dateneingangsbuffer sondern wirkt unmittelbar auf den Interpreter.
Das FIFO wird dabei nicht gelöscht.



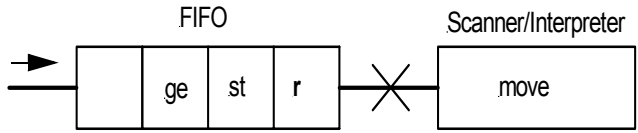
Erzeugen einer automatischen Statusrückmeldung

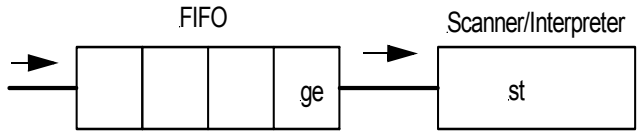
Mit der folgenden Befehlssequenz kann die sperrende Wirkung eines Kommandos für die Erzeugung einer automatische Statusrückmeldung genutzt werden:

10 10 2 move
0 0 0 r
st
ge

Corvus liefert so automatisch eine Statusrückmeldung wenn der Positionierbefehl ***10 10 2 move*** abgeschlossen ist.

Der Befehl ***0 0 0r*** selbst hat keine Wirkung, sondern hat nur die gewünschte Aufgabe die Ausführung der Statusabfrage so lange zu sperren bis der ***move*** abgearbeitet ist.

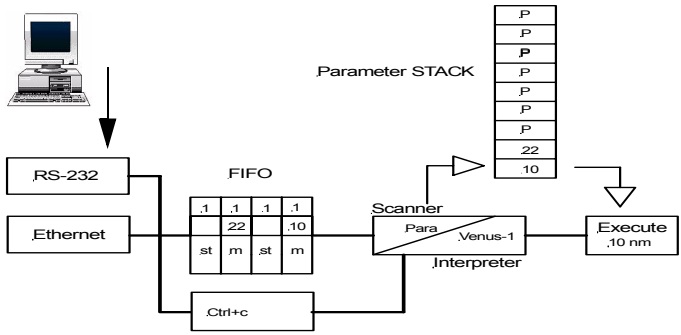




Corvus Kommunikationskonzept

Ethernet und RS-232

Corvus ist standardmässig mit einer RS-232 Schnittstelle ausgestattet. Optional kann die Ethernetschnittstelle freigeschaltet werden.
Beide Schnittstellen sind dann gleichzeitig empfangsbereit. Die Datenrückmeldung erfolgt automatisch immer auf die Schnittstelle von der die Datenanfrage stammt.
Terminal und Host Mode wird von beiden Schnittstellen unterstützt.



Kommunikation

mode

Beschreibung:

Mit dem Kommando ***mode*** wird die Kommunikation auf den Terminal oder Host Mode eingestellt.

Im **Terminal Mode** überträgt die Steuerung automatisch eine Bildschirmmaske mit Positionsanzeige und Venus-1 Kommando Eingabezeile an alle aktiven Schnittstellen.


speicherbar

Im **Host Modus** werden nur Daten übertragen wenn diese vom Host angefordert werden.

Alternativ ist es auch möglich die beiden Modi mit Hilfe von DIP-Schalter-6 direkt an der Steuerung einzustellen.

Siehe **Corvus Betriebsanleitung**.

Syntax:

[Index] ***mode***

[Index]	Beschreibung
0	Host Mode
1	Terminal Mode

Beispiel:

1 mode
Die Steuerung wird in den Terminal Mode geschaltet

setipadr

Beschreibung:

Mit dem Kommando ***setipadr*** wird die IP-Adresse der Steuerung für die Kommunikation mit der Ethernetschnittstelle festgelegt.

Die Subnetmask ist derzeit fest vorgegeben.
Sie hat den Wert: 255.255.255.0
Die Anzahl der Netzknoten ist damit auf 254 begrenzt.


speicherbar

Die Netzwerk-Maske muss bei der Installation des Netzwerk-Protokolls genauso wie die IP-Adresse angegeben werden. Empfängt die Steuerung im Netz einen Frame, so vergleicht-sie zunächst anhand der Netzwerk-Maske, ob der Frame überhaupt für sie in diesem Netzwerk-Segment bestimmt ist. Die richtige Netzwerk-Nummer ermittelt sie dabei aus der eigenen IP-Adresse und der 255er-Maske. Nur bei Übereinstimmung schaut die Steuerung anschließend, ob die Knoten-Nummer gleich der eigenen ist. Trifft auch dies zu, dann erst liest sie den gesamten Inhalt des Frames. Aufgrund dieses Ablaufs ist es ersichtlich, wie wichtig eine korrekte Adressierung bei Verwendung des TCP/IP-Protokolls ist, da sonst keine Kommunikation zwischen den einzelnen Geräten im Netzwerk stattfinden kann.

Syntax:

[AAA]_[BBB]_[CCC]_[DDD]_setipadr

Die Blöcke sind mit einem Leerzeichen zu trennen !

Partnerbefehl:

getipadr

Beispiel:

192_168_128_0_setipadr

getipadr

Beschreibung:

Der Befehl ***getipadr*** liest die eingestellte IP-Adresse der Steuerung zurück.

Syntax:

getipadr

Rückmeldung:

[Adresse]

Verwandter Befehl:

getmacadr (Auslesen der MAC-Adresse)

Beispiel:

getipadr

Rückmeldung:

192.168.128.0

Die Blöcke sind bei der Rückmeldung mit einem Punkt getrennt.

Grundeinstellungen

getfpara

Beschreibung:

Das Kommando ***getfpara*** aktiviert die Werkseinstellung.

Die aktuellen Einstellungen werden überschrieben aber nicht abgespeichert.

Syntax:

getfpara

Rückmeldung:

keine

Beispiel:

getfpara

setdim

Beschreibung:

Mit dem Kommando ***setdim*** wird festgelegt wieviele Parameter die Steuerung bei dimensionsabhängigen Kommandos erwartet oder selbst zurückliefert.
Mit dem Befehl ***setdim*** ist es **nicht** möglich eine Achse abzuschalten.
Die Wirkung der Einstellung ist auf die folgenden Kombinationen begrenzt.

setdim	Erwartete oder zurückgelieferte Achsenparamter
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

Beispiele:

Positionierkommandos
Abhängig von der Einstellung ***setdim*** müssen 1, 2 oder 3 Koordinatenwerte mit dem Positionskommando übertragen werden.

Positionsrückmeldung
Abhängig von der Einstellung ***setdim*** werden 1, 2 oder 3 Positionswerte mit dem Kommando ***pos*** zurückgeliefert.

Syntax:

[Dimension] ***setdim***

	Wertebereich
[Dimension]	1, 2, 3

Partnerbefehl:

getdim

Beispiel:

2 setdim

getdim

Beschreibung:

Das Kommando ***getdim*** liest die mit ***setdim*** eingestellte Dimension zurück

Syntax:

getdim

Rückmeldung:

[Dimension]

	Wertebereich
[Dimension]	1, 2, 3

Beispiel:

getdim

setunit

Beschreibung:

Das Kommando ***setunit*** definiert die physikalische Einheit der Ein- und Ausgabewerte.
Mit dem Index 1 bis 3 kann jeder Achse eine eigene Einheit zugeordnet werden.
Index 0 bestimmt die Einheit der virtuellen 0-Achse und damit der Geschwindigkeit und Beschleunigung.

Die Einheiten der Geschwindigkeitswerte die mit ***setcalvel*** und ***setrmvel*** eingestellt wird können mit ***setunit*** nicht beeinflusst werden, diese sind fest auf Umdrehungen/s eingestellt.

Die Einheit Microstep wird aus Kompatibilitätsgründen von Corvus emuliert. Die Auflösung der Steuerung ist damit reduziert: 1 Microstep = 1 Umdrehung / 40000
Für Neuentwicklungen sollte diese nicht mehr verwendet werden.

Syntax:

[Index] [Achse] ***setunit***

	Wertebereich
[Index]	0, 1, 2, 3, 4, 5, 6
[Achse]	0, 1, 2, 3

[Index]	**Masseinheit**
0	Microstep
1	µm
2	mm
3	cm
4	m
5	inch
6	mil (1/ 1000 inch)

Verwandter Befehl:

getunit

Beispiel:

2 0 setunit
Die virtuelle 0-Achse wird auf **mm** eingestellt.
Damit erfolgt die Ein- und Ausgabe des Geschwindigkeits- und Beschleunigungswerte in dieser Einheit.

1 1 setunit
Die Einheit für Achse-1 wird auf **µm** eingestellt.
Alle achsspezifischen Ein- Ausgaben sowie auch die Positionsrückmeldung erfolgen damit in dieser Einheit.

getunit

Beschreibung:

Das Kommando ***getunit*** liefert die mit ***setunit*** vorgenommene Einstellungen zurück.

Syntax:

[Achse] ***getunit***

	Wertebereich
[Achse]	-1, 0, 1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1, 2, 3, 4, 5, 6

Beispiel:

1 getunit

Rückmeldung:
2

-1 getunit

Rückmeldung:
2 1 1 1

setumotmin

Beschreibung:

speicherbar

[Mit dem Kommando ***setumotmin*** wird ein Stromindex übergeben der das Haltemoment sowie das Drehmoment im unteren Drehzahlbereich beeinflusst.
Der übergebene Wert hat keinen Bezug zu den sich damit einstellenen physikalischen Größen.

Mit größerem Wert des Index vergrößert sich auch der Phasenstrom, dies führt zu einer steigenden Verlustleistung am Motor und stärkerer Belastung der Motorendstufe.

Syntax:

[Index] [Achse] ***setumotmin***
[

	Wertebereich
[Index]	0 - 3000
[Achse]	1, 2, 3

Verwandte Befehle:

getumotmin, setumotgrad

Beispiel:

2000 1 setumotmin

getumotmin

Beschreibung:

Das Kommando ***getumotmin*** liest den durch ***setumotmin*** eingestellten Stromindex der Achse zurück.

Syntax:

[Achse] ***getumotmin***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0 - 3000

Beispiel:

1 getumotmin

Rückmeldung:
1000

-1 getumotmin

Rückmeldung:
1000
1000
750

setumotgrad

Beschreibung:

speicherbar

Mit dem Kommando ***setumotgrad*** wird ein Stromindex übergeben der das Drehmoment im mittleren und oberen Drehzahlbereich beeinflusst.
Der übergebene Wert hat keinen Bezug zu den sich damit einstellenen physikalischen Größen.

Mit größerem Wert des Index vergrößert sich auch der Phasenstrom, dies führt zu einer steigenden Verlustleistung am Motor und stärkerer Belastung der Motorendstufe.

Syntax:

[Index] [Achse] ***setumotgrad***

	Wertebereich
[Index]	0 - 300
[Achse]	1, 2, 3

Verwandte Befehle:

getumotgrad, setumotmin

Beispiel:

70 1 setumotgrad

getumotgrad

Beschreibung:

Der Befehl ***getumotgrad*** liest den durch ***setumotgrad*** eingestellten Wert zurück.

Syntax

[Achse] ***getumotgrad***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0 - 300

Beispiel:

1 getumotgrad

Rückmeldung:
50

-1 getumotgrad

Rückmeldung:
50
50
100

setpolepairs

Beschreibung:

Mit dem Kommando ***setpolepairs*** wird die Anpassung an die Polpaarzahl des Motors vorgenommen.
In der untenstehenden Tabelle sind Polpaarzahl und Motortyp dargestellt.


speicherbar

Polpaare	Motortyp
50	Schrittmotor 1.8°
100	Schrittmotor 0.9°

Syntax:

[Polpaare] [Achse] ***setpolepairs***

	Wertebereich
[Polpaare]	50, 100
[Achse]	1, 2, 3

Partnerbefehl:

getpolepairs

Beispiel:

50 1 setpolepairs

Achse-1 wird für einen 1,8° Schrittmotor konfiguriert.

getpolepairs

Beschreibung:

Das Kommando ***getpolepairs*** liest die eingestellte Polpaarzahl zurück.

Syntax:

[Achse] ***getpolepairs***

Parameter	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[Wert]

	Wertebereich
[Wert]	50, 100

Beispiel:

1 getpolepairs

Rückmeldung:
50

-1 getpolepairs

Rückmeldung:
50 100 50

setaxis

Beschreibung:


speicherbar

Das Kommando ***setaxis*** aktiviert oder deaktiviert die spezifizierte Achse für die Positionierung oder die Endschalterfahrt.
Gleichzeitig kann die Wirkung der Kommandos ***setpos, cal*** und ***rm*** auf die Positionsanzeige und die Limits beeinflußt werden.
Die Einstellung ist für den programmierten und manuellen Betrieb gültig.

Syntax:

[Wert] [Achse] ***setaxis***

	Wertebereich
[Wert]	0, 1, 2, 3, 4
[Achse]	1, 2, 3

[Wert] = 0: Die Achse ist konsequent abgeschaltet
Die Kommandos ***cal***, ***rm*** und ***setpos*** bewirken ein zurücksetzen der Position, die Limits der Achse werden nicht verändert.

[Wert] = 1: Achse konsequent eingeschaltet.
Die Kommandos ***cal***, ***rm*** und ***setpos*** bewirken bei dieser Achse ein zurücksetzen der Position und der Limits.

[Wert] = 2: Die Achse ist begrenzt eingeschaltet.
Die Endschalterfahrt wird bei dieser Achse nicht ausgeführt.
Die Kommandos ***cal***, ***rm*** und ***setpos*** bewirken ein Rücksetzen der Position.
Die Limits der Achse bleiben erhalten.

[Wert] = 3: Die Achse ist konsequent abgeschaltet.
Die Kommandos ***cal***, ***rm*** und ***setpos*** haben keine Wirkung auf die Position und die Limits der Achse.

[Wert] = 4: Die Achse ist beqrenzt eingeschaltet.
Die Endschalterfahrt wird bei dieser Achse nicht ausgeführt.
Die Kommandos ***cal***, ***rm*** und ***setpos*** haben keine Wirkung auf die Position und die Limits der Achse

Verwandter Befehl:

getaxis

Beispiel:

1 3 setaxis
Achse-3 wird eingeschaltet

0 1 setaxis
Achse-1 wird abgeschaltet.

getaxis

Beschreibung:

Das Kommando ***getaxis*** liefert die mit ***setaxis*** vorgenommene Einstellungen zurück.

Syntax:

[Achse] ***getaxis***

Rückmeldung:

[Wert]

	Wertebereich
[Wert]	0, 1, 2, 3, 4

Beispiel:

2 getaxis

Rückmeldung:
1

-1 getaxis

Rückmeldung:
1 2 2

setcalvel

Beschreibung:

speicherbar

Mit dem Kommando ***setcalvel*** werden zwei Geschwindigkeiten festgelegt, mit denen die Steuerung die cal Endschalterfahrt ausführt.

1. Geschwindigkeit in den Endschalter hinein.
2. Geschwindigkeit aus dem Endschalter heraus.

Die Einstellung ist für alle Achsen gültig.

HINWEIS!

Aus Kompatibilitätsgründen erfolgt die Angabe der Geschwindigkeiten in der Einheit Umdrehungen/s.
Die daraus resultierende Geschwindigkeit in mm/s ergibt sich aus der Spindelsteigung der virtuellen 0-Achse.
Siehe “” auf Seite 53.

Syntax:

[Geschwindigkeit] [Index] ***setcalvel***

[

	Wertebereich	Einheit
[Geschwindigkeit]	0 - 45	Umdrehung/s
[Index]	1, 2	-

[Index]	Beschreibung
1	Geschwindigkeit in den Endschalter hinein
2	Geschwindigkeit aus dem Endschalter heraus

Verwandte Befehle:

getcalvel, setrmvel

Beispiel:

```
2  0 setpitch
2  1 setcalvel
1  2 setcalvel
```

Die Steuerung bewegt die Achsen mit 2 U/s (4 mm/s) in die cal-Endschalter hinein und mit 1 U/s (2 mm/s) aus den cal-Endschaltern heraus.

getcalvel

Beschreibung:

Das Kommando ***getcalvel*** liest die mit ***setcalvel*** eingestellte Geschwindigkeit der Endschalterfahrt in den cal-Endschalter zurück.

Syntax:

getcalvel

Rückmeldung:

[Wert 1]
[Wert 2]

	Wertebereich	Einheit
[Wert 1]	0 - 45	Umdrehung/s
[Wert 2]	0 - 45	Umdrehung/s

Beispiel:

getcalvel

Rückmeldung:

2.000000
0.250000

setrmvel

Beschreibung:

speicherbar

Das Kommando ***setrmvel*** definiert die beiden Geschwindigkeiten, mit denen die Steuerung die rm-Endschalterfahrt der Achsen ausführt.

1. Geschwindigkeit: In den Endschalter hinein.
2. Geschwindigkeit: Aus dem Endschalter heraus.

Die Einstellung ist für alle Achsen gültig.

HINWEIS!

Aus Kompatibilitätsgründen erfolgt die Angabe der Geschwindigkeiten in der Einheit Umdrehungen/s.
Die daraus resultierende Geschwindigkeit in mm/s ergibt sich aus der Spindelsteigung der virtuellen 0-Achse.
Siehe “” auf Seite 53.

Syntax:

[Geschwindigkeit] [Index] ***setrmvel***

	Wertebereich	Einheit
[Geschwindigkeit]	0 - 45	Umdrehung/s
[Index]	1, 2	-

I

[Index]	Beschreibung
1	Geschwindigkeit in den Endschalter hinein
2	Geschwindigkeit aus dem Endschalter heraus

Verwandte Befehle:

getrmvel, setcalvel

Beispiel:

```
2  0 setpitch
2  1 setrmvel
1  2 setrmvel
```

Die Steuerung bewegt die Achsen mit 2 U/s (4 mm/s) in die rm-Endschalter hinein und mit 1 U/s (2 mm/s) aus den rm-Endschaltern heraus.

getrmvel

Beschreibung:

Das Kommando ***getrmvel*** liest die mit ***setrmvel*** eingestellte Geschwindigkeit für die Endschalterfahrt in den rm-Endschalter zurück.

Syntax:

getrmvel

Rückmeldung:

[Wert 1]
[Wert 2]

	Wertebereich	Einheit
Wert 1	0 - 45	Umdrehung/s
Wert 2	0 - 45	Umdrehung/s

Beispiel:

getrmvel

Rückmeldung:

2.000000
0.250000

setaccelfunc

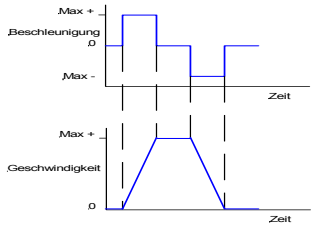
Beschreibung:

speicherbar

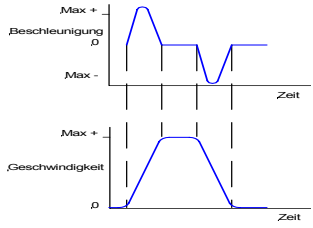
Das Kommando ***setaccelfunc*** definiert die Beschleunigungsfunktion, mit der die Positionierung der Achsen ausgeführt wird. Die Einstellung wirkt auf alle Achsen.

Folgende Beschleunigungsfunktionen sind möglich:

- Lineare Beschleunigung (Trapez)
- Sin² -Beschleunigung (S-Kurve)



Lineare Beschleunigung



sin²- Beschleunigung

Syntax:

[Index] ***setaccelfunc***

[Index]	Beschreibung
0	Lineare Beschleunigung (Werkseinstellung)
1	Sin²-Beschleunigung

Verwandter Befehl:

getaccelfunc

Beispiel:

1 setaccelfunc

Die Achsen werden mit einer Sin²-Beschleunigung Funktion positioniert.

getaccelfunc

Beschreibung:

Das Kommando ***getaccelfunc*** liest die eingestellte Beschleunigungsfunktion zurück.

Syntax:

getaccelfunc

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1

Beispiel:

getaccelfunc

setcalswdist

Beschreibung:

speicherbar

[Mit dem Kommando ***setcalswdist*** kann für jede Achse ein zusätzlicher Abstand zu den Endschaltern festgelegt werden.
Die Endschalterfahrt besteht dann aus den folgenden Strecken :

1. Stecke zum Endschalter hin
2. Strecke aus dem Endschalter heraus
3. Strecke, die durch ***setcalswdist*** festgelegt wurde.

HINWEIS!

Aus Kompatibilitätsgründen erfolgt die Angabe der Strecke in der Einheit Umdrehungen/s.
Die daraus resultierende Distanz in mm ergibt sich aus der Spindelsteigung der virtuellen 0-Achse.
Siehe “” auf Seite 53.

Syntax:

[Strecke] [Achse] ***setcalswdist***

	Wertebereich	**Einheit**
[Strecke]	0 - >1000	Umdrehungen
[Achse]	1, 2, 3	

Verwandter Befehl:

getcalswdist

Beispiel:

5 1 setcalswdist
Für Achse-1 wird ein zusätzlicher Abstand von 5 Umdrehungen zu den Endschaltern festgelegt.

getcalswdist

Beschreibung

Das Kommando ***getcalswdist*** liest die durch ***setcalswdist*** vorgenommene Einstellung zurück.

Syntax:

[Achse] ***getcalswdist***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Wert]

	Einheit
[Wert]	U/s

Beispiel:

1 getcalswdist

Rückmeldung:

5.00000

-1 getcalswdist

Rückmeldung:

5.00000
0.00000
0.00000

setpitch

Beschreibung:

Mit dem Kommando ***setpitch*** wird die Steuerung an das Übersetzungsverhältnis von Motorumdrehung und Bewegungseinheit angepaßt.


speicherbar

Der Wert von ***setpitch*** entspricht der resultierenden Bewegung bei einer Motorumdrehung.

$$\text{Pitch} = \frac{\text{Bewegungseinheit}}{\text{Motorumdrehung}}$$

Syntax:

[Pitch] [Achse] ***setpitch***

	Wertebereich	**Einheit**
[Pitch]	0.0001 bis 4095	mm
[Achse]	1, 2, 3	

Partnerbefehl:

getpitch

Beispiel:

4.0009 1 setpitch

Bei Achse-1 beträgt das Übersetzungsverhältnis der Motorumdrehung zur resultierenden Bewegung Faktor 4.0009

Weitere Beispiele auf der folgenden Seite.

Beispiele:

Spindelübersetzung:

Eine Spindel erzeugt bei einer Motorumdrehung eine Strecke von 2mm.

Pitch = 2mm / 1 Motorumdrehung = 2

Das Kommando für Achse-1:
2 1 setpitch

Spindel mit Getriebeübersetzung :

Antriebseinheit mit Spindel 4mm
Die Motorumdrehung wird durch ein Getriebe im Verhältnis 120:1 übersetzt.

Pitch = 4mm / 120 Motorumdrehungen = 0.0333

Das Kommando für Achse-3:
0.0333 3 setpitch

Rundtisch mit Getriebe 1:1

Angabe in Umdrehung:
Pitch = 1 Umdrehung / 1 Motorumdrehungen = 1

Angabe in Grad:
Pitch = 360° / 1 Motorumdrehung = 360

Rundtisch mit Getriebe 120:1

Angabe in Umdrehung:
Pitch = 1 Umdrehung / 120 Motorumdrehungen = 0.00833

Angabe in Grad:
Pitch = 360° / 120 Motorumdrehungen = 3

getpitch

Beschreibung:

Das Kommando ***getpitch*** liest die durch ***setpitch*** vorgenommene Einstellung der Achse.

Syntax:

[Achse] ***getpitch***

Parameter	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Wert]

	Einheit
[Wert]	mm

Beispiel:

2 getpitch

Rückmeldung:
4.000900

-1 getpitch

Rückmeldung:
4.000900
2.000000
2.000000

setsw

Beschreibung:

speicherbar

Der Befehl ***setsw*** passt die Endschaltereingänge cal und rm an das Schaltverhalten der Endschalter an.
Mit der Einstellung "ignorieren" wird der Endschaltereingang abgeschaltet.
Folgende Funktionen sind möglich:

- Öffner
- Schliesser
- Endschalter ignorieren

Ein auf "Ignorieren" gesetzter Endschalter kann keine Sicherheitsfunktion erfüllen.

Syntax:

[Funktion] [Endschalter] [Achse] ***setsw***

NPN Endschalter schalten gegen Masse.
PNP Endschalter schalten gegen VCC

[Funktion]	Beschreibung NPN	Beschreibung PNP
0	Schliesser nach GND	Öffner nach VCC
1	Öffner nach GND	Schliesser nach VCC
2	Ignorieren	Ignorieren

[Endschalter]	Beschreibung
0	cal-Endschalter
1	rm-Endschalter

Beipiele:

0 0 1 setsw
2 1 2 setsw

Für NPN Endschalter:
Konfiguriert den cal-Endschaltereingang von Achse-1 als Schliesser gegen GND.
Der rm-Endschaltereingang von Achse-2 wird abgeschaltet.

getsw

Beschreibung:

Das Kommando ***getsw*** liest die Einstellung der Endschaltereingänge zurück.

Syntax:

[Achse] ***getsw***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Funktion cal-Eingang] [Funktion rm-Eingang]

	Wertebereich
Funktion cal-Eingang	0, 1, 2
Funktion rm-Eingang	0, 1, 2

Beispiel:

3 getsw ***-1 getsw***

Rückmeldung: 0 0 1 0 2 2
0 0

setnselpos

Beschreibung:

Das Kommando ***setnselpos*** bestimmt ob die intern errechneten Positionsdaten (Soll-Position) oder die von einem Längenmesssystem erzeugten Positionsdaten (Ist-Position) mit dem Kommando ***p*** zurückgeliefert werden.
Diese Einstellung beeinflusst auch auf die Darstellung der Position im Terminal Mode.

HINWEIS!

Für die Darstellung der Ist-Position muss die Steuerung an ein Längenmesssystem angeschlossen sein.
Siehe Betriebsanleitung: Funktionen / Closed Loop

Syntax:

[Index] [Achse] ***setnselpos***

	Wertebereich
Index	0,1
Achse	1, 2, 3

[Index]	**Beschreibung**
0	Rückmeldung der Soll-Position
1	Rückmeldung der Ist-Position

Partnerbefehl:

getnselpos

Beispiel:

0 3 setnselpos
1 1 setnselpos

Achse-3 liefert die Soll-Position.
Achse-1 liefert die Ist-Position.

getnselpos

Beschreibung:

Das Kommando ***getnselpos*** liefert die Einstellung von ***setnselpos***

Syntax:

[Achse] ***getnselpos***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0,1

Beispiel:

3 getnselpos

setcloop

Beschreibung:


speicherbar

Das Kommando ***setcloop*** aktiviert für die gewählte Achse den Betrieb im geschlossenen Regelkreis.
Die Steuerung benötigt in diesem Modus die Positionsdaten eines externen Längenmeßsystems.

Bitte beachten Sie die weiteren Closed-Loop Einstellungen: ***setclperiod***, ***setclpara*** und ***setnselpos***

Syntax:

[Index] [Achse] ***setcloop***

	Wertebereich
[Index]	0, 1
[Achse]	1, 2, 3

[Index]	**Beschreibung**
0	Closed-Loop abgeschaltet
1	Closed-Loop eingeschaltet

Partnerbefehl:

getcloop

Beispiel:

1 2 setcloop
0 3 setcloop

Closed-Loop für Achse-2 eingeschaltet für Achse-3 abgeschaltet.

getcloop

Beschreibung:

Das Kommando ***getcloop*** liest die Einstellung von ***setcloop***.

Syntax

[Achse] ***getcloop***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1

Beispiel:

1 getcloop

Rückgabe:
1

setref

Beschreibung:

Das Kommando ***setref*** legt fest ob und in welcher Richtung die Flanke des Referenzsignals ausgewertet wird.

speicherbar

Syntax:

[Index] [Achse] ***setref***

	Wertebereich
[Index]	0, 1, 2
[Achse]	1, 2, 3

[Wert]	Beschreibung
0	Steigende Flanke
1	Fallende Flanke
2	Das Referenzsignal wird nicht ausgewertet

Partnerbefehle:

getref / getrefst

Beispiel:

0 1 setref

Der Referenzeingang der Achse-1 reagiert auf die steigende Signalflanke des Referenzsignals.

getref

Beschreibung:

Das Kommando ***getref*** liefert die Einstellungen des Befehls ***setref*** zurück.

Syntax:

[Achse] ***getref***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1, 2

Beispiel:

1 getref

setclperiod

Beschreibung:

Der Befehl ***setclperiod*** adaptiert die spezifizierte Achse an die Auflösung und Zählrichtung eines analogen Längenmeßsystems.
Der Wert von ***setclperiod*** ermittelt sich aus der resultierenden Bewegung pro Massstabsperiode.

Bei digitalen Messsystemen wird die gleiche Anpassung mit dem Befehl ***setclfactor*** vorgenommen.

Durch Ändern des Vorzeichens kann die **Zählrichtung** des Messeingangs an die Drehrichtung des Motors angepasst werden.

Syntax:

[Zählrichtung] [Periode] [Achse] ***setclperiod***

	Wertebereich	**Einheit**
[Zählrichtung]	+,-	Vorzeichen
[Periode]	0.0000001-1.999999	unit
[Achse]	1, 2, 3	

Partnerbefehl:

getclperiod

Beispiele:

- 0.002 3 setclperiod

Beispiele für die Berechnung von ***setclperiod*** auf der folgenden Seite.

Beispiel: Lineares Messsystem

Eine mit Spindel getriebene Achse ist mit einem linearem Massstab ausgestattet. Das Messsystem hat eine Teilungsperiode von 20 µm. Die Spindelsteigung beträgt 10 mm.

Bei linearen Messsystemen besteht eine direkte Verbindung zwischen der Bewegung und dem Messsystem.
Die resultierende Bewegung innerhalb der Massstabsperiode ist damit gleich der Teilungsperiode des Messsystems also unabhängig von einem Getriebe oder der Spindelsteigung.

setclperiod = 0.020

Wird die gleiche Anordnung mit Getriebe 5:1 ausgestattet, muss die Spindelsteigung auf 2 mm eingestellt werden.
Der Wert für ***setclperiod*** ändert sich nicht, da die resultierende Bewegung pro Massstabsperiode gleich bleibt.

Beispiel: Rotationsgeber mit 1000 Perioden pro Umdrehung

Der Rotationsgeber ist auf der Motorwelle montiert. Die Spindelsteigung beträgt 10 mm.
Die resultierende Bewegung pro Masstabsperiode ist hier abhängig von der Getriebeübersetzung und der Spindelsteigung bzw. ob der Rotationsgebers vor oder nach dem Getriebe montiert ist.

Geber auf Motorwelle kein Getriebe:
setclperiod = 10 mm / 1000 Perioden = 0.01

Getriebe 120 : 1, Geber auf Motorwelle:
Pitch = 1 / 120 = 0.00833
setclperiod = 0.00833 / 1000 = 0.00000833

Getriebe 120 : 1, Geber am Getriebe
Pitch = 1 / 120 = 0.00833
Resultierende Bewegung und Messperiode sind direkt gekoppelt:
setclperiod = 1 / 1000 = 0.001

Beispiel: Rundtisch mit Drehgeber in der Einheit Grad

Der Drehgeber liefert 18000 Perioden pro Umdrehung und ist auf der Motorwelle montiert. Getriebe 120:1.
Für die Darstellung in Grad muss zunächst der Parameter ***setpitch*** bestimmt werden.

setpitch = Resultierende Bewegung / Motorumdrehung
= 360° / 120 Umdrehungen = 3

Mit der Angabe 360 0 0 ***move*** wird so bei Achse-1 eine Drehung um 360° erzeugt.
Danach erfolgt die Einstellung für ***setclperiod:***

Eine Motorumdrehung erzeugt 18000 Signalperioden.
Resultierender Winkel = 3 Grad (360 / 120)
Winkel pro Signalperiode = 3 Grad / 18000 = 0.0001666

setclperiod = 0.0001666

getclperiod

Beschreibung:

Der Befehl ***getclperiod*** liest die Einstellungen von ***setclperiod*** zurück.

Syntax:

[Achse] ***getclperiod***

Parameter	Wertebereich
[Achse]	1, 2, 3

Rückmeldung

[Wert]

	Wertebereich	Einheit
[Wert]	0.0000001-1.999999	unit

Beispiel:

1 getclperiod

Rückmeldung:

0.000000100

setclfactor

Beschreibung:

Der Befehl ***setclfactor*** adaptiert die spezifizierte Achse an die Teilungsperiode eines digitalen Längenmesssystems.
Unter digitalem Messsystem
Der Wert von ***setclfactor*** enspricht der Anzahl der Impulse pro Motorumdrehung.

Für analoge Messsysteme wird die gleiche Anpassung mit dem Befehl ***setclperiod*** vorgenommen.

Durch Ändern des Vorzeichens kann die **Zählrichtung** des Messeingangs an die Drehrichtung des Motors angepasst werden.

Syntax:

[Zählrichtung] [Impulse] [Achse] ***setclfactor***

Parameter	Wertebereich	Einheit
Zählrichtung	+,-	Vorzeichen
Impulse	1-5000	
Achse	1, 2, 3	

Partnerbefehl:

getclfactor

Beispiele:

- 500 3 setclfactor

Beispiele für die Berechnung von ***setclfactor*** auf der folgenden Seite.

getclfactor

Beschreibung:

Der Befehl ***getclfactor*** liest die Einstellungen von ***setclfactor*** zurück.

Syntax:

[Achse] ***getclfactor***

Parameter	Wertebereich
Achse	-1, 1, 2, 3

Rückmeldung

Wert

	Wertebereich	Einheit
Wert	0-5000	Impulse/Umdrehung

Beispiel:

1 getclfactor

Rückmeldung:

500

setclpara

Beschreibung:

Mit dem Kommando ***setclpara*** wird der Positionsregler für den Closed Loop Betrieb eingestellt.

speicherbar

Die folgenden Parameter können am Regler verändert werden:

- Einstellung der P-I-D Parameter.
- Einstellen einer Achsabschaltung bei Positionsabweichung.
- Begrenzung der vom I-Regler erzeugten Positionierschwindigkeit beim Nachregeln auf die Zielposition
- Modifikation des I-Anteils während der Positionierung, abhängig von der Position und der Geschwindigkeit.

Syntax:

[**P**] [**I**] [**D**] [**16383**] [**SP1**] [**SP2**] [**dpos**] [**ivel**] [**cutoff**] [**SP3**] [**np**] [**Achse**] ***setclpara***

Der Befehl besteht aus einer Zeile von maximal 10 Parametereinträgen, die mit Leerzeichen voneinander getrennt sind. Die Eingabe wird mit der Anzahl der angegebenen Parametern (np), der Achsenzuordnung (Achse) sowie dem Kommando selbst abgeschlossen.
Bei der Änderung eines einzelnen Parametereintrages müssen alle Parameter die diesem Wert vorangestellt ebenfalls übertragen werden.

Parameter	Funktionsbeschreibung
[P]	P-Anteil des Positionsreglers. (Nur für Linearmotorantriebe)
[I]	I-Anteil des Positionsreglers.
[D]	D-Anteil des Positionsreglers. (Nur für Linearmotorantriebe)
[16383]	Wert für den Einfluss des I-Anteils auf die Positionskorrektur. Die Werkseinstellung ist der maximale Wert und sollte nicht verändert werden.
[SP1]	Boost-Faktor (Nur für Linearmotorantriebe)
[SP2]	Lastwinkelvorgabe (Nur für Linearmotorantriebe)
[dpos]	Maximal erlaubter Positionsfehler. Bei Überschreitung erfolgt die Abschaltung der Achse. Maschinenfehler 13 wird gesetzt. Mit der Einstellung = 0 ist die Funktion abgeschaltet.

Parameter	Funktionsbeschreibung
[ivel]	Diese Funktion begrenzt die vom I-Anteil erzeugte Nachregelgeschwindigkeit bei einer Positionskorrektur. Der Eingriff wirkt im Stillstand und während der Fahrt. Mit der Einstellung = 0, ist die Funktion abgeschaltet und die Nachregelgeschwindigkeit maximal.
[cutoff]	Mit Hilfe dieser Funktion kann der I-Anteil des Reglers so eingestellt werden, dass ein möglichst geringes Einschwingverhalten in die Zielposition erreicht wird. Gleichzeitig bleibt die Laufruhe erhalten. Während der Fahrt erfolgt eine dynamische Reduzierung des I-Anteils abhängig von der Sollgeschwindigkeit. Bei Erreichen der eingestellten cutoff Geschwindigkeit ist der Einfluss des I-Anteils dann auf ein Minimum reduziert. Im Stillstand bzw. unterhalb der cutoff Geschwindigkeit ist der Einfluss des I-Reglers wieder maximal. Mit der Einstellung = 0 ist die Funktion abgeschaltet.
[SP3]	Keine Funktion
[np]	Anzahl der Parameter die mit dem Kommando übergeben werden.
[Achse]	Achsenzuordnung

Parameter	Wertebereich	Default	Einheit
P		0	-
I		10	1/s
D		0	s
16383		16383 (maximaler Wert)	
SP1		0	
SP2		0	
dpos		0 0 = abgeschaltet	mm
ivel		0 0 = abgeschaltet	mm/s
cutoff		2 0 = abgeschaltet	mm/s
SP3	0	0	
np			
Achse	1, 2, 3		

Partnerbefehle:

getclpara, setcloop, setclperiod

Beispiele:

0_20_2_1_ setclpara

Hier wurde für Achse-1 die Einstellung am I-Regler verändert, alle weiteren Parameter blieben unverändert.

0_15_0_16383_0_0_1_2_1_8_3_setclpara

Hier wurden dem Positionsregler der Achse-3 insgesamt 8 Parameter übergeben.

getclpara

Beschreibung:

Das Kommando ***getclpara*** liest die Einstellung des Positionsreglers der Achse. Es werden immer alle 10 Parameter des Reglers zurückgeliefert.

Syntax:

[Achse] ***getclpara***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[**P**] [**I**] [**D**] [**16383**] [**SP1**] [**SP2**] [**dpos**] [**ivel**] [**cutoff**] [**SP3**]

Beispiel:

1 getclpara

Rückmeldung:

0.000000 10.000000 0.000000 16383.000000 0.000000 0.000000 0.000000 2.000000 1.000000 0.000000

Geschwindigkeit und Beschleunigung

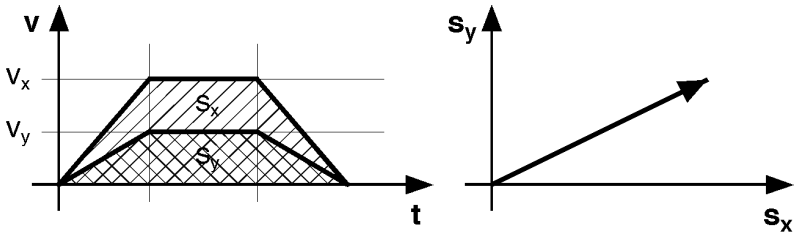
setvel (sv)

Beschreibung

speicherbar

Das Kommando ***setvel*** bestimmt die Geschwindigkeit **v**x für die programmierten Positionierung.
Beim der programmierten Positionierung werden alle aktiven Achsen gleichzeitig gestartet und erreichen zur gleichen Zeit das Ziel. Die Geschwindigkeit bezieht sich deshalb auf die Achse, die den längsten Verfahrweg **s**x zurücklegen muss. Siehe Diagramm.
Die Maximalgeschwindigkeit der anderen Achsen ergibt sich aus dem Verhältnis ihrer Verfahrstrecken **s**y zur Verfahrstreccke **s**x

$$v_y \quad \frac{s_y}{s_x} \times v_x$$



Syntax:

[Geschwindigkeit] ***setvel***

	Wertebereich
min. Geschwindigkeit	15,26 nm/s
max. Geschwindigkeit	60 U/s x Spindelsteigung

	Einheit
[Geschwindigkeit]	unit

unit = Eingestellte Einheit

Weitere Geschwindigkeitsbefehle

setcalvel / setrefvel / setjoystickspeed

Beispiel:

100 sv

unit = mm
Pitch: 4mm
Die Maximalgeschwindigkeit der schnellsten Achse für die programmierte Positionierung, wird auf 100 mm/s begrenzt.

getvel (gv)

Beschreibung:

Das Kommando ***getvel (gv)*** liefert die eingestellte Geschwindigkeit für die programmierte Positionierung zurück.

Synax:

getvel

Rückmeldung:

[Geschwindigkeit]

	Einheit
[Geschwindigkeit]	unit

Beispiel:

gv

Rückmeldung:
180.000000

setaccel (sa)

Beschreibung:


speicherbar

Das Kommando ***setaccel*** bestimmt die Beschleunigung der programmierten Positionierung.
Da die Achsen linear interpoliert werden, bezieht sich diese Angabe auf die Achse, die den längsten Verfahrweg zurücklegen muss.
Siehe “setvel (sv)” auf Seite 76.

Die Maximalbeschleunigung der restlichen Achsen ergibt sich aus deren Maximalgeschwindigkeiten.

Die Beschleunigungsrampe und die Bremsrampe sind identisch.

Syntax:

[Beschleunigung] ***setaccel***

	Wertebereich bei unit = mm	**Einheit**
[Beschleunigung]	0 - 2400	unit/s²

Verwandte Befehle:

getaccel / setmanaccel

Beispiel:

500 sa

getaccel (ga)

Beschreibung:

Der Befehl ***getaccel*** liest die eingestellte Beschleunigung der Achse zurück.

Syntax:

getaccel

Rückmeldung:

[Beschleunigung]

	Wertebereich bei unit = mm	Einheit
[Beschleunigung]	0 - 2400	unit/s²

Beispiel:

ga

Rückmeldung:
2400000.000000 (wenn unit = µm)

setrefvel

Beschreibung:

Der Befehl ***setrefvel*** bestimmt die Geschwindigkeit mit der die Positionierung zur Referenzmarke eines Messsystems durchführt wird.

speicherbar

HINWEIS!

Als Referenzfahrt wird die Fahrt zur Referenzmarke eines Messsystems bezeichnet.

Syntax:

[Wert] [Index] ***setrefvel***

	Wertebereich	Einheit
[Wert]	0 - 45	mm/s
[Index]	1	

Verwandter Befehl

getrefvel / setvel

Beispiel:

0.5 1 setrefvel

Die Geschwindigkeit der Referenzfahrt wird hier auf 0.5mm/s eingestellt.

getrefvel

Beschreibung:

Der Befehl ***getrefvel*** liest die eingestellte Geschwindigkeit mit der die Steuerung die Referenzfahrt durchführt.

Aus Gründen der Kompatibilität zu den Steuerungen MC-2000 bzw. mc-compact liefert ***getrefvel*** zwei Parameter in zwei Zeilen.

Syntax:

getrefvel

Rückmeldung:

[Wert]
[NF]

NF: Keine Funktion

	Wertebereich	Einheit
[Wert]	0 - 45	mm/s

Beispiel:

getrefvel

Rückmeldung:

10.000000
0.050000

setmanaccel

Beschreibung:

Das Kommando ***setmanaccel*** bestimmt die Beschleunigung der Achse für den manuellen Betrieb.
Dies gilt für den Betrieb mit Joystick oder Handrad.

Syntax:

[Beschleunigung] ***setmanaccel***

Parameter	Wertebereich	Einheit
[Beschleunigung]	0 - 2400	mm/s²

Verwandte Befehle

getmanaccel / setaccel

Beispiel:

100 setmanaccel

Die manuelle Positionierung wird auf den Wert 100 mm/s² eingestellt.

getmanaccel

Beschreibung:

Das Kommando ***getmanaccel*** liest die Einstellung der Beschleungung für den manuellen Betrieb.

Syntax:

getmanaccel

Rückmeldung:

[Wert]

Parameter	Wertebereich	Einheit
[Wert]	0 - 2400	mm/s²

Beispiel:

getmanaccel

Rückmeldung:
2400.000000

Positionierkommandos

move (m)

Beschreibung:

Das Kommando ***move*** positioniert die Achsen zu den angegebenen Koordinaten.
Unter Berücksichtigung der eingestellten Geschwindigkeit, Beschleunigung und den vorgegebenen Verfahrgrenzen errechnet die Steuerung daraus ein Fahrprofil für alle Achsen. Die Achsen werden gleichzeitig gestartet und erreichen gleichzeitig ihre Zielkoordinate.

Bezugspunkt ist der Koordinatennullpunkt, der entweder bei der cal- Endschalterfahrt oder durch den Befehl ***setpos*** festgelegt wurde.

Das Kommando ***status*** liefert die Rückmeldung über den aktuellen Zustand der Positionierung.

Mit ***Ctrl+c*** oder ***abort*** kann die Positionierung abgebrochen werden.

Syntax:

[Achse-1] [Achse-2] [Achse-3] ***move***

	Wertebereich bei unit = m m	**Einheit**
[Achse -1]	+/- 16383	unit
[Achse -2]	+/- 16383	unit
[Achse -3]	+/- 16383	unit

Die Anzahl der Parameter ist abhängig von der Einstellung der Dimension.

setdim	Erwartete Anzahl der Achskoordinaten
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

Verwandter Befehl:

rmove, speed

Beispiele:

Dimension = 3
12.5 20 0.0001 m
Absolute Positionierung aller 3 Achsen.

Dimension = 1
12.5 m
Absolute Positionierung von Achse-1

Dimension = 2
12.5 20 m
Absolute Positionierung von Achse-1 und Achse-2

rmove (r)

Beschreibung:

Der Befehl ***rmove*** positioniert die Achsen relativ zu den aktuellen Koordinaten.
Unter Berücksichtigung der eingestellten Geschwindigkeit, Beschleunigung und den vorgegebenen Verfahrgrenzen errechnet die Steuerung daraus ein Fahrprofil für alle Achsen. Die Achsen werden gleichzeitig gestartet und erreichen gleichzeitig ihre Zielkoordinate.

Das Kommando ***status*** liefert die Rückmeldung über den aktuellen Zustand der Positionierung.

Mit ***Ctrl+c*** oder ***abort*** kann die Positionierung abgebrochen werden.

Syntax:

[Achse-1] [Achse-2] [Achse-3] ***rmove***

Die Anzahl der Parameter ist abhängig von der Einstellung der Dimension. Siehe Befehl ***setdim***

Parameter	Wertebereich bei unit = m m	Einheit
Achse -1	+/- 16383	unit
Achse -2	+/- 16383	unit
Achse -3	+/- 16383	unit

Verwandter Befehl:

move, speed

Beispiele:

Dimension = 3
12.5 20 0.0001 r
Relative Positionierung aller 3 Achsen.

Dimension = 1
12.5 rmove
Relative Positionierung von Achse-1

Dimension = 2
12.5 20 r
Relative Positionierung von Achse-1 und Achse-2

refmove

Beschreibung:

Das Kommando ***refmove*** positioniert alle aktiven Achsen zur Referenzmarkierung des Längenmesssystems.
Im Kommando wird die Richtung sowie die Strecke vorgegeben innerhalb der die Steuerung die Referenzmarke sucht.

Die Geschwindigkeit der Referenzfahrt wird mit dem Befehl ***setrefvel*** festgelegt.
Hat die Steuerung während der Fahrt eine Referenzmarke erkannt, wird die Positionierung mit der eingestellen Systembeschleunigung beendet.
Wird keine Referenzmarke gefunden, beendet die Steuerung die Fahrt nach erreichen der vorgegebenen Strecke. Das Ergebnis der Referenzfahrt wird mit dem Befehl ***getrefst*** gelesen. Der Abbruch der Referenzfahrt erfolgt mit dem Kommando ***Ctrl+c*** .


HINWEIS!

Für die Funktion müssen die Achsen wie folgt eingestellt sein:

- ***setcloop*** = 1
- ***setaxis*** = 1
- ***setref*** = 0 oder 1

Soll nur eine einzelne Achse eine Referenzfahrt ausführen, müssen die anderen Achsen deaktiviert werden oder *setref* = 2 eingestellt sein.

refmove **sperrt den Interpreter. Während der Ausführung können deshalb keine weiteren Kommandos verarbeitet werden.**

Syntax:

[Strecke] ***refmove***

	Wertebereich	Einheit
[Strecke]	+/-1000	Umdrehungen

Verwandte Befehle

setref, setrefvel, getrefst

Beispiel:

100 refmove

Die Steuerung positioniert auf die Referenzmarke des Messsystems. Alle aktiven Achsen werden dabei maximal 100 Umdrehungen in positive Richtung bewegt.

-7 refmove

Die Steuerung positioniert auf die Referenzmarke des Messsystems. Alle aktiven Achsen werden dabei maximal 7 Umdrehungen in negative Richtung bewegt.

calibrate (cal)

Beschreibung:

Das Kommando ***cal*** löst die Endschalterfahrt zum cal-Endschalter aus, hierbei werden die dafür aktivierten Achsen in negative Richtung positioniert bis der cal-Schalter betätigt ist. Die Steuerung positioniert danach wieder in Richtung positiver Positionswerte vor den Endschalter.
Mit dem Befehl ***setcalswdist*** kann eine zusätzliche Distanz zum Endschalter definiert werden.
Die Koordinate am Ende der cal-Endschalterfahrt wird als das untere Limit erfasst und kann danach nicht mehr unterschritten werden.
Mit der Einstellung ***setaxis*** wird die Wirkung der Endschalterfahrt auf die Limits, die Koordinaten und die aktuelle Position beeinflusst. Siehe “setaxis” auf Seite 40.

Mit ***Ctrl+c*** wird die Endschalterfahrt sofort abgebrochen und der Nullpunkt an dieser Stelle gesetzt.

Das untere Limit, sowie der Koordinatennullpunkt werden aus Sicherheitsgründen nicht gespeichert.

HINWEIS!

Das Kommando *cal* sperrt den Interpreter, das heisst die Steuerung kann weiterhin Kommandos empfangen aber diese nicht ausführen.

Die Vorgehensweise um automatisch eine Statusrückmeldung bei Beedigung der cal Prozedur zu bekommen, wird in der Einfühung zu Venus-1 beschrieben.

Syntax:

calibrate oder ***cal***

Beispiel:

cal

rangemeasure (rm)

Beschreibung:

Der Befehl ***rm*** löst die Endschalterfahrt zum rm-Endschalter aus, hierbei werden die dafür aktivierten Achsen in positive Richtung positioniert bis der rm-Schalter betätigt ist. Die Steuerung positioniert danach wieder in Richtung negativer Positionswerte vor den Endschalter.
Mit dem Befehl ***setcalswdist*** kann eine weitere Distanz zum Endschalter definiert werden.
Die Koordinate am Ende der rm-Endschalterfahrt wird als das obere Limit erfasst und kann danach nicht mehr überschritten werden.

HINWEIS!

Für die korrekte Ermittlung des maximalen Verfahrbereichs muss zuvor der Befehl cal ausgeführt werden
Mit der Einstellung ***setaxis*** wird die Wirkung der Endschalterfahrt auf die Limits, die Koordinaten und die aktuelle Position beeinflusst. Siehe “setaxis” auf Seite 40.
Mit ***Ctrl+c*** wird die Endschalterfahrt abgebrochen.

Das obere Limit wird aus Sicherheitsgründen nicht gespeichert.

HINWEIS!

Das Kommando *rm* sperrt den Interpreter, die Steuerung kann zwar weiterhin Kommandos empfangen aber diese nicht ausführen.
Die Kommandos werden erst Beendigung der Endschalterfahrt in der Reihenfolge ihres Eintrags ausgeführt.

Syntax:

rangemeasure oder ***rm***

Beispiel:

rm
Abhängig von der Einstellung ***setaxis*** wird für alle Achsen die rm-Endschalterfahrt ausgeführt.

Abbruchkommandos

Ctrl+c

Beschreibung:

Mit dem Kommando ***Ctrl+c*** wird der momentan vom Interpreter ausgeführte Befehl abgebrochen. Kommandos die sich im Daten-FIFO befinden werden dabei nicht gelöscht.
Eine durch ***Ctrl+c*** abgebrochene Positionierung wird kontrolliert mit der eingestellten Systembeschleunigung beendet. ***Ctrl+c*** durchläuft nicht das Daten-FIFO und kann damit durch kein anderes Kommando gesperrt werden.

Ctrl+c kann auch dazu verwendet werden die Endschalterprozedur abzubrechen und dabei das untere bzw. obere Verfahrlimit festzulegen.

Das Kommando darf nicht für eine schnelle Abfolge von Start und Abbruchprozeduren verwendet werden. Hierfür sei der Befehl *abort* oder der Befehl *speed* und *stop-speed* empfohlen.

Syntax:

Ctrl+c (ASCII 3)

Verwandter Befehl:

abort

Beispiel:

Ctrl+c
Der aktuelle ausgeführte Befehl wird abgebrochen.

abort

Beschreibung:

Mit ***abort*** wird der momentan ausgeführte Befehl abgebrochen.
abort durchläuft im Unterschied zu ***Ctrl+c*** das Daten-FIFO und kann durch ein blockierendes Kommando verzögert werden.

Beispiel:
100 0 0 move
ge
abort

ge blockiert ***abort*** und wird erst ausgeführt nachdem der ***100 0 0 move*** abgeschlossen ist.

Syntax:

abort

Verwandter Befehl:

Ctrl+c

Siehe “Ctrl+c” auf Seite 95.

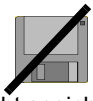
Beispiel:

abort

Arbeitsbereich und Bezugspunkt

align

Beschreibung:


nicht speicherbar

Der Befehl ***align*** ermöglicht eine beliebige Drehung des von Achse-1 und Achse-2 gebildeten Koordinatensystems um einen Nullpunkt.

Vorgehensweise:

1. Verfahrbereichsgrenzen mit cal / rm festlegen.
2. Dimension auf 2 setzen (2 setdim)
3. Nullpunkt festlegen (0 0 setpos)
4. Lage der neuen X- oder Y-Achse mit ***align*** festlegen.


HINWEIS!

Vor der align Prozedur muss die Dimension auf 2 eingestellt sein.

Die Positionierbefehle ***move*** und ***rmove*** beziehen sich nach der der Drehung auf das neue Koordinatensystem.
Das Koordinatensystem der 3. Achse wird nicht verändert.

Die Verfahrbereichsgrenzen werden auch nach der Drehung überwacht und können nicht überschritten werden.
Der Befehl *getlimit* liefert die Werte des ursprünglichen Koordinatensystems.

Mit dem Befehl ***ico*** wird das Koordinatensystem wieder zurückgesetzt.

Syntax:

[0] [0] [OrgX] [OrgY] [XY] ***align***

OrgX und OrgY sind die Orginalkoordinaten, durch den die neue X oder Y-Achse verläuft.

	Beschreibung
[OrgX]	Orignalkoordinate X
[OrgY]	Orginal koordinate Y
[XY]	Koordinatenzuordnung X- oder Y-Achse 1 = X-Achse 2 = Y-Achse

	Wertebereich bei unit = m m	**Einheit**
[OrgX]	+/- 16383	unit
[OrgY]	+/- 16383	unit
[XY]	1, 2	

Rückmeldung:

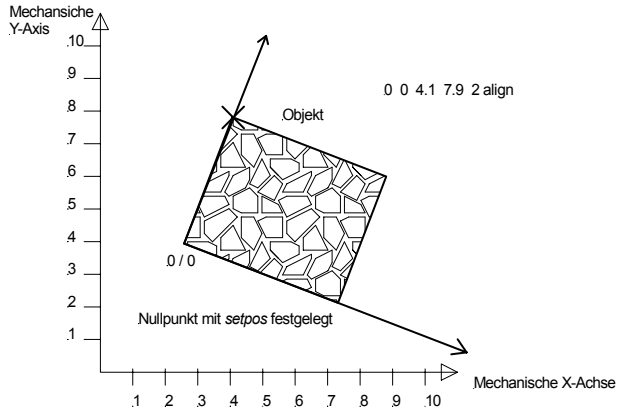
Eine Rückmeldung ob das Koordinatensystem gedreht wurde erhält man mit dem Befehl ***getico***
.

Beispiel:

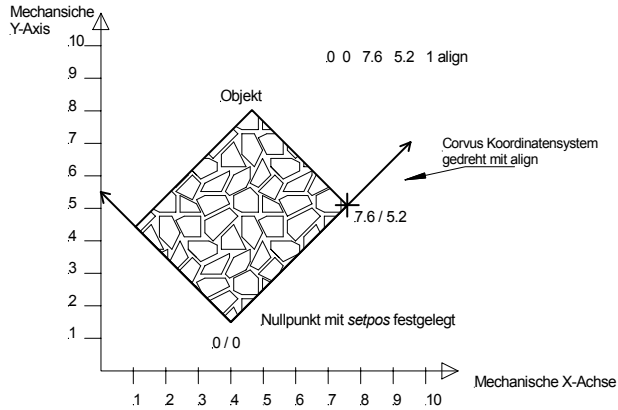
0 0 10 10 1 align
Die Gerade zwischen den Punkte 0 / 0 und 10 / 10 wird zur neuen X-Achse

0 0 10 1 2
Die Gerade zwischen den Punkten 0/0 und 10 / 1 wird zur Y-Achse.
Siehe die Darstellungen auf der folgenden Seite:

Das untenstehende Bild zeigt ein Objekt wie es beispielweise unter einem Mikroskop zu finden ist. Die Ursprünglichen Koordinaten werden durch den motorischen Tisch bestimmt. Mit ***align*** wird das Koordinatensystem auf das Objekt bezogen, wobei hier die Lage der neuen Y-Achse mit dem Befehl übergeben wird.



Beim Bild unten werden die Koordinaten der neuen X-Achse mit dem ***align*** Kommando übergeben.



ico

Beschreibung:

Der Befehl ***ico*** setzt ein gedrehtes Koordinatensystem wieder zurück.
Siehe “align” auf Seite 98.

Syntax:

ico

Beispiel:

ico

getico

Beschreibung:

Der Befehl ***getico*** liest den Status des Koordinatensystems. Damit kann überprüft werden ob das Koordinatensystem gedreht ist oder nicht.

Syntax:

getico

Rückmeldung

[Index]

[Index]	Beschreibung
0	Koordinatensystem gedreht
1	Koordinatensystem ist nicht gedreht

	Wertebereich
[Index]	0, 1

Beispiel:

getico

setpos

Beschreibung:


nicht speicherbar

Mit dem Kommando ***setpos*** wird der Koordinatenursprung an einer beliebigen Koordinate innerhalb des Arbeitsbereichs festgelegt.
Für Sonderfälle kann damit eine relative Nullpunktsverschiebung durchgeführt werden.
Falls die Koordinaten der Limits schon erfasst sind, werden diese bezogen auf diesen Nullpunkt neu kalkuliert.

Syntax:

[Achse-1] [Achse-2] [Achse-3] ***setpos***

	Wertebereich Beispiel: mm	**Einheit**
[Achse-1]	+/-16383	unit
[Achse-2]	+/-16383	unit
[Achse-3]	+/-16383	unit

Beispiele:

0 0 0 setpos
Die aktuelle Position wird bei allen Achsen der Koordinatennullpunkt

10 10 10 setpos / unit = mm
Der Nullpunkt wird hier bezogen auf den aktuellen Nullpunkt für alle Achsen um 10 mm in positive Richtung verschoben.
Hatte der aktuelle Nullpunkt die Koordinaten 0 0 0 liefert das Kommando ***pos*** nach der Verschiebung die Positionskoordinaten -10 -10 -10.

setlimit

Beschreibung:

Mit dem Kommando ***setlimit*** ist es möglich sogenannte Softwarelimits festzulegen, die den Verfahrbereich begrenzen. Der maximale Verfahrbereich innerhalb der Hardlimits bzw. Endschalter muss zuvor durch die cal- und rm-Endschalterfahrt bestimmt sein.
Im Normalbetrieb werden die Limits nicht mehr überschritten. Eine Positionierung, die den Verfahrbereich überschreiten würde, wird mit der eingestellten Systembeschleunigung abgebremst und kommt auf dem Limit zum Stillstand.

HINWEIS!

Bedingungen für die Festlegung von Softwarelimits

- Der maximale Verfahrbereich muss vor Ausführung des Befehls durch die cal- und rm-Endschalterfahrt bestimmt sein.
- Der Wert der unteren Grenze muß kleiner sein als der Wert der oberen Grenze
- Die aktuelle Position muss sich innerhalb des vorgegebenen Limits befinden.

Syntax:

[-A1] [-A2] [-A3] [A1+] [A2+] [A3+] ***setlimit***

Die Anzahl der übergebenen Achsparametern ist abhängig von der Einstellung ***setdim***

	Beschreibung
[-A1]	unteres Limit Achse-1
[-A2]	unteres Limit Achse-2
[-A3]	unteres Limit Achse-3

	Beschreibung
[A1+]	oberes Limit Achse-1
[A2+]	oberes Limit Achse-2
[A3+]	oberes Limit Achse-3

	Wertebereich	**Einheit**
[-A1] [-A2] [-A3]	-16383	unit
[A1+] [A2+] [A3+]	+16383	unit

Partnerbefehl:

getlimit

Beispiel:

getdim = 3

0 0 0 12 25 30 setlimit

Damit ergeben sich folgende Verfahrbereichsgrenzen:

Achse-1: 0 bis 12
Achse-2: 0 bis 25
Achse-3: 0 bis 30

getlimit

Beschreibung:

Der Befehl ***getlimit*** ermittelt die gültigen Verfahrgrenzen aller Achsen.
Abhängig von der Einstellung ***setdim*** werden die Werte in 1, 2 oder 3 Zeilen zurückgeliefert.
Wurde kein Limit festgelegt, meldet die Steuerung folgenden Wert für das untere und obere Limit:
-16383.000000 16383.000000

I

Syntax:

getlimit

Rückmeldung:

		Einheit	Achse
[unteres Limit]	[oberes Limit]	unit	1
[unteres Limit]	[oberes Limit]	unit	2
[unteres Limit]	[oberes Limit]	unit	3

Beispiel:

getlimit

Unit: mm

0.000000 7.723750
0.000000 7.723750
-16383.000000 16383.000000

Statusabfragen

geterror (ge)

Beschreibung:

Mit dem Kommando ***geterror*** wird die Steuerung auf allgemeine Systemfehler überprüft. Es wird immer der zuletzt aufgetretene Fehler angezeigt.
Nachdem das Kommando ausgeführt wurde, ist der Fehlerspeicher wieder gelöscht.

Syntax:

geterror

Rückmeldung:

[Fehlercode]

Fehlercode	Beschreibung
1....4	Interner Fehler
1001	Falscher Parametertyp
1002	Zu wenige Parameter für den Befehl auf dem Stack
1003	Wertebereich des Parameters ist überschritten
1004	Verfahrbereich sollte überschritten werden
1008	Zu wenige Parameter für den Befehl auf dem Stack
1009	Zu wenig Platz auf dem Stack
1010	Kein Speicher mehr frei
1015	Parameter ausserhalb des Verfahrbereichs
2000	Unbekanntes Kommando

Beispiel:

ge

status (st)

Beschreibung:

Mit dem Kommando ***status*** wird der augenblickliche Betriebszustand der Achse abgefragt.
Die dezimale Rückmeldung ist bitcodiert und reflektiert die untenstehenden Betriebszustände.
Für eine korrekte Auswertung der Zustände ist es notwendig die Statusabfrage auszumaskieren.

Binär	Dezimal	Funktion
D0	1	Status der Befehlsausführung
D1	2	Status Joystick od. Handrad
D2	4	Status Taster A
D3	8	Maschinenfehler
D4	16	Status Speed Modus
D5	32	Status In-Window
D6	64	Status Verfahrbereichsbegrenzung
D7	128	Motorfreigabe od. Überstrom

D0:

0	Interpreter ist bereit einen neuen Befehl auszuführen.
1	Interpreter führt einen Befehl aus.

D1: Kommando *joystick*

0	Manueller Betrieb nicht aktiv
1	Manueller Betrieb aktiv

D2:Taster A an Corvus Frontplatte

0	Taster A nicht betätigt
1	Taster A betätigt

D3:

0	Keine Funktion.
1	Keine Funktion.

D4: Kommando *speed*

0	Speed Funktion aktiv.
1	Speed Funktion aus.

D5: Kommando *setclwindow*

0	Position ausserhalb der Zielfenster.
1	Position aller aktiven Achsen ist im Zielfenster.

D6: Kommando *setinfunc*

0	Das externe Gerät liefert kein Eingangssignal.
1	Eingangssignal von externem Gerät erkannt.

D7:

0	Motorendstufen aktiv.
1	Externe Abschaltung der Motorendstufen.

Syntax:

status oder ***st***

Rückmeldung

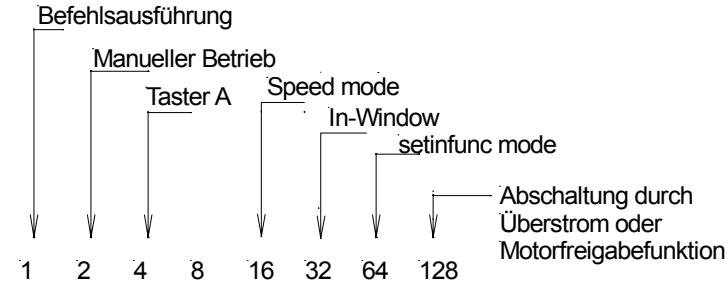
[Wert]

Beispiel:

status
Rückmeldung: **2**

Die Steuerung ist bereit ein neues Kommando auszuführen.
Der Joystick ist eingeschaltet.

Weitere Beispiele auf der folgenden Seite:



D0	D1	D2	D3	D4	D5	D6	D7	Codierte Statusrückmeldung
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	2
0	1	0	0	0	0	1	0	66
0	1	1	0	0	0	0	0	6
0	1	1	0	0	0	0	1	134

Beispiele

pos (p)

Beschreibung:

Das Kommando ***pos*** oder ***p*** liefert die aktuelle Position der Achsen bezogen auf den Koordinatennullpunkt.
Die Einstellung ***setdim*** bestimmt welche Achsen angezeigt werden.
Mit dem Befehl ***setselpos*** wird festgelegt ob die intern kalulierte Position oder die von einem externen Messsystem erfasste Position geliefert wird.

Syntax:

pos oder ***p***

Rückmeldung:

[Pos-1] [Pos-2] [Pos-3]

Werte	Beschreibung	Einheit
[Pos-1]	Position der Achse-1	unit
[Pos-2]	Position der Achse-2	unit
[Pos-3]	Position der Achse-3	unit

Die Darstellung von Pos-2 und Pos-3 ist abhängig von der Einstellung ***setdim***.

Beispiel:

pos

Rückmeldung: unit= mm / Dimension = 3

1.00000 19.00000 0.00000

identify

Beschreibung:

Der Befehl ***identify*** liefert verschiedene Informationen zur Hardware der Steuerung.

Dies sind:
- Gerätebezeichnung
- Firmware Version
- Hardware Version
- Interner Schalter
- Hex kodierte Dip-Schaltereinstellung

Syntax:

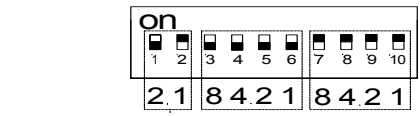
identify

Rückmeldung:

[Model] [HW-Rev] [SW-Rev] [Board-SW] [FP-SW]

Werte	Beschreibung
[Modell]	Gerätebezeichnung
[HW-Rev]	Hardware Version
[SW-Rev]	Software Version
[Board-SW]	Interner Schalter
[FP-SW]	Einstellung des Konfigurationsschalters. Die Schalterstellung ist hexadezimal kodiert.

Kodierung der Schalterstellung in FP-SW



Verwandter Befehl:

version

Beispiel:

identify

Rückmeldung: Corvus 1 312 1 10F

version

Beschreibung:

Der Befehl ***version*** liefert die Firmwareversion der Steuerung.

Syntax:

version

Rückmeldung

[Versionsnummer]

Beispiel:

version

Rückgabe:

3.23

getswst

Beschreibung:

Das Kommando ***getswst*** zeigt den Schaltzustand der Endschaltereingänge an.

[Achse] ***getswst***

	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[cal]] [rm]

cal = 0	cal-Endschalter nicht betätigt
cal = 1	cal-Endschalter betätigt

rm = 0	rm-Endschalter nicht betätigt
rm = 1	rm-Endschalter betätigt

Beispiele:

3 getswst
Rückmeldung:
0 0
Es ist kein Endschalter von Achse-3 betätigt.

-1 getswst
Rückmeldung
0 0 1 0 0 0
Der cal-Endschalter der Achse-2 ist betätigt.

getrefst

Beschreibung:

Das Kommando ***getrefst*** liefert das Ergebnis der Referenzfahrt als dezimale Zahl.
In der binären Darstellung D0 bis D2 können daraus folgende Zustände entnommen werden:

- D0 (1) = Status der Referenzmarke
- D1 (2) = Status der Endschalter
- D2 (4) = Pegel des Referenzsignals

(Siehe “setref” auf Seite 62.)

Kodierung	Wertebereich
D0 = 0	Referenzmarke gefunden
D0 = 1	Referenzmarke nicht gefunden
D1 = 0	Endschalter nicht betätigt
D1 = 1	Endschalter betätigt
D2 = 0	Positiver Signalpegel
D2 = 1	Nullpegel

Syntax:

[Achse] ***getrefst***

	Wertebereich
[Achse]	-1, 1, 2, 3

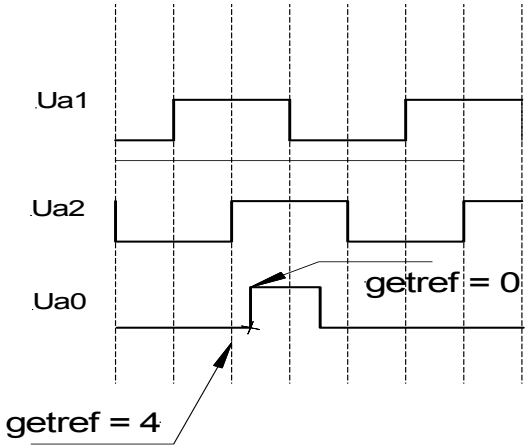
Rückmeldung:

[Dezimaler Wert]

Typische Werte
0, 1, 3, 4, 5

Beispiele:

Wert	Beschreibung
0	Referenzmarke erreicht, postiver Pegel
1	Referenzmarke nicht erreicht
3	Referenzmarke nicht gefunden, Endschalter betätigt
4	Referenzmarke erreicht, Nullpegel
5	Referenzmarke nicht erreicht, Nullpegel



Ua1 / Ua2 Quadraturzählersignal
Ua0 = Referenzsignal
Einstellung ***setref*** = 1

Beispiel:

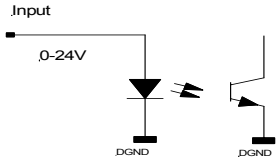
2 getrefst

Input / Output Funktionen

getin

Beschreibung:

Das Kommando ***getin*** liest den Status der digitalen Eingänge Din-1, Din-2, Din-3. Die Rückmeldung ist bitcodiert.



Syntax:

getin

Rückmeldung:

[bitcodierter Wert]

	Wertebereich
[bitcodiert	0-7

Beispiel:

getin

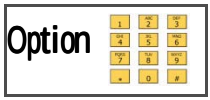
[bitcodiert	**Eingang Din-1**	**Eingang Din-2**	**Eingang Din-3**
0*	0	0	0
1**	1	0	0
2	0	1	0
4	0	0	1

*0 : Eingangsspannung 0-2V

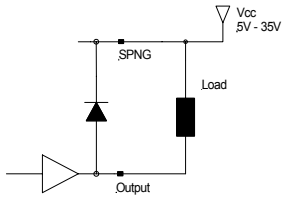
**1: Eingangsspannung 3-24V

setout

Beschreibung:



Das Kommando ***setout*** schaltet die Open Collector Ausgänge Dout1- Dout3.
Die Eingabe erfolgt als bitcodierte Zahl.
Im ON Zustand wird der Ausgangstransistor gegen DGND geschaltet.



Syntax:

[bitcodiert] ***setout***

[bitcodiert]	Dout 1	Dout 2	Dout 3
0	OFF	OFF	OFF
1	ON	OFF	OFF
2	OFF	ON	OFF
3	ON	ON	OFF
4	OFF	OFF	ON
5	ON	OFF	ON
6	OFF	ON	ON
7	ON	ON	ON

Partnerbefehl:

getout

Beispiel:

7 setout

getout

Beschreibung:

Das Kommando g***etout*** liest den mit ***setout*** eingestellten Ausgabewert zurück.

Syntax:

getout

Rückmeldung:

[bitcodiert]

Beispiel:

getout

setinfunc

Beschreibung:

speicherbar

Der Befehl ***setinfunc*** definiert eine zusätzliche Verfahrwegsbegrenzung, die durch Eingangssignale an den Eingängen Din-1 bis Din-3 aktiviert werden.
Damit kann die Positionierung jeder Achse grundsätzlich oder richtungsgebunden gesperrt werden.
Die Funktion wirkt auf die programmierte und manuelle Bewegung.
Bei einer laufenden Positionierung werden zunächst alle Achsen gestoppt, danach gelten die mit ***setinfunc*** festgelegten Bedingungen.
Wird das sperrende Eingangsignal zurückgenommen, kann die Achse wieder normal positioniert werden.

Ist eine ***setinfunc*** Bedingung aktiv wird dies im Statusbit D6 angezeigt. Zusätzlich meldet die LED-Diagnoseanzeige diesen Zustand durch Blinken der betreffenden Axis-LED.

Syntax:

[Index] [Eingang] [Achse] ***setinfunc***

	Wertebereich
[Index]	0, 1, 2, 3
[Eingang]	1, 2, 3
[Achse]	1, 2, 3

[Index]	**Funktionen der Abschaltung**
0	keine Abschaltfunktion aktiviert
1	Nach Abschaltung nur noch in negative Richtung
2	Nach Abschaltung nur noch in positive Richtung
3	Nach Abschaltung keine Richtung möglich

Partnerbefehl:

getinfunc

Beispiel:

1 1 3 setinfunc
2 2 3 setinfunc

Für Achse-3 wurden folgende Abschaltbedingungen festgelegt:
Bei Signaleingang an Din-1 werden alle Achsen sofort gestoppt, danach kann die 3.Achse nur noch in negative Richtung positioniert werden.

Bei Signaleingang an Din-2 werden alle Achsen sofort gestoppt, danach kann die 3.Achse nur noch in positive Richtung positioniert werden.

getinfunc

Beschreibung:

Das Kommando ***getinfunc*** liefert die Einstellung der Abschaltfunktion ***setinfunc*** zurück.

Syntax:

[Eingang] [Achse] ***getinfunc***

	Wertebereich
[Eingang]	1, 2, 3
[Achse]	1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1, 2, 3

Beispiel:

1 3 getinfunc

Sonderfunktionen

setclwindow

Beschreibung:


speicherbar

Der Befehl ***setclwindow*** aktiviert ein Positionszielfenster für den Closed Loop Betrieb.
Befinden sich alle beteiligten Achsen innerhalb ihres Zielfensters, wird Statusbit D5 gesetzt, gleichzeitig blinkt die an der Diagnoseanzeige zugeordnete Closed Loop LED.
Mit Window Parameter=0 wird das Fenster abgeschaltet.

Syntax:

[Window] [Achse] ***setclwindow***

	Wertebereich	Einheit	Funktion
[Window]	0.0000-100	user	Fenster aktiv
[Achse]	1, 2, 3	-	

***Fenster abschalten:**

[Window]	0	user	Fenster nicht aktiv

*Werkseitige Einstellung

Partnerbefehl:

getclwindow

Beispiel:

0.001 1 setclwindow
User Einheit = mm
Für Achse-1 wird ein Positionsfenster von 1µm festgelegt.

getclwindow

Beschreibung:

Der Befehl ***getclwindow*** liest die Einstellung der Fensterbreite für das Zielfensters im Closed Loop Betrieb.

[

Syntax:

[Achse] ***getclwindow***

	Wertebereich
[Achse]	-1, 1,2,3

Rückmeldung:

[Wert]

	Wertebereich	**Einheit**
[Wert]	0.0000 - 100	user

Rückmeldung:

1 getclwindow

Rückmeldung:
0.001

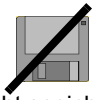
-1 getclwindow

Rückmeldung
0.001 0.002 0.000

setmp

Beschreibung:

Mit dem Kommandol ***setmp*** besteht die Möglichkeit die Motore stromlos zu schalten.
Alle anderen Funktionen der Steuerung arbeiten weiterhin.



nicht speicherbar

Syntax:

[Schalter] [Achse] ***setmp***

	Wertebereich
[Schalter]	0, 1
[Achse]	1, 2, 3

[Schalter]	**Funktion**
0	Der Motor wird stromlos geschaltet und besitzt nur noch sein natürliches Rastmoment
1	Der Motor ist normal bestromt

Beispiele:

0 1 setmp

Achse-1 wird stromlos geschaltet.

getmp

Beschreibung:

Das Kommando ***getmp*** liest die mit ***setmp*** vorgenommene Einstellung.
Die Abschaltung der Motorendstufe über das "safety device" damit nicht erkannt werden.

Syntax:

[Achse] ***getmp***

Parameter	Wertebereich
[Achse]	-1, 1, 2, 3

Rückmeldung:

[Schalter]

	Wertebereich
[Schalter]	0, 1

Beispiele:

1 setmp ***-1 getmp***

Rückmeldung:

0 0 1 1

randmove

Beschreibung

Der Befehl ***randmove*** erzeugt für aktive Achsen zufällige Positionsdaten innerhalb des gültigen Verfahrbereichs.
Die Position, die Geschwindigkeit und die Beschleunigung werden zufällig eingestellt.
Die Funktion wird sofort abgebrochen wenn die Steuerung ein beliebiges ASCII-Zeichen empfängt.

Der Befehl kann nur ausgeführt werden,
wenn zuvor die Limits mit den Kommandos *calibrate* und *rangemeasure* bestimmt oder mit *setlimit* festgelegt wurden.

Syntax:

randmove

Beispiel:

```
cal
rm
randmove
```

Zunächst werden die Limits der Achsen ermittelt, danach der "random move" gestartet. Dieser positioniert alle aktiven Achsen auf zufällige Koordinaten innerhalb des Verfahrbereichs.

Joystick / Handrad

joystick (j)

Beschreibung:

speicherbar

Der Befehl ***joystick*** aktiviert oder deaktiviert den Joystickbetrieb.
Bei aktiven Joystickbetrieb wird Statusbit D1 gesetzt.
Eine Meldung erfolgt auch durch die LED-Diagnoseanzeige.

Spezielle Funktion für den Joystickbetrieb

Nach dem Einschalten der Steuerung werden die Nullstellungspegel der Joystickpotentiometer geprüft.
Eine Toleranz von +/-10% ist zulässig. Bei größeren Abweichungen kann die betreffende Achse nicht für den Joystickbetrieb aktiviert werden.

Syntax:

[Index] ***joystick***

Parameter	Wertebereich
[Index]	0, 1

Funktion

Index	Funktion
0	Manueller Betrieb ausgeschaltet
1	Manueller Betrieb eingeschaltet

Beispiel:

1 j

Der manuelle Betrieb wird eingeschaltet.

setjoysticktype

Beschreibung:

Dieses Kommando wurde übernommen um die Kompatibilität der Steuerung Corvus zu den Vorgängermodellen MC-2000 und mc-compact zu gewährleisten.
Eine Änderung der Werkseinstellung ist nicht notwendig.

Syntax:

[Index] ***setjoysticktype***

Index= 3*

* Werkseinstellung

Partnerbefehl:

getjoysticktype

Beispiel:

3 setjoysticktype

getjoysticktype

Beschreibung:

Das Kommando ***getjoysticktype*** liest die durch ***setjoysticktype*** festgelegte Einstellung zurück.

Syntax:

getjoysticktype

Rückmeldung:

[Index]

Beispiel:

getjoysticktype

setjoyspeed (js)

Beschreibung:

speicherbar

Das Kommando ***setjoyspeed*** definiert die Geschwindigkeit mit der die Achsen bei maximaler Auslenkung des Joysticks positioniert werden.

Syntax:

[Geschwindigkeit] ***setjoyspeed***

	Einheit
[Geschwindigkeit]	Unit

	Wertebereich
min. Geschwindigkeit	15.62nm/s
max. Geschwindigkeit	180 mm/s

Partnerbefehl:

getjoyspeed

Beispiel:

20 setjoyspeed

unit = mm

Die Joystickgeschwindigkeit wird auf 20 mm/s eingestellt.

getjoyspeed

Beschreibung:

Das Kommando ***getjoyspeed*** liest die eingestellte maximale Geschwindigkeit für den Joystickbetrieb zurück.

Syntax:

getjoyspeed

Rückmeldung:

[Wert]

	Einheit
[Wert]	unit

Beispiel:

getjoyspeed

Rückmeldung:

20.000000

setjoybspeed

Beschreibung:

speicherbar

Mit dem Kommandol ***setjoyb**speed*** kann eine zusätzliche Joystickgeschwindigkeit festgelegt werden. Diese wird mit dem linken und rechten Taster am Joystick aktiviert.

* button

Syntax:

[Geschwindigkeit] ***setjoybspeed***

	Einheit
[Geschwindigkeit]	Unit

	Wertebereich
min. Geschwindigkeit	15.62nm/s
max. Geschwindigkeit	60 UmdrehungenxSpindelsteigung

Partnerbefehl:

getjoybspeed

Beispiel:

0.01 setjoybspeed
unit = mm
So lange der Joysticktaster gedrückt ist, beträgt die maximale Joystickgeschwindigkeit 0.01 mm/s.

getjoybspeed

Beschreibung:

Mit dem Befehl ***getjoybspeed*** wird die Einstellung der zweiten Joystickgeschwindigkeit abgefragt.

Syntax:

getjoybspeed

Rückmeldung:

[Wert]

	Einheit
[Wert]	unit

Beispiel:

getjoybspeed

Rückmeldung:

0.010000

Systemkommandos

save

Beschreibung:

Das Kommando ***save*** speichert alle speicherbaren Parameter in den nichtflüchtigen Speicher der Steuerung. Diese Einstellungen bleiben auch nach dem Abschalten der Steuerung erhalten.
Der Befehl beendet die aktuelle programmierte Positionierung. Die Joystickpositionierung wird nur während der Speicherphase unterbrochen.

Eine direkte Rückmeldung nach Beendigung des Speichervorgangs erfolgt nur im Terminal Modus.

Mit der Befehlsfolge ***save status*** kann auch im Host Mode eine automatische Rückmeldung erzeugt werden.

Speicherbare Parameter sind in diesem Handbuch mit untenstehenden Symbol gekennzeichnet.

speicherbar

Syntax:

save

Rückmeldung:

Im Terminal Modus wird ein ***OK*** zurückgeliefert.

Beispiel:

save

restore

Beschreibung:

Der Befehl ***restore*** bewirkt ein Wiederherstellen der zuletzt mit ***save*** gespeicherten Einstellungen.

Eine direkte Rückmeldung nach Beendigung der Wiederherstellung erfolgt nicht.
Diese kann mit der Befehlsfolge ***restore status*** erzeugt werden.

Syntax:

restore

Rückmeldung:

keine direkte Rückmeldung

Eine indirekte Rückmeldung mit der Befehlsfolge ***restore status*** ist möglich.

Verwandter Befehl:

getfpara

Beispiel:

restore

reset

Beschreibung:

Das Kommando ***reset*** bewirkt einen kompletten Neustart der Steuerung und ist mit dem Ein-Ausschalten vergleichbar. Die Position, die Limits, die Koordinatenfestlegungen durch ***setpos*** und der Stack werden gelöscht.

Während der Resetphase wird die LED-Diagnoseanzeige kurz abgeschaltet. Ein kurzer Signalton signalisiert danach die Betriebsbereitschaft.

Syntax:

reset

Beispiel:

reset

clear

Beschreibung:

Das Kommando ***clear*** löscht den Inhalt des Parameter Stack.

HINWEIS!

Der Parameter Stack kann maximal 99 Parametersätze aufnehmen.
Im Normalfall sollten sich nur wenige Parameter auf dem Stack befinden, da der Interpreter bei der Verarbeitung des Kommandos alle benötigten Parameter vom Stack nimmt.

Bei nicht korrekt eingehaltener Synax besteht die Gefahr dass es zu einer überschüssigen Anzahl von Parametern auf dem Stack kommt, der in der Folge zu einem Überlauf des Stack führen kann.

Mit dem Kommando ***gsp*** kann die Anzahl der Parameter die sich auf dem Stack befinden gelesen werden.

Syntax:

clear

Verwandter Befehl:

gsp

Beispiel:

clear

Löscht den Parameter Stack

gsp

Beschreibung:

Der Befehl ***gsp*** ermittelt die Anzahl der Elemente auf dem Parameter Stack.
Siehe “clear” auf Seite 144.

Syntax:

gsp

Rückmeldung:

[Wert]

	Wertebereich
[Wert]	0 -99

Verwandter Befehl:

clear

Beispiel:

gsp

Rückmeldung:
2

Liste der Kommandos