Handbuch

Venus-1
Kommandosprache

für

Corvus

high resolution positioning controller

SMC Corvus
SMC Corvus eco
SMC PCI

Über diese Dokumentation

In diesem Handbuch ist die Programmiersprache Venus-1 der Positioniersteuerungen Corvus TT, Corvus eco und Corvus PCI beschrieben.

Zur besseren Übersicht sind die Kommandos in praxisnahe Gruppen eingeteilt.

Im Anhang befindet sich eine Zusammenfassung aller Kommandos mit Kurzbeschreibung und Beispielen.

Verwendete Symbole

Zur verdeutlichung besonderer Inhalte werden die folgenden Symbole verwendet:

Symbol	Beschreibung
	Dieser Hinweis muss dringend beachtet werden.
	Wichtige Information und Hilfestellung
Option 	Zeigt an, dass diese Option mit einem Freischaltcode aktiviert werden kann.
Option 	Zeigt an, dass die Steuerung für diese Option mit zusätzlicher Hardware ausgestattet werden muss. Diese Erweiterung kann nur im Werk, oder durch eingewiesenes Fachpersonal erfolgen.
Venus-1	Venus-1 Kommando ***kursiv*** geschrieben
	Zeigt an, dass diese Einstellung in der Steuerung gespeichert werden kann.
	Zeigt an, dass diese Einstellung nicht gespeichert werden kann.

Inhaltsverzeichnis

Kommunikation

Geschwindigkeit und Beschleunigung

Positionierkommandos

Endschalterfunktionen

Sicherheitsfunktionen

Position / Bezugspunkt / Koordinatensytem

Statusabfragen

Input / Output Funktionen

Closed Loop Kommandos

Trigger-Output-Funktionen

Trigger-Input Funktionen

Joystick / Handrad

Systemkommandos

Fehlerkorrektur

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 erstmalig für die Steuerungen mc-compact, smc-compact, MC-2000 und MC-3000 verwendet wurde.

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	-
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 der Positionierbefehle

[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 der Dimension mit dem Kommando *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.

Es 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** werden die Kommandos mit der CR (carriage return) abgeschlossen.
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

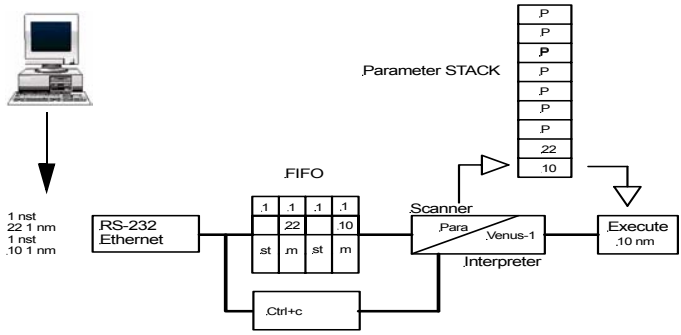
Die von der Schnittstelle übertragenen Zeichen werden zunächst in diesen Eingangspuffer übertragen. Der Speicher besitzt eine FIFO Struktur (First_In_First_Out).

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

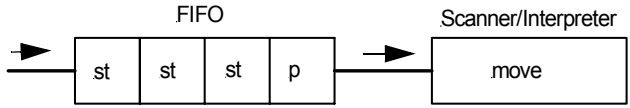
Während der Interpreter eine Positionierung ausführt, ist die gleichzeitige Verarbeitung verschiedener Kommandos möglich. Diese werden auch als **nicht sperrende Kommandos** bezeichnet.

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

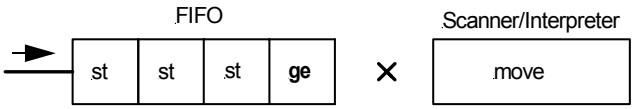
Der Interpreter verarbeitet mehrere Kommandos gleichzeitig:

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.



Beispiele nicht sperrender Kommandos

Wichtige Kommandos die keine Sperrung der Befehlsverarbeitung bewirken. Diese Liste ist nicht vollständig.

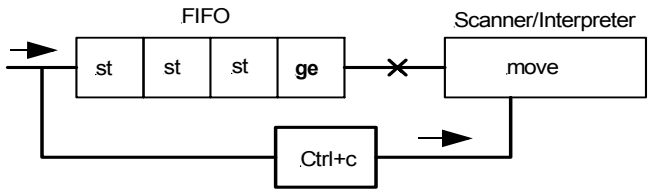
Venus-1 Kommando	Funktion
st	Statusrückmeldung
p	Aktuelle Position lesen
getin	Digitalen Eingang lesen
setout	Digitalen Ausgang schreiben
abort	Der aktuell ausgeführte Befehl wird abgebrochen. Achtung: Dieser Abbruchbefehl durchläuft im Gegensatz zu Ctrl+c das Daten-FIFO und kann somit durch ein anders Kommando blockiert werden.
Ctrl+c	Der aktuell ausgeführte Befehl wird 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 Dateneingangspuffer sondern wirkt unmittelbar auf den Interpreter.
Das FIFO wird dabei nicht gelöscht.



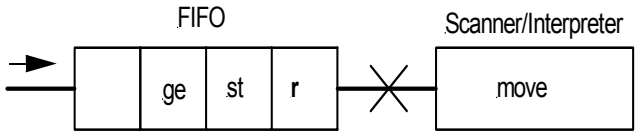
Erzeugen einer automatischen Statusrückmeldung

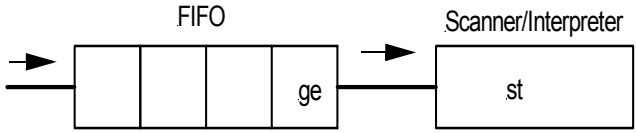
Mit der folgenden Befehlssequenz kann die sperrende Wirkung eines Kommandos für die Erzeugung einer automatische Statusrückmeldung genutzt werden:

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





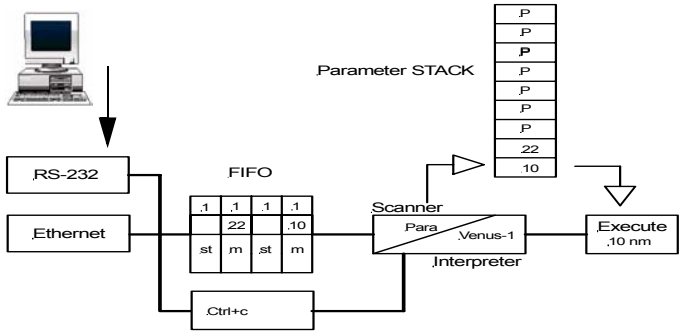
Corvus Kommunikationskonzept

Ethernet und RS-232

Corvus ist standardmäßig mit einer RS-232 Schnittstelle ausgestattet. Optional kann die On-Board verfügbare Ethernetschnittstelle freigeschaltet werden.
Beide Schnittstellen sind gleichzeitig empfangsbereit. Die Datenrückmeldung erfolgt automatisch immer auf die Schnittstelle von der die Datenanfrage stammt.

Terminal und Host Mode wird von beiden Schnittstellen unterstützt.

Weiterhin verfügbar ist die IEEE 488 Schnittstelle.



Grundeinstellungen

setpitch

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***setpitch*** wird die Steuerung an das Verhältnis von Motorumdrehung und resultierender linearer Bewegung angepaßt.
Der Wert von ***setpitch*** entspricht der resultierenden linearen Bewegung bei einer Motorumdrehung.

Syntax:

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

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

Partnerbefehl:

getpitch

Beispiel:

4.009 1 setpitch

Bei Achse-1 erzeugt eine Motordrehung eine lineare Bewegung von 4.009 mm.

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

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getpitch*** liefert die mit ***setpitch*** vorgenommene Einstellung zurück.

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

setdim

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***setdim*** wird die Achsdimension eingestellt.
Damit ist auch festgelegt, wieviele Achsparameter bei dimensionsabhängigen Kommandos von der Steuerung erwartet bzw. zurückgeliefert werden.

Einstellung	Erwartete Achsparameter
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

Die Anzahl der Parameter bei Positionierkommandos:
Abhängig von der Einstellung ***setdim*** müssen bei einem Positionierkommando die Koordinaten für Achse-1, Achse-2 und Achse-3 übertragen werden.

Die Anzahl der Parameter bei Positionsrückmeldungen
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

Corvus TT	Corvus eco	Corvus PCI

I

Beschreibung:

Das Kommando ***getdim*** liest die Einstellung der Dimension zurück.

Syntax:

getdim

Rückmeldung:

[Dimension]

	Wertebereich
[Dimension]	1, 2, 3

Beispiel:

getdim

setunit	Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setunit*** definiert die physikalische Einheit der Ein- und Ausgabewerte für Geschwindigkeit, Beschleunigung und Position.
Diese können über den Achsenindex jeder Achse individuell zugeordnet werden.

Die Einheit der Geschwindigkeit und Beschleunigung wird mit Hilfe einer virtuellen Achse festgelegt. Diese Achse hat den Achsenindex 0.

Folgende Funktionen sind Sonderfälle:
setrmvel, setcalvel, setrefvel
Für diese Befehle gilt die Einheit Umdrehungen/s

Die Einheit Microstep wird aus Kompatibilitätsgründen von Corvus emuliert. Die Auflösung der Steuerung ist damit reduziert:
1 Motorumdrehung entspricht dann 40000 Microsteps
Für Neuentwicklungen sollte diese Einheit nicht verwendet werden.

Syntax:

[Unit] [Achsen] ***setunit***

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

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

Verwandter Befehl:

getunit

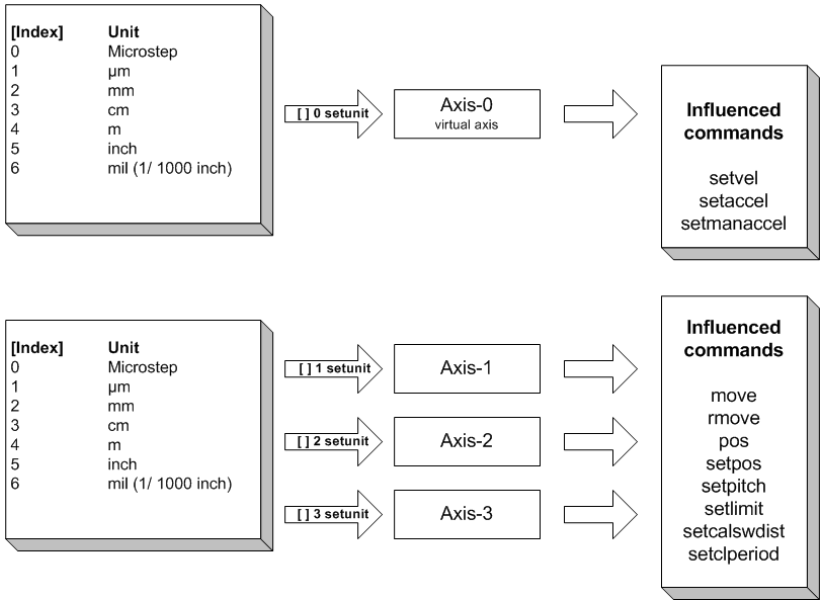
Beispiel:

2 0 setunit

Damit erfolgt die Ein- und Ausgabe des Geschwindigkeits- und Beschleunigungswerte in der Einheit mm.

1 1 setunit

Alle achsspezifischen Ein- Ausgaben sowie die Positionsrückmeldung der Achse-1 erfolgen damit in der Einheit µm.



getunit

Corvus TT	Corvus eco	Corvus PCI

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

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***setumotmin*** wird die Motorspannung für den Stillstand und unteren Drehzahlbereich eingestellt.

setumotmin beeinflusst damit das Haltemoment des Motors.

Bei größerem Wert umotmin vergrößert sich der Phasenstrom des Motors. Dies führt zu einer größeren Verlustleistung am Motor und stärkeren Belastung der Motorendstufe.

Syntax:

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

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

Verwandte Befehle:

getumotmin, setumotgrad

Beispiel:

2000 1 setumotmin

getumotmin

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getumotmin*** liest die durch *setumotmin* vorgenommene Einstellung zurück.

Syntax:

[Achse] ***getumotmin***

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

Rückmeldung:

[Index]

	Wertebereich	**Einheit**
[Index]	0 - 3000	mV

Beispiel:

1 getumotmin

Rückmeldung:
1000

-1 getumotmin

Rückmeldung:
1000
1000
750

setumotgrad

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***setumotgrad*** wird die Motorspannung bzw. der Phasenstrom für den mittleren und oberen Drehzahlbereich eingestellt.
Die Einstellung beeinflusst damit das Drehmoment bei mittlerer Drehzahl.

Mit größerem Wert von *setumotgrad* vergrößert sich der Phasenstrom und das Drehmoment während der Fahrt.
Achtung: Dies erhöht auch die Verlustleistung am Motor und führt zu einer stärkeren Belastung der Motorendstufe.

Syntax:

[Wert] [Achse] ***setumotgrad***

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

Verwandte Befehle:

getumotgrad, setumotmin

Beispiel:

70 1 setumotgrad

getumotgrad

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

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

Syntax

[Achse] ***getumotgrad***

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

Rückmeldung:

[Wert]

	Wertebereich
[Index]	0 - 300

Beispiel:

1 getumotgrad

Rückmeldung:
50

-1 getumotgrad

Rückmeldung:
50
50
100

setpolepairs

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***setpolepairs*** wird die Anpassung an die Polpaarzahl des Schrittmotors vorgenommen.
In der untenstehenden Tabelle sind zwei typische Schrittmotortypen und deren Polpaarzahl aufgelistet.

Schrittmotortype	Polpaare
Hybrid Schrittmotor mit Vollschrittwinkel = 1.8°	50
Hybrid Schrittmotor mit Vollschrittwinkel = 0.9°	100

Syntax:

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

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

Partnerbefehl:

getpolepairs

Beispiel:

50 1 setpolepairs

Achse-1 wird für einen Schrittmotor mit 50 Polpaaren konfiguriert.

getpolepairs

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Syntax:

[Achse] ***getpolepairs***

Parameter	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[Wert]

Rückmeldung	Wertebereich
[Wert]	50, 100

Beispiel:

1 getpolepairs

Rückmeldung:
50

-1 getpolepairs

Rückmeldung:
50 100 50

j

setaxis

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setaxis*** aktiviert oder deaktiviert die Achse für die Positionierung und die Endschalterfahrt. Zusätzlich wird die Wirkung der Kommandos *cal, rm* und *setpos* auf die Positionsanzeige und Limits beinflusst.

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 immer abgeschaltet
Die Kommandos ***cal***, ***rm*** und ***setpos*** bewirken ein zurücksetzen der Position.
Die Limits der Achse werden nicht verändert.

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

[Wert] = 2: Die Achse ist eingeschränkt aktiv.
Nur 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 immer abgeschaltet.
Die Kommandos ***cal***, ***rm*** und ***setpos*** haben keine Wirkung auf die Position und die Limits der Achse.

[Wert] = 4: Die Achse ist eingeschränkt eingeschaltet.
Nur 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

getaxis

Corvus TT	Corvus eco	Corvus PCI

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

setpowerup

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***setpowerup*** können Power-Up Befehle, automatisch nach dem Einschalten der Steuerung ausgeführt werden.
Jedem Befehl wurde eine binärer Wert D0 bis D4 zugeordnet.

Es ist möglich mehrere Power-Up Befehle zu kombinieren, hierzu müssen deren binäre Werte addiert und als dezimaler Parameter mit dem Kommando übergeben werden.
Sinnvolle Befehlskombinationen werden durch folgende Parameterwerte erzeugt: 0, 1, 2, 3, 4, 5, 6, 7, 15, 16

Syntax:

[Parameter] ***setpowerup***

	wirksame Kombinationen
[Parameter]	0, 1, 2, 3, 4, 5, 6, 7, 15, 16

bin / dez	**Befehl**	**Beschreibung**
D0 (1)	*1j*	Joystick ein/aus Mit powerup = 0 bleibt die vorher gespeicherte Joystickein-stellung gültig.
D1 (2)	*cal*	Alle aktiven Achsen fahren zum Endschalter cal.
D2 (4)	*rm*	Alle aktiven Achsen fahren zum Endschalter rm.
D3 (8)	*rand-move*	Zufallsfahrt aller aktiven Achsen innerhalb der Limits Die Verfahrbereich zwischen den Endschaltern muss zuvor ermittelt sein, sonst besteht die Gefahr, dass die Achsen den max. Verfahrbe-reich überschreiten!

bin / dez	Befehl	Beschreibung
D4 (16)	*cal* *rm* *0 0*m* *abhängig von *setdim*	Endschalterfahrt *cal* u. *rm*, danach Fahrt zum Nullpunkt der Achsen.

Beispiele:

1 setpowerup
Der Joystick wird eingeschaltet

15 setpowerup
Nach dem Einschalten der Steuerung wird zunächst die Endschalterfahrt in den cal- und rm-Endschalter ausgeführt, danach werden die aktiven Achsen auf zufällige Positionen innerhalb der Limits positioniert.

getpowerup

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getpowerup*** liest die Power-Up Einstellung der Steuerung zurück.

Syntax:

getpowerup

Rückmeldung:

[Parameter]

Parameter	0, 1, 2, 3, 4, 5, 6, 7, 15, 16

Beispiel:

getpowerup

Rückmeldung:
3

setphaseares

Corvus TT | Corvus eco | Corvus PCI

Verfügbar ab Firmwareversion 3.6.3

Beschreibung:

Mit dem Kommando ***setphaseares*** kann die Winkelschrittauflösung der Motorendstufen stufenweise reduziert werden.
In der untersten Stufe (2 Bit) werden die Endstufen im Schrittmotor Makroschritt betrieben.

Syntax:

[Auflösung (Bit)] [Achse] ***setphaseares***

	Wertebereich
[Auflösung] *	2....16
[Achse]	1, 2, 3

* Werkseinstellung = 16

Polpaare = 50

Bit	**Schrittauflösung (Steps/Umdr.)**
2	200
16	>600.000

Polpaare = 100

Bit	**Schrittauflösung (Steps/Umdr.)**
2	400
16	>1.200.000

Beispiel:

2 1 setphaseres

Achse-1 wird auf Makroschrittbetrieb eingestellt.

getphaseares

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 3.6.3

Beschreibung:

Das Kommando ***getphaseares*** liefert die eingestellte Stufe der Schrittwinkelauflösung zurück.

Syntax:

[Achse] ***getphaseares***

Rückmeldung:

[Auflösung (Bit)]

	Wertebereich
[Auflösung]	2.....16

Bit	**Polpaarzahl****	**Schrittauflösung (Steps/Umdr.)**
2	50	200
16	50	>600.000

** siehe Kommando *setpolepairs*

Bit	**Polpaarzahl**	**Schrittauflösung (Steps/Umdr.)**
2	100	400
16	100	>1.200.000

Beispiel:

2 getphaseares

setmotiondir

Corvus TT | Corvus eco | Corvus PCI

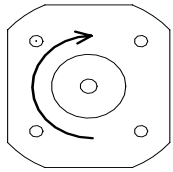
Verfügbar ab Firmwareversion 4.4.0

Beschreibung:

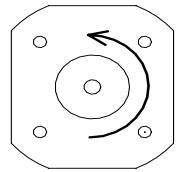
Mit Kommando ***setmotiondir*** wird die Motordrehrichtung festgelegt.

Bei Änderung der werkseitigen Einstellung (setmotiondir=0) wird gleichzeitig die Funktion der Endschaltereingänge cal und rm getauscht.
Das bedeutet, dass die Steuerung bei der cal-Endschalterfahrt den Endschalter am rm-Eingang erwartet.

Motordrehrichtung bei einer Positionierung in Richtung positiver Koordinaten



(Werkseinstellung)



Drehrichtung wenn mit ***setmotiondir*** gedreht

Syntax:

[Funktion] [Achse] ***setmotiondir***

[Funktion]	Beschreibung
0	Werkseitige Einstellung
1	Umkehr der Drehrichtung. Die Funktion der Endschaltereingänge cal und rm ist getauscht

Beispiel:

1 1 setmotiondir

getmotiondir

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.4.0

Beschreibung:

Das Kommando ***getmotiondir*** zeigt an ob die Motordrehrichtung von der Standardeinstellung abweicht.

.

Syntax:

[Achse] ***getmotiondir***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[0,1]

	Funktion
[0]	Standardeinstellung
[1]	Motordrehrichtung gedreht. Funktion der Endschalter cal/rm ist getauscht.

Beispiel:

1 getmotiondir

Kommunikation

mode

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***mode*** wird der Terminal- oder Host Mode eingestellt.
Im **Terminal Mode** überträgt die Steuerung automatisch eine Bildschirmmaske mit Positionsanzeige und einer Kommandoeingabezeile an die Schnittstellen.
Im **Host Mode** überträgt die Steuerung nur Daten, wenn diese vom Host durch ein Venus-1 Kommando 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

Corvus TT

Beschreibung:

Mit dem Kommando ***setipadr*** wird die Ethernet IP-Adresse der Steuerung festgelegt.

Weitere Einstellungen:
Subnetmask: 255.255.255.0 (feste Einstellung)
Port: 23 / Telnet
Socket: TCP/IP Winsock

Syntax:

[AAA]_[BBB]_[CCC]_[DDD]_setipadr
Die Blöcke sind mit einem Leerzeichen zu trennen!

Partnerbefehl:

getipadr

Beispiel:

192_168_128_0_setipadr

getipadr

Corvus TT

Beschreibung:

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

Werkseitige Einstellung

```
IP address: 192.168.1.2
```

Syntax:

getipadr

Rückmeldung:

[AAA].[BBB].[CCC].[DDD]

Verwandter Befehl:

getmacadr (Auslesen der MAC-Adresse)

Beispiel:

getipadr

Rückmeldung:

192.168.128.0

Die Blöcke sind bei der Rückmeldung mit einem Punkt getrennt.

Geschwindigkeit und Beschleunigung

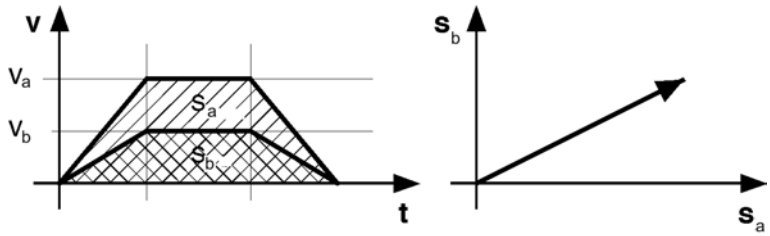
setvel (sv)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung

Mit dem Kommando ***setvel*** wird die Geschwindigkeit **v**a festgelegt, mit der die Steuerung die programmierte Bewegung durchführt.
Für jede Positionierung errechnet die Steuerung unter Berücksichtigung der vorgegebenen Positionierstrecke aller aktiven Achsen ein individuelles Geschwindigkeitsprofil.
Die mit ***setvel*** festgelegte Geschwindigkeit bezieht sich dabei auf die Achse, die den längsten Verfahrweg **s**a zurücklegen muss, siehe Diagramm.
Die Maximalgeschwindigkeit der anderen Achsen ergibt sich aus dem Verhältnis der Verfahrstrecken **s**b oder **s**c zur Verfahrstrecke **s**a
Die Rotationsgeschwindigkeit des Motors ergibt sich aus der Einstellung ***setvel*** und *setpitch*.

$$v_b = \frac{s_b}{s_a} \cdot v_a \qquad v_c = \frac{s_c}{s_a} \cdot v_a$$



Bei der programmierten Positionierung werden alle aktiven Achsen gleichzeitig gestartet und erreichen zur gleichen Zeit das Ziel.

Syntax

[Geschwindigkeit] ***setvel***

	Wertebereich
min. Geschwindigkeit	15,26 nm/s
max. Geschwindigkeit	48 U/s bei pitch = 4mm ->180mm/s 60 U/s (optional)

	Einheit
[Geschwindigkeit]	Eingestellte Einheit der 0-Achse

Weitere Geschwindigkeitsbefehle

setcalvel / setrefvel / setjoystickspeed

Beispiel:

100 sv

unit = mm, pitch = 4mm
Für die programmierte Positionierung wird die Maximalgeschwindigkeit auf 100 mm/s begrenzt.

getvel (gv)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Syntax:

getvel

Rückmeldung:

[Geschwindigkeit]

	Einheit
[Geschwindigkeit]	unit

Beispiel:

gv

Rückmeldung:
180.000000

setaccel (sa)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***setaccel*** wird die Beschleunigung festgelegt, mit der die programmierten Positionierung ausgeführt wird.
Die Achsen werden linear interpoliert, daher bezieht sich diese Angabe auf die sogenannte Führungsachse. Die Führungsachse ist die Achse, die den längsten Verfahrweg zurücklegen muss.
Die Maximalbeschleunigung der anderen Achsen ergibt sich aus dem Verhältnis der Maximalgeschwindigkeiten, siehe *setvel*.
Die Maximal einstellbare Beschleunigung ist abhängig von der Spindelsteigung und Polpaarzahl.

Beschleunigungsrampe und die Bremsrampe sind identisch.

Syntax:

[Beschleunigung] ***setaccel***

	Wertebereich	**Einheit**
[Beschleunigung]	$0 - \frac{100000}{s^2} * \frac{\text{pitch [unit]}}{\text{Polpaare}}$	unit/s²

Polpaare: 50 oder 100 (siehe Kommando *setpolepairs*)

getaccel / setmanaccel

Verwandte Befehle:

Beispiel:

500 sa

getaccel (ga)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Syntax:

getaccel

Rückmeldung:

[Beschleunigung]

	Wertebereich	**Einheit**
[Beschleunigung]	$0 - \frac{100000}{s^2} * \frac{\text{pitch [unit]}}{\text{Polpaare}}$	unit/s²

Beispiel:

ga

Rückmeldung:
2400000.000000 (wenn unit = µm)

setaccelfunc

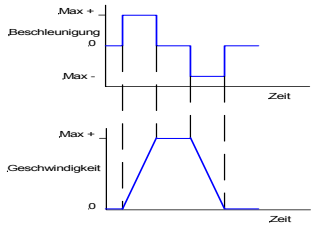
Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

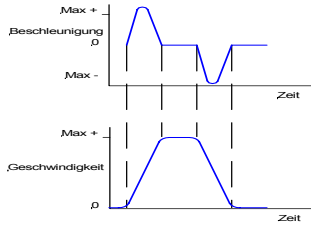
Das Kommando ***setaccelfunc*** legt die Beschleunigungsfunktion fest, mit der die Positionierung der Achsen ausgeführt wird. Die Einstellung wirkt auf alle Achsen.

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

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

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

Syntax:

getaccelfunc

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1

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

Beispiel:

getaccelfunc

setmanaccel

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setmanaccel*** stellt die Beschleunigung der Achse für den manuellen Betrieb ein.
Dies Einstellung 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

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getmanaccel*** liest die eingestellte Beschleunigung für den manuellen Betrieb.

Syntax:

getmanaccel

Rückmeldung:

[Wert]

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

Beispiel:

getmanaccel

Rückmeldung:
2400.000000

setcalvel

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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.

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.

Syntax:

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

	Wertebereich	Einheit
[Geschwindigkeit]	0 - 45	Umdrehungen/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

Spindelsteigung de 0-Achse wird auf 2mm eingestellt.
Die Steuerung bewegt die Achsen mit 2U/s (4 mm/s) in die cal-Endschalter hinein und mit 1U/s (2 mm/s) aus den cal-Endschaltern heraus.

getcalvel

Corvus TT	Corvus eco	Corvus PCI

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

setncalvel

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.1.1

Beschreibung:

Mit dem Kommando ***setncalvel*** werden die Geschwindigkeiten festgelegt, mit denen die Steuerung die ncal Endschalterfahrt ausführt.
Die Funktion arbeitet mit zwei Geschwindigkeiten:
1. Geschwindigkeit in den Endschalter hinein.
2. Geschwindigkeit aus dem Endschalter heraus.

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.

Syntax:

[Geschwindigkeit] [Index] [Achse] ***setncalvel***

	Wertebereich	Einheit
[Geschwindigkeit]	0 - 45	Umdrehung/s
[Index]	1, 2	-
[Achse]	1, 2, 3	-

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

Verwandte Befehle:

getncalvel, setnrmvel

Beispiel:

2 0 setpitch
2 1 1 setncalvel
1 2 1 setncalvel

Spindelsteigung de 0-Achse wird auf 2mm eingestellt.
Die Steuerung bewegt die Achse-1 mit 2 U/s (4 mm/s) in den cal-Endschalter hinein und mit 1 U/s (2 mm/s) aus dem cal-Endschalter heraus.

getncalvel

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.1.1

Beschreibung:

Das Kommando ***getncalvel*** liest die mit ***setncalvel*** eingestellte Geschwindigkeit der cal Endschalterfahrt zurück.

Syntax:

[Achse] getncalvel

Rückmeldung:

[Wert 1]
[Wert 2]

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

Beispiel:

1 getncalvel

Rückmeldung:

2.000000
0.250000

setrmvel

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

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

Die Einstellung ist für alle Achsen gültig.

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.

Syntax:

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

	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:

getrmvel, setcalvel

Beispiel:

2 0 setpitch

2 1 setrmvel
1 2 setrmvel

Die Spindelsteigung der 0-Achse wird auf 2mm eingestellt. 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

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getrmvel*** liest die mit *setrmvel* eingestellten Geschwindigkeiten für die Endschalterfahrt 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

setnrmvel

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.0.0

Beschreibung:

Das Kommando ***setnrmvel*** definiert zwei Geschwindigkeiten, mit denen die Steuerung die nrm-Endschalterfahrt ausführt.
1. Geschwindigkeit in den Endschalter hinein.
2. Geschwindigkeit aus dem Endschalter heraus.

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.

Syntax:

[Geschwindigkeit] [Index] [Achse] ***setnrmvel***

	Wertebereich	**Einheit**
[Geschwindigkeit]	0 - 45	Umdrehung/s
[Index]	1, 2	-
[Achse]	1, 2, 3,	-

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

Verwandte Befehle:

getnrmvel, setncalvel

Beispiel:

2 0 setpitch
2 1 2 setnrmvel
1 2 2 setnrmvel

Die Spindelsteigung der 0-Achse wird auf 2mm eingestellt.
Die Steuerung bewegt Achse-2 mit 2 U/s (4 mm/s) in den rm-Endschalter hinein und mit 1 U/s (2 mm/s) aus dem rm-Endschalter heraus.

getnrmvel	Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.0.0

Beschreibung:

Das Kommando ***getnrmvel*** liest die mit ***setnrmvel*** eingestellte Geschwindigkeit für die rm Endschalterfahrt zurück.

Syntax:

[Achse] getnrmvel

Rückmeldung:

[Wert 1]
[Wert 2]

	Wertebereich	**Einheit**
Wert 1	0 - 45	Umdrehungen/s
Wert 2	0 - 45	Umdrehungen/s

Beispiel:

2 getnrmvel

Rückmeldung:

2.000000
0.250000

setrefvel

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***setrefvel*** legt die Geschwindigkeit fest, mit der die Positionierung zur Referenzmarke ausgeführt wird.
Siehe auch ***refmove.***

Damit die Referenzmarkierung exakt und reproduzierbar angefahren werden kann, sollte eine möglichst kleine Referenzgeschwindigkeit gewählt werden.

Syntax:

[Geschwindigkeit] [Index] ***setrefvel***

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

Verwandter Befehle:

getrefvel / setvel / refmove / setref

Beispiel:

0.5 1 setrefvel

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

getrefvel

Corvus TT | Corvus eco | Corvus PCI

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 zurück. Der zweite Parameter ist irrelevant.

Syntax:

getrefvel

Rückmeldung:

[Geschwindigkeit]
[nf]

	Wertebereich	Einheit
[Geschwindigkeit]	0 - 45	mm/s
[nf]	keine Funktion	

Beispiel:

getrefvel

Rückmeldung:

0.500000
0.010000

Positionierkommandos

move (m)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***move*** positioniert die Achsen zu absoluten 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 für die absolute Positionierung 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)

Corvus TT	Corvus eco	Corvus PCI

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***

Relative Koordinaten	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

speed

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***speed*** wird die Achse im sogenannten speed mode bewegt. Damit erfolgt die Positionierung durch Angabe einer Geschwindigkeit und Bewegungsrichtung.
Der Geschwindigkeitswert und die Bewegungsrichtung können hierbei jederzeit (on the fly) verändert werden.

Ein aktiver speed modus wird im Statusbit reflektiert, siehe Kommando ***status***.
Der Arbeitsbereich wird durch die Limits begrenzt.
Die Bewegung aller Achsen wird durch das Kommando ***stopspeed*** oder ***Ctrl+c*** abgebrochen.

Syntax:

[Richtung] [Geschwindigkeit] [Achse] ***speed***

	Wertebereich	**Einheit**
[Richtung]	+ / -	
[Geschwindigkeit]	0-60 *	U/s
[Achse]	1,2,3	

* abhängig von der freigeschalteten Geschwindigkeit

Beispiel:

10 1 speed
Achse-1 wird mit 10 U/s in positive Richtung bewegt.
-0.1 2 speed
Achse-2 wird mit 0.1 U/s in negative Richtung bewegt.

stopspeed

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***stopspeed*** wird der speed mode für alle Achsen mit der eingestellten Systembeschleunigung abgebrochen.

Die Aktivität des speed mode wird im Status (*st*) angezeigt.

Syntax:

stopspeed

Beispiel:

stopspeed

test

Corvus TT	Corvus eco	Corvus PCI

Beschreibung

Der Befehl ***test*** aktiviert eine Testroutine, mit der die ausgewählte Achse schrittweise zu den Limits positioniert wird. Die Schrittgrösse ist frei wählbar.
Bei Erreichen der Limits, wird die Achse in die entgegengesetzte Richtung positioniert.
Die Positionierung beginnt immer am Nullpunkt der Achse.
Die Funktion wird abgebrochen wenn die Steuerung ein beliebiges ASCII-Zeichen empfängt.

Die Funktion kann erst ausgeführt werden, wenn alle Limits festgelegt sind.

Syntax:

[Schrittgrösse] [Achse] ***test***

Parameter	Wertebereich	Einheit
Schrittgrösse	beliebig	unit
Achse	1, 2, 3	

Beispiel:

cal
rm
10 1 test

unit = mm, die Limits sind festgelegt!

Achse-1 wird zum Nullpunkt bewegt und dannach
in Schritten von 10mm in positive Richtung positioniert.
Beim Erreichen des oberen Limits, erfolgt die gleiche Positionierung in negative Richtung.
Der Ablauf wird abgebrochen wenn die Schnittstelle ein ASCII Zeichen empfängt.

randmove

Corvus TT	Corvus eco	Corvus PCI

Beschreibung

Der Befehl ***randmove*** erzeugt für alle aktiven Achsen zufällige Positionsdaten innerhalb des gültigen Verfahrbereichs.
Die Geschwindigkeit und die Beschleunigung werden ebenfalls zufällig eingestellt, die aktuellen Einstellungen werden dabei aber nicht überschritten.

Die Funktion wird 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.

Endschalterfunktionen

calibrate (cal)

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***cal*** löst die Endschalterfahrt aller aktiven Achsen zum cal-Endschalter aus. Hierbei werden die aktiven Achsen gleichzeitig in negative Richtung positioniert. bis der cal-Endschalter betätigt ist. Die Steuerung positioniert danach wieder in Richtung positiver Positionswerte bis 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.
Mit ***Ctrl-C*** wird die Endschalterfahrt sofort abgebrochen und der Nullpunkt gesetzt.

Das untere Limit, sowie der Koordinatennullpunkt werden aus Sicherheitsgründen nicht gespeichert.
Das Kommando *cal* ist ein sperrendes Kommando.
Das heisst die Steuerung kann zwar weiterhin Kommandos empfangen aber diese nicht ausführen.
Ein während der cal-Fahrt empfangenes Kommando wird erst nach Beendigung des cal Prozedur ausgeführt.

Verwandter Befehl:

ncal

Syntax:

calibrate oder ***cal***

Beispiel:

cal

rangemeasure (rm)

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

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

Für die korrekte Ermittlung des maximalen Verfahrbereichs muss zuvor der Befehl *cal* ausgeführt werden.
Das Kommando *rm* sperrt den Interpreter, die Steuerung kann während der Befehl ausgeführt wird weiterhin Kommandos empfangen aber diese nicht ausführen. Empfangene Kommandos werden erst Beendigung der Endschalterfahrt in der Reihenfolge ihres Eintrags ausgeführt.

Mit der Einstellung ***setaxis*** wird die Wirkung der Endschalterfahrt auf die Limits, die Koordinaten und die aktuelle Position beeinflusst.

Syntax:

rangemeasure oder ***rm***

Beispiel:

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

getcaldone

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.42

Beschreibung:

Mit dem Befehl ***getcaldone*** wird abgefragt ob die Endschalterfahrt ***cal*** oder ***rm*** ausgeführt wurde.

Der Status der ***rm-***Endschalterfahrt wird bei einer ***cal-***Endschalterfahrt gelöscht.

Syntax:

[Achse] getcaldone

Rückmeldung:

[Dezimaler Wert]

	Wertebereich
[Wert]	0 - 3

[Wert]	***cal* ausgeführt**	***rm* ausgeführt**
0	nein	nein
1	ja	nein
2	nein	ja
3	ja	ja

Beispiel:

1 getcaldone
Rückmeldung:
1
Bei Achse-1 wurde der Befehl ***cal*** ausgeführt.

setsw	Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Befehl ***setsw*** werden die Endschaltereingänge cal und rm an das Schaltverhalten der Endschalter angepasst. Die Einstellung "ignorieren" schaltet den Endschaltereingang ab. Es sind folgende Funktionen möglich:

- Endschalter als Öffner
- Endschalter als Schliesser
- Endschalter ignorieren

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

Syntax:

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

[Funktion]	NPN-Schalter *	PNP-Schalter **
0	Schliesser	Öffner
1	Öffner	Schliesser
2	Ignorieren	Ignorieren

* NPN-Schalter schaltet gegen Masse.
** PNP-Schalter schaltet nach VCC

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

Beispiele:

0 0 1 setsw
Gilt für NPN Endschalter:
Konfiguriert den cal-Endschaltereingang von Achse-1 als Schliesser gegen GND.
2 1 2 setsw
Der rm-Endschaltereingang von Achse-2 wird abgeschaltet.

getsw

Corvus TT	Corvus eco	Corvus PCI

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 0 0 1 0 2 2

Axis-1

getswst

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getswst*** zeigt den Schaltzustand der Endschaltereingänge cal und rm an.

Syntax:

[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.

setcalswdist

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***setcalswdist*** kann jeder Achse ein zusätzlicher Abstand zu den Endschaltern vorgegeben werden.
Dieser Abstand wirkt auf beide Endlagen der Achse.
Der verfügbare Arbeitsbereich verkleinert sich dadurch entsprechend.

Damit ergibt sich folgender Ablauf bei der Endschalterfahrt:

1. Fahrt in den Endschalter, bis dieser betätigt ist
2. Fahrt aus dem Endschalter heraus, bis wieder unbetätigt
3. Fahrt (Strecke) die durch ***setcalswdist*** festgelegt wurde zusätzlich vom Endschalter weg.

Syntax:

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

	Wertebereich	**Einheit**
[Strecke]	0 - 16383	User unit
[Achse]	1, 2, 3	

Verwandter Befehl:

getcalswdist

Beispiel:

0.5 1 setcalswdist

Für Achse-1 wird ein zusätzlicher Abstand
von 0.5 User-Einheiten zu den Endschaltern cal und rm festgelegt.

getcalswdist

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getcalswdist*** liest Einstellung von *setcalswdist* zurück.

Syntax:

[Achse] ***getcalswdist***

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

Rückmeldung:

[Wert]

	Einheit
[Wert]	Umdrehungen

Beispiel:

-1 getcalswdist

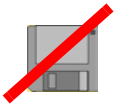
Terminal:

```
Axis: 1  Cal switch distance: 0.0000000000
Axis: 2  Cal switch distance: 0.0000000000
Axis: 3  Cal switch distance: 0.0000000000
```

setlimit

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:



Mit dem Kommando ***setlimit*** werden Softlimits für alle Achsen festgelegt, damit lässt sich der Verfahrbereich der Achsen beliebig einschränken.
Ein Positionierauftrag, der den mit ***setlimit*** festgelegten Verfahrbereich überschreiten würde, wird mit der eingestellten Systembeschleunigung abgebremst und kommt an der Verfahrbereichsgrenze zum Stillstand.
Hierbei wird im Fehlerspeicher der Fehlercode 1004 abgelegt (siehe *ge*)

Mit dem Befehl ***setnlimit*** können die Softlimits für einzelne Achsen festgelegt werden.

Im manuellen Betrieb werden die Limits durch Anfahren und betätigen der Endschalter ermittelt.

Bedingungen für die Festlegung von Soft-Limits

- Wurden Hard-Limits festgelegt, können die Soft-Limits nur innerhalb der Hard-Limits liegen.
- Der Wert des unteren Limits muß kleiner sein als der Wert des oberen Limits.
- Bei der Übertragung der Soft-Limits muss sich die aktuelle Position innerhalb dieser Limits befinden.
- Für abgeschaltete Achsen ***getaxis*** = 0) wird kein Limit übernommen.
- Bei Ausführung der Befehle ***cal*** und ***rm*** werden die Koordinaten der Soft-Limits von den Hard-Limits überschrieben.
- Mit dem Befehl ***reset*** wird die Steuerung neu gestartet und alle Limits auf den maximalen Wert gesetzt.

Syntax:

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

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

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

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

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

Partnerbefehl:

getlimit, getnlimit, setnlimit

Beispiel:

getdim = 3

0 0 -10 12 25 30 setlimit

Damit ergeben sich folgende Verfahrbereichsgrenzen:
Achse-1: 0 bis 12
Achse-2: 0 bis 25
Achse-3: -10 bis 30

getlimit

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***getlimit*** ermittelt die Verfahrgrenzen, die durch die Befehle ***cal, rm***, ***setlimit*** oder den manuellen move in die Endschalter festgelegt wurde.
Abhängig von der Einstellung ***setdim*** werden die Werte in 1, 2 oder 3 Zeilen zurückgeliefert.
Mit dem Kommando ***getnlimit*** können die Limits auch einzeln abgefragt werden.
Sind alle Achsen aktiv und wurde kein Limit festgelegt, liefert die Steuerung mit ***getlimit*** folgende Werte:

-16383.000000 16383.000000
-16383.000000 16383.000000
-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

0.000000 7.723750
0.000000 7.723750
-16383.000000 16383.000000

ncal

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.0.0

Beschreibung:

Mit dem Kommando ***ncal*** ist es möglich jede Achse separat zu ihrem unteren Endschalter zu bewegen.
Im Gegensatz zum Kommando ***cal*** wird der Interpreter bei ***ncal*** nicht blockiert. Dadurch können auch während der Befehl ausgeführt wird alle Statusabfragen beantwortet werden.
Der Ablauf der Endschalterfahrt ***ncal*** entspricht im sonstigen dem von ***cal.***
Die Geschwindigkeit, mit der diese Endschalterfahrt ausgeführt wird, kann mit dem Kommando ***ncalvel*** festgelegt werden.

Mit ***Ctrl-C*** wird die Endschalterfahrt sofort abgebrochen und der Nullpunkt gesetzt.

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

Syntax:

[Achse] ncal

Beispiel:

1 ncal

nrm

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.0.0

Beschreibung:

Mit dem Kommando ***nrm*** ist es möglich jede Achse separat zu ihrem oberen Endschalter zu bewegen.

Im Gegensatz zum Kommando ***rm*** wird der Interpreter durch das Kommando ***nrm*** nicht blockiert. Dadurch können während der Endschalterfahrt alle Statusabfragen beantwortet werden.
Der Ablauf der Endschalterfahrt ***nrm*** entspricht im sonstigen dem von ***rm.***

Mit ***Ctrl-C*** wird die Endschalterfahrt sofort abgebrochen und das obere Limit gesetzt.

Syntax:

[Achse] nrm

Beispiel:

1 nrm

getnlimit

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Mit dem Befehl ***getnlimit*** werden die gültigen Verfahrbereichsgrenzen einer einzelnen Achse ermittelt.

Das Kommando ***getlimit*** liefert die Limits aller Achsen.

Ist die Achse aktiv und ist noch kein Limit festgelegt, liefert die Steuerung mit ***getnlimit*** den maximalen Verfahrbereich.
-16383.000000 16383.000000

Syntax:

[Achse] getnlimit

Rückmeldung:

		Einheit
[unteres Limit]	[oberes Limit]	unit

Beispiel:

1 getnlimit

0.000000 12.00000

org	Corvus TT	-	-

Verfügbar ab Firmwareversion 4.1.0

Beschreibung:

Mit dem Kommando ***org*** wird die ausgewählte Achse zum org-Schalter bewegt (*setorg* muss dafür eingeschaltet sein). Ist der Schalter betätigt, wird die Achse in umgekehrter Richtung aus dem Schalter herausgefahren, bis dieser wieder unbetätigt ist.
Die Geschwindigkeit mit der diese beiden Bewegungen ausgeführt werden wird mit dem Befehl ***setncalvel*** festgelegt.
Wird der Schalter nicht innerhalb der vorgegebenen Strecke erreicht, stoppt die Achse und es wird Fehlercode 1011 erzeugt.

Mit Ctrl-C wird die Prozedur abgebrochen.

Ist der org-Schalter bei Eingabe des ***org*** Kommandos schon betätigt, wird sofort die zweite Bewegungrichtung ausgeführt, bis der Schalter unbetätigt ist.
Auch in diesem Fall wird Fehlercode 1011 erzeugt.

Syntax:

[Richtung] [rel. Strecke] [Achse] ***org***

	Wertebereich	**Einheit**
[Richtung]	+ / -	
[rel. Strecke]		Einheit
[Achse]	1, 2, 3	

Beispiele:

-10 1 org
Achse-1 wird 10 (units) in negative Richtung bewegt, bis der org-Schalter betätigt ist. Die Achse wird danach in positive Richtung bewegt, bis der Schalter wieder unbetätigt ist.

setorg

Corvus TT	-	-

Verfügbar ab Firmwareversion 4.1.0.

Beschreibung:

Das Kommando ***setorg*** aktiviert oder deaktiviert den org-Schaltereingang.

Syntax:

[Schalter] [Achse] ***setorg***

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

[Schalter]	**Funktion**
0	org-Eingang deaktiviert
1	org-Eingang aktiviert

Beispiele:

1 1 setorg

Der org-Eingang wird für Achse-1 eingeschaltet.

0 2 setorg
Der org-Eingang wird für Achse-2 abgeschaltet.

getorg	Corvus TT	-	-

Verfügbar ab Firmwareversion 4.1.0

Beschreibung:

Das Kommando ***getorg*** liefert die Einstellung des org-Schaltereingangs einer selektierten Achse zurück.

Syntax:

[Achse] ***getorg***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[index]

[index]	**Funktion**
0	org-Eingang deaktiviert
1	org-Eingang aktiviert

Beispiel:

1 getorg

setorgsw	Corvus TT	-	-

Verfügbar ab Firmwareversion 4.1.0.

Beschreibung:

Der Befehl ***setorgsw*** passt den org-Schaltereingang an das Schaltverhalten des org-Schalters an.

Folgende Einstellungen sind möglich:
- Schalter als Öffner
- Schalter als Schliesser

Syntax:

[Typ] [Achse] ***setorgsw***

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

[Typ]	**NPN Schalter ***	**PNP Schalter ****
0	Schliesser	Öffner
1	Öffner	Schliesser

* NPN-Schalter schalten gegen Masse.
** PNP-Schalter schalten gegen VCC.

Beispiele:

1 1 setorgsw

Für NPN Endschalter:
Konfiguriert den org-Endschaltereingang von Achse-1 als Öffner.

getorgsw	Corvus TT	-	-

Verfügbar ab Firmwareversion 4.1.0.

Beschreibung:

Das Kommando ***getorgsw*** liest die Einstellungen des org-Schaltereingangs zurück.

Syntax:

[Achse] ***getorgsw***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[Typ]

[Typ]	**NPN Schalter ***	**PNP Schalter ****
0	Schliesser	Öffner
1	Öffner	Schliesser

Beispiel:

3 getorgsw

Rückmeldung:
1

getorgswst	Corvus TT	-	-

Verfügbar ab Firmwareversion 4.1.0

Beschreibung:

Das Kommando ***getorgswst*** zeigt den aktuellen Schaltzustand des org-Schalters an.

Syntax:

[Achse] ***getorgswst***

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

Rückmeldung:

0	org-Schalter nicht betätigt
1	org -Schalter betätigt

Beispiele:

1 getorgswst
Rückmeldung:
0
Der org-Schalter von Achse-1 ist nicht betätigt

-1 getswst
Rückmeldung
0 1 0
Der org-Schalter der Achse-2 ist betätigt.

Sicherheitsfunktionen

Ctrl-C

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit 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 dazu verwendet werden die Endschalterprozedur abzubrechen und dabei die unteren bzw. oberen Verfahrbereichgrenzen 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 *stopspeed* empfohlen.

Syntax:

Ctrl-C entspricht Dezimal 3

Verwandter Befehl:

abort, ***Ctrl+D***, ***Ctrl+B***

Beispiel:

Ctrl-C
Der aktuell ausgeführte Befehl wird abgebrochen.

Ctrl-B

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***Ctrl-B*** wird der aktuell vom Interpreter ausgeführte Befehl abgebrochen, gleichzeitig werden alle Motorendstufen stromlos geschaltet.
Die Kommunikation mit der Steuerung ist weiterhin möglich.

Die Fortsetzung des normalen Betriebs ist erst nach einem Reset (Kommando ***reset***) oder einem Neustart der Steuerung möglich.

Syntax:

Ctrl-B entspricht Dezimal 4

Verwandter Befehl:

abort, ***Ctrl-C***, ***Ctrl-D***

Beispiel:

Ctrl-B
Der aktuell ausgeführte Befehl wird abgebrochen.
Die Motorendstufen werden stromlos geschaltet.

abort

Corvus TT	Corvus eco	Corvus PCI

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

Durch *ge* wird das Kommando ***abort*** blockiert und erst ausgeführt, nachdem das Kommando *100 0 0 move* und *ge* ausgeführt ist.

Syntax:

abort

Verwandter Befehl:

Ctrl-C

Beispiel:

abort

setinfunc

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***setinfunc*** konfiguriert die digitalen Eingänge für eine Sicherheitsabschaltung oder Begrenzung der Positionierrichtung.
Es lassen sich verschiedene Wirkungsweisen der Sicherheitsabschaltung konfigurieren.

- Die Positionierung wird für beide Richtungen gesperrt
- Die Positionierung ist für die negative Richtung gesperrt
- Die Positionierung ist für die positive Richtung gesperrt.

Für diese Funktion muss die Steuerung mit der Option "Input/Output" ausgestattet sein.
Details dazu, finden Sie in der Betriebsanleitung.

Die Funktion wirkt auf die programmierte und manuelle Bewegung. Die Motore bleiben bei der Abschaltung betromt. Wird die Abschaltung bei einer laufenden Positionierung ausgelöst, werden zunächst alle Achsen mit der eingestellten Verzögerung (Kommando ***sa***) angehalten. Danach gilt die mit ***setinfunc*** festgelegte Wirkungsweise auf die einzelnen Achsen.

Achsen für die keine Abschaltbedingung festgelegt wurde, können weiter positioniert werden.
Der Positionierbefehl für diese Achsen wird aber nicht ausgeführt, wenn gleichzeitig eine der abgeschalteten Achsen bewegt werden soll.
Wird das Abschaltsignal zurückgenommen, können die Achsen wieder normal positioniert werden.

Eine aktive ***setinfunc*** Bedingung wird im Statusbit D6 angezeigt. Zusätzlich meldet die LED-Diagnoseanzeige diesen Zustand durch Blinken der betreffenden Axis-LED.

Syntax:

[Wirkung] [Eingang] [Achse] ***setinfunc***

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

[Wirkung]	**Funktion**
0	keine Beschränkung
1	Die Achse wird für die positive Richtung gesperrt.
2	Die Achse wird für die negative Richtung gesperrt.
3	Die Achse wird für beide Richtungen gesperrt

Partnerbefehl:

getinfunc

Beispiel:

1 1 3 setinfunc
2 2 3 setinfunc

Für Achse-3 wurden folgende Abschaltbedingungen festgelegt:
Bei Abschaltsignal an Din-1 werden alle Achsen sofort gestoppt, danach kann die Achse-3 nur noch in negative Richtung positioniert werden.
Bei Abschaltsignal an Din-2 werden alle Achsen sofort gestoppt, danach kann die Achse-3 nur noch in positive Richtung positioniert werden.
Alle nicht limitierten Achsen können danach wieder normal positioniert werden.

getinfunc

Corvus TT	Corvus eco	Corvus PCI

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:

[Funktion]

	Wertebereich
[Funktion]	0, 1, 2, 3

[Funktion]	**Beschreibung**
0	keine Beschränkung
1	Die Achse wird für die positive Richtung gesperrt.
2	Die Achse wird für die negative Richtung gesperrt.
3	Die Achse wird für beide Richtungen gesperrt

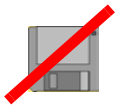
Beispiel:

1 3 getinfunc

setmp

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:



Mit dem Kommando ***setmp*** können die einzelnen Motorendstufen stromlos geschaltet werden. Alle anderen Funktionen der Steuerung bleiben aktiv.

Abhängig von der Lage und der Belastung des Motors kann sich der Rotor beim Ein- und Ausschalten maximal um eine Motorraststellung bewegen.

Syntax:

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

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

[Schalter]	**Funktion**
0	Der Motor ist stromlos
1	Der Motor ist normal bestromt

Beispiel:

0 1 setmp

Achse-1 wird stromlos geschaltet.

getmp

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getmp*** liest die mit *setmp* vorgenommene Einstellung zurück.

Syntax:

[Achse] ***getmp***

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

Rückmeldung:

[Schalter]

	Wertebereich
[Schalter]	0, 1

[Schalter]	Funktion
0	Der Motor ist stromlos
1	Der Motor ist normal bestromt

Beispiele:

1 setmp
Rückmeldung:
0

-1 getmp
Rückmeldung:
0 1 1

Position / Bezugspunkt / Koordinatensytem

pos (p)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***pos*** oder ***p*** liefert die aktuelle Position der Achsen bezogen auf den Koordinatennullpunkt zurück.
Mit der Einstellung *setdim* wird bestimmt welche Achsen angezeigt werden.
Der Befehl ***setnselpos*** legt fest, ob die intern kalkulierte Position oder die von einem externen Messsystem erfasste Position zurückgeliefert wird.

Syntax:

pos oder ***p***

Rückmeldung:

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

	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

setpdisplay

Beschreibung:

Mit dem Kommando ***setpdisplay*** kann das Format der Positionsanzeige im Terminal und Host mode eingestellt werden.
Die Positionsauflösung wird von dieser Einstellung nicht beeinflusst.

Syntax:

[VK] [NK] [Achse] ***setpdisplay***

	Funktion	Wertebereich
VK	Anzahl der Vorkommastellen	nicht begrenzt
NK	Anzahl der Nachkommastellen	nicht begrenzt
Achse		1, 2, 3

Partnerbefehl:

getpdisplay

Beispiel:

1 3 1 setpdisplay
2 10 2 setpdisplay

```
X:0.050
Y:50.0000000000
```

getpdisplay

Beschreibung:

Das Kommando ***getpdisplay*** liefert die Einstellung von *setpdisplay*.

Syntax:

[Achse] ***getpdisplay***

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

Rückmeldung:

Host mode:
[VK] [NK]

Terminal mode:

```
Axis: 1 field width: 1 precision width 3
Axis: 2 field width: 2 precision width 10
```

Beispiel:

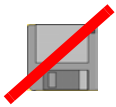
1 getpdisplay

setpos

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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:

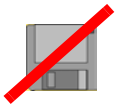
0 0 0 setpos
Die aktuelle Position wird bei allen Achsen der Koordinatennullpunkt

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.

align

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:



Mit ***align*** wird das orthogonale Koordinatensystem der Achse-1 und Achse-2 (X und Y) gedreht. Die Drehung erfolgt um den Nullpunkt.
Die Positionierbefehle *move* und *rmove* verwenden nach der Drehung das neue Koordinatensystem.
Das Koordinatensystem der 3. Achse wird nicht verändert.

Die Limits werden auch nach der Drehung überwacht und können nicht überschritten werden.

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

Um den Befehl ausführen zu können, muss die Dimension der Steuerung zuvor auf 2 eingestellt werden (*setdim*=2).

Syntax:

[0] [0] [OrgX] [OrgY] [X/Y] ***align***

Mit Eingabe der Koordinaten 0 0 | OrgX OrgY wird die Lage der Bezugsachse festgelegt.
X/Y legt fest ob die Bezugsachse die X- oder Y-Achse ist.

	Beschreibung
[OrgX]	X-Wert der Bezugsachse
[OrgY]	Y-Wert der Bezugsachse
[X/Y]	Bezugsachse 1 = X-Achse 2 = Y-Achse

	Wertebereich	**Einheit**
[OrgX]	+/- 16383mm	unit
[OrgY]	+/- 16383mm	unit
[X/Y]	1, 2	

Rückmeldung:

Das Kommando ***getico*** liefert den Wert 0 wenn das Koordinatensystem gedreht wurde.

.

Beispiel:

0 0 10 10 1 align

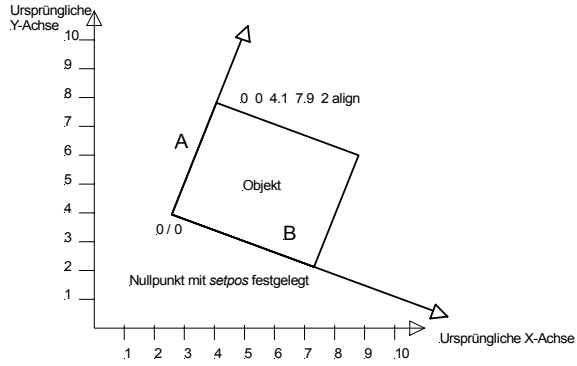
Die X-Achse des Koordinatensystems wurde um den Nullpunkt auf die Koordinaten 0 / 0 und 10 / 10 gedreht.
Die Lage der Y-Achse ergibt sich entsprechend.

Weitere Beispiele auf der folgenden Seite.

In der unten stehenden Zeichnung wird ein Objekt auf einem Mikroskop Scanningtisch dargestellt.
Das Objekt liegt gedreht auf dem Tisch.
Aufgabe ist es das Koordinatensystem der Steuerung an die Objektkoordinaten anzupassen.

Mit dem Befehl 0_0_4.1_7.9_2_***align*** wird die Objektkante A, die durch die Koordinaten 0 / 0 | 4,1 / 7,9 verläuft, zur Bezugsachse und als neue Y-Achse festgelegt.
Das Koordinatensystem der Steuerung wird damit gegen den Uhrzeigersinn um den Nullpunkt gedreht.
Die Steuerung ist damit an das Objektkoordinatensystem angepasst.

Wahlweise kann auch Objektkante B als Bezugsachse der neuen X-Achse festgelegt werden. Damit wäre die gleiche Anpassung erreicht.



ico

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***ico*** setzt ein gedrehtes Koordinatensystem wieder auf den Standardwert zurück.

Syntax:

ico

Partnerkommandos:

getico, align

Beispiel:

ico

getico

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Befehl ***getico*** wird überprüft ob das Koordinatensystem der Steuerung durch das Kommando ***align*** gedreht wurde.

Syntax:

getico

Rückmeldung

[Index]

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

	Wertebereich
[Index]	0, 1

Beispiel:

getico

Statusabfragen

status (st)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***status*** wird der augenblickliche Betriebszustand der Steuerung abgefragt.

Jedem Zustand wurde eine binärer Wert von D0 bis D8 zugeordnet. Treten mehrere Zustände gleichzeitig auf, wird dieser Wert addiert und als dezimaler Wert zurückgeliefert.

Für die Auswertung der Statusrückmeldung ist es notwendig den zurückgelieferten dezimalen Wert in eine binäre Zahl zu wandeln und die entsprechenden Stellen auszumaskieren.

Im Status werden folgende Zustände reflektiert:

Binär	Dezimal	Funktion
D0	1	Status der Befehlsausführung
D1	2	Status Joystick oder Handrad
D2	4	Status Taster A
D3	8	Maschinenfehler
D4	16	Status Speed Modus
D5	32	Status In-Window
D6	64	Status setinfunc
D7	128	Motorfreigabe od. Überstrom
D8	256	Joysticktaster gedrückt

D0:

0	Positionierung beendet
1	Positionierung wird ausgeführt

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 liegen im Zielfenster.

D6: Kommando *setinfunc*

0	Kein Eingangssignal
1	Eingangssignal aktiv

D7:

0	Motorendstufen aktiv.
1	Motorendstufen extern abgeschaltet

D8:

0	Joysticktaster unbetätigt.
1	Joysticktaster betätigt

Syntax:

status oder ***st***

Rückmeldung

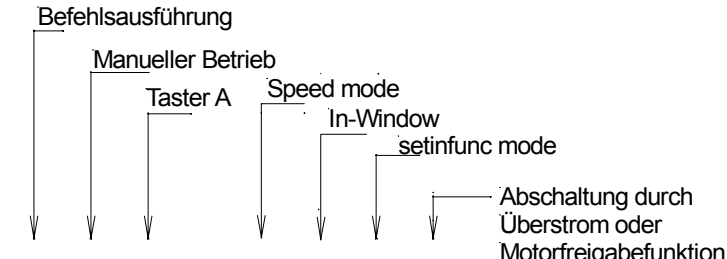
[Wert]

Beispiel:

st

Rückmeldung: **2**

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



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

geterror (ge)

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***geterror*** wird die Steuerung auf allgemeine Systemfehler überprüft. Es wird immer der zuletzt aufgetretene Fehler angezeigt.
Mit der Abfrage *ge* wird dieser Fehlereintrag gelöscht.

Syntax:

geterror

Rückmeldung:

[Fehlercode]

Fehler	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

getmerror (gme)

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***getmerror*** liest den Maschinenfehlerspeicher aus.
Dort werden bis zu 10 Fehler abgespeichert. Zur Kennzeichnung eines Maschinenfehlers wird Statusbit 3 gesetzt, siehe Kommando *status*, und die Error LED der Diagnoseanzeige aktiviert.
Beim Auslesen durch das Kommando ***gme*** wird immer der älteste Eintrag angezeigt und aus dem Fehlerspeicher gelöscht. Erst wenn alle Fehler entfernt sind, wird der Fehlerstatus zurückgesetzt. Die Error LED wird erst nach einem *reset* oder nach erneutem Einschalten der Steuerung deaktiviert.
Der Speicherinhalt wird nach Abschalten der Steuerung oder *reset* gelöscht.

Syntax:

getmerror oder ***gme***

Rückmeldung:

[Fehlercode]

Fehlercode	Beschreibung
0	Keine Maschinenfehler
1	Der Fehlerspeicher ist übergelaufen
10	Motor disable Funktion aktiv oder 12V Versorgung defekt
13	Überschreitung des maximalen Positionierfehlers im Closed Loop Betrieb Siehe Kommando setclpara (7. Parameter)
23	RS422 Encoder Fehler

Beispiel:

gme

gsp

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***gsp*** ermittelt die Anzahl der Elemente auf dem Parameter Stack.

Syntax:

gsp

Rückmeldung:

[Wert]

	Wertebereich
[Wert]	0 -99

Verwandter Befehl:

clear

Beispiel:

gsp

Rückmeldung:
2

getticks (gt)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***gt*** liefert die Anzahl der ausgeführten Prozessorzyklen zurück. Jeder Zähler entspricht 250µs. Nach 298 Stunden wird der Zähler zurückgesetzt.
Die Funktion kann als Zeitstempel für die Zuordnung von Daten oder Ereignissen genutzt werden.

Syntax:

gt

Beispiel:

gt

Liefert hier: 10922835

Entspricht: 10922835*250µs = 2730,708s

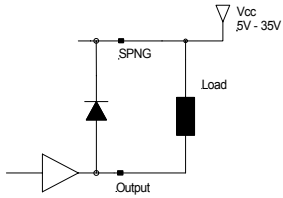
Input / Output Funktionen

setout

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***setout*** schaltet die Open Collector Ausgänge Dout1- Dout3.
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

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getout*** liest den mit ***setout*** eingestellten Ausgabewert zurück.

Syntax:

getout

Rückmeldung:

[bitcodiert]

Beispiel:

getout

setaout

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***setaout*** erzeugt eine analoge Ausgangsspannung zwischen 0 und 1000mV mit einer Auflösung von 8-Bit.
Es stehen zwei Ausgangskanäle zur Verfügung.

Es wird ein Lastwiderstand von 1 MOhm (oder höher) am Ausgang empfohlen. Insbesondere bei deutlich kleinerem Lastwiderstand kann ansonsten der ausgegebene Spannungswert erheblich nach unten variieren.

Syntax:

[Spannung] [Kanal] ***setaout***

	Wertebereich	Einheit
[Spannung]	0-1000 *	mV
[Kanal]	1,2	

*bei 1 MOhm Lastwiderstand

Partnerbefehl:

getaout

Beispiel:

100 1 setaout
An Kanal 1 wird an einer Last von 1 MOhm eine Ausgangspannung von 100 mV erzeugt.

getaout

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getaout*** liest die mit ***setaout*** eingestellte Ausgangsspannung zurück.

Syntax:

[Kanal] ***getaout***

Einheit= [mV]

Rückmeldung:

[Spannung]

Beispiel:

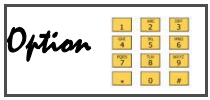
1 getaout

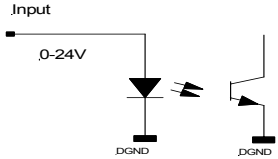
getin

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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





Syntax:

getin

Rückmeldung:

[bit-codierter Rückgabewert]

	Wertebereich
[bit-codiert]	0,1,2,3,4,5,6,7

[Rückgabe wert]	**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

Beispiel:

getin

Closed Loop Kommandos

setnselpos

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***setnselpos*** wird festgelegt ob die Steuerung die intern errechneten Positionsdaten (Soll-Position) oder die von einem Längenmesssystem erzeugten Positionsdaten (Ist-Position) zurückliefert.
Diese Einstellung gilt auch für den Terminal Mode.

Für die Darstellung der Ist-Position muss die Steuerung mit der Funktion "Closed Loop" ausgestattet 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

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Syntax:

[Achse] ***getnselpos***

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

Rückmeldung:

[Index]

	Wertebereich
[Index]	0,1

Beispiel:

3 getnselpos

setcloop

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setcloop*** aktiviert den Betrieb "geschlossenen Regelkreis". Die Steuerung verarbeitet in diesem Modus die Positionsdaten eines externen Längenmesssystem.

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

Corvus TT	Corvus eco	Corvus PCI

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

setclpara

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

speicherbar

Mit dem Kommando ***setclpara*** wird der Positionsregler für den Closed Loop Betrieb eingestellt.
Damit können **alle** Parameter mit einem einzigen Kommando übertragen werden.

Folgende Einstellungen sind möglich:

- Einstellung der P-I-D Regelparameter.
- Achsabschaltung bei Überschreitung einer vorgegebenen Positionsabweichung.
- Begrenzung der vom I-Regler erzeugten Nachregel-geschwindigkeit
- Dynamische Begrenzung des I-Anteils abhängig von der aktuellen Geschwindigkeit (anti wind up).

Syntax:

[**P**] [**I**] [**D**] [**16383**] [**SP5**] [**SP6**] [**dpos**] [**ivel**] [**cutoff**] [**SP10**] [**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), dem Achsenindex (Achse) sowie dem Kommando selbst abgeschlossen.
Bei der Änderung eines einzelnen Parametereintrages müssen alle Parameter die diesem Wert vorangestellt sind, ebenfalls übertragen werden.

Parameter	**Funktionsbeschreibung**
[P] (SP1)	P-Anteil des Positionsreglers. (Nur für Linearmotorantriebe zu verwenden)
[I] (SP2)	I-Anteil des Positionsreglers Reglerparameter für den Schrittmotorantrieb 
[D] (SP3)	D-Anteil des Positionsreglers. (Nur für Linearmotorantriebe zu verwenden)
[16383] (SP4)	Diese Werkseinstellung darf nicht verändert werden.
[SP5]	Boost-Faktor (Nur für Linearmotorantriebe zu verwenden)
[SP6]	Lastwinkelvorgabe (Nur für Linearmotorantriebe zu verwenden)

[dpos] (SP7)	Maximal erlaubter Positionsfehler. Bei Überschreitung erfolgt die Abschaltung der Achse. Maschinenfehler 13 wird gesetzt. Mit der Einstellung = 0 ist die Funktion abgeschaltet.
[ivel] (SP8)	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] (SP9)	Diese Funktion wird eingesetzt um die Laufruhe im geschlossenen Regelkreis und das Einschwingverhalten in die Zielkoordinate zu optimieren. Mit cutoff wird der I-Anteil abhängig von der aktuellen Geschwindigkeit reduziert. Der I-Anteil hat zu Beginn der Fahrt den maximalen Wert, diese Wert wird bis zur cutoff Geschwindigkeit dynamisch auf ein Minimum reduziert. Unterhalb der cutoff Geschwindigkeit ist der Einfluss des I-Anteils wieder maximal. Mit der Einstellung = 0 ist diese Funktion abgeschaltet.
[SP10]	Keine Funktion
[Achse]	Achsenindex

Parameter	Wertebereich	Default	Einheit
P		0	-
I		10	1/s
D		0	s
16383	Darf nicht ver-ändert werden	16383	
SP1		0	
SP2		0	
dpos		0 0 = aus	mm
ivel		2 0 = aus	mm/s
cutoff		2 0 = aus	mm/s
SP3	0	0	
Achse	1, 2, 3		

Ab Firmware Version 4.50 gibt es die Möglichkeit die Parameter der Reglereinstellung auch einzeln anzugeben.
Siehe Kommando *sp*

Partnerbefehle:

getclpara, setcloop, setclperiod, sp

Beispiele:

0_20_2_1_ setclpara

An Achse-1 werden die Einstellungen für P und I übertragen. Alle weiteren Parameter bleiben unverändert.

0_15_0_16383_0_0_1_2_2_9_3_setclpara

Bei Achse-3 wurden neun Parameter übertragen.

getclpara

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getclpara*** liest die Einstellung des Positionsreglers der Achse zurück.
Dabei 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

setsp	Corvus TT	Corvus eco	Corvus PCI

Gültig ab Firmwareversion 4.50

Beschreibung:

Mit ***setsp*** ist es möglich die Parameter des Closed Loop Reglers einzeln zu übertragen.
Die Funktionalität entspricht der von Kommando *setclpara*.

Folgende Einstellungen sind möglich:

- Einstellung der P-I-D Regelparameter.
- Achsabschaltung Überschreitung einer vorgegebenen Positionsabweichung.
- Begrenzung der vom I-Regler erzeugten Nachregel-geschwindigkeit
- Dynamische Begrenzung des I-Anteils abhängig von der aktuellen Geschwindigkeit (anti windup).

Syntax:

[**SP 1-10**] [**Achse**] ***setsp***

Parameter	Funktionsbeschreibung
[SP1] (P)	P-Anteil des Positionsreglers. (Nur für Linearmotorantriebe zu verwenden)
[SP2] (I)	I-Anteil des Positionsreglers Reglerparameter für den Schrittmotorantrieb 
[SP3] (D)	D-Anteil des Positionsreglers. (Nur für Linearmotorantriebe zu verwenden)
[SP4] (16383)	Diese Werkseinstellung darf nicht verändert werden.
[SP5]	Boost-Faktor (Nur für Linearmotorantriebe zu verwenden)
[SP6]	Lastwinkelvorgabe (Nur für Linearmotorantriebe zu verwenden)
[SP7] (dpos)	Maximal erlaubter Positionsfehler. Bei Überschreitung erfolgt die Abschaltung der Achse. Maschinenfehler 13 wird gesetzt. Mit der Einstellung = 0 ist die Funktion abgeschaltet.
[SP8] (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.

Parameter	Funktionsbeschreibung
[SP9] (cutoff)	Diese Funktion wird eingesetzt um die Laufruhe im geschlossenen Regelkreis und das Einschwingverhalten in die Zielkoordinate zu optimieren. Mit cutoff wird der I-Anteil abhängig von der aktuellen Geschwindigkeit reduziert. Der I-Anteil hat zu Beginn der Fahrt den maximalen Wert, diese Wert wird bis zur cutoff Geschwindigkeit dynamisch auf ein Minimum reduziert. Unterhalb der cutoff Geschwindigkeit ist der Einfluss des I-Anteils wieder maximal. Mit der Einstellung = 0 ist diese Funktion abgeschaltet.
[SP10]	Keine Funktion
[Achse]	Achsenindex

Parameter	Bereich	Default	Einheit
SP1 (P)		0	-
SP2 (I)I		10	1/s
SP3 (D)		0	s
SP4 (16383)		16383 (maximaler Wert)	
SP5		0	
SP6		0	
SP7 (dpos)		0 0 = aus	mm
SP8 (ivel)		2 0 = aus	mm/s
SP9 (cutoff)		2 0 = aus	mm/s
SP10	0	0	
Achse	1, 2, 3		

Partnerbefehle:

setclpara, getsp

Beispiel:

100 2 setsp

Der I-Anteil des Reglers von Achse-1 wird auf 100 1/s eingestellt.

getsp

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***getsp*** können die Reglerparameter einzeln ausgelesen werden

.

Syntax:

[SP Index] [Achse] ***getsp***

	Wertebereich
[Achse]	1, 2, 3
[SP Index]	1 - 10

Rückmeldung:

[**SP**]

Beispiel:

1 getsp

Rückmeldung:
100

setscaleinterface

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Mit dem Kommando ***setscaleinterface*** wird das Closed Loop Interface ausgewählt.

gespeichert nach "reset"

Corvus hat eine On-Board digitale Closed Loop Schnittstelle für RS-422 Quadratursignale und kann mit einem Zusatzmodul (sin/cos Modul) auch analoge Positionsmesssysteme mit 12Bit oder 16Bit Auflösung auswerten

Die Einstellung muss mit ***save*** gespeichert werden. Danach ist ein ***reset*** auszuführen.

Syntax:

[Index] [Achse] ***setscaleinterface***

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

[Index]	**Beschreibung**
0	Quadratur Interface abgeschaltet
1	Quadratur Interface eingeschaltet
2	Analoges Messinterface (sin/cos Modul) eingeschaltet

Beispiel:

2 1 setscaleinterface

Das analoge Messinterface für Achse-1 wird initialisiert.

getscaleinterface

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***getscaleinterface*** überprüft welche der beiden Closed Loop Schnittstellen initialisiert ist.

Syntax

[Achse] ***getscaleinterface***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1, 2

Beispiel:

1 getscaleinterface

Rückgabe:
1

setscaletype

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***setscaletype*** wird das Messinterface der Steuerung an den Typ des Messsystems angepasst.

Die Einstellung unterscheidet:
- lineare Messsysteme sowie analoge Drehgeber
- digitale Drehgeber

.

Syntax:

[Index] [Achse] ***setscaletype***

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

Index	**Beschreibung**
0	lineares Messsystem (analog oder digital) analoger Drehgeber
1	digitaler Drehgeber mit RS-422 Ausgang

Beispiel:

1 1 setscaletype

Adaptiert das Messinterface der Achse-1 für einen digitalen Drehgeber mit RS-422 Ausgang.

getscaletype

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getscaletype*** überprüft für welchen Typ Messsystem das Messinterface initialisiert ist.

Syntax

[Achse] ***getscaletype***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1

Beispiel:

1 getscaletype

Rückgabe:
1

setclfactor

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***setclfactor*** wird für die Anpassung an einen digitalen Drehgeber verwendet. Damit wird Strichauflösung und Zählrichtung des Gebers übernommen. Der Strichwert entspricht der Anzahl der Impulse, die der Drehgeber bei einer Motorumdrehung erzeugt.
Die Zahlrichtung wird durch das Vorzeichens festgelegt.

Für den Anschluss eines digitalen Drehgebers muss das RS-422 Messinterface der Steuerung freigeschaltet sein. Damit ergibt sich folgende Einstellungen für die Parameter scaletype und scaleinterface
:

Kommando	Wert
setscaletype	0
setscaleinterface	1

Syntax:

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

Parameter	Wertebereich	Einheit
Zählrichtung	+/-	Vorzeichen
Impulse	1 - 50000	
Achse	1, 2, 3	

Partnerbefehl:

getclfactor, setscaletype, setscaleinterface

Beispiel:

- 500 3 setclfactor

getclfactor

Corvus TT	Corvus eco	Corvus PCI

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	1 - 50000	Impulse/Umdrehung

Beispiel:

1 getclfactor

Rückmeldung:

-500

setclperiod

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Befehl ***setclperiod*** wird das analoge und digitale Messinterface der Steuerung an lineare Messsysteme mit Ausgängen RS-422, 1Vss oder MR, sowie Drehgebern mit 1Vss Ausgang angepasst.
Für digitale Drehgeber gibt es hierfür das Kommando ***setclfactor***.

Der Wert von ***setclperiod*** entspricht der resultierenden Strecke in mm beim Durchlauf einer Massstabsperiode.

Bei analogen Drehgebern muss die Spindel- und Getriebeübersetzung bei der Berechnung von ***setclperiod*** berücksichtigt werden.
Die Zählrichtung des Messeingangs wird durch das Vorzeichen angepasst.

Syntax:

[Zählrichtung] [Weg/Periode] [Achse] ***setclperiod***

	Wertebereich	Einheit
[Zählrichtung]	+,-	
[Weg/Periode]	0.0000001-1.999999	mm
[Achse]	1, 2, 3	

Wichtige Befehle:

getclperiod, setscaleinterface, setscaletype

Beispiel:

- 0.002 3 setclperiod

Beispiel: Lineare Messsysteme

Ein Spindelantrieb mit 10 mm Spindelsteigung ist mit einem linearen Messsystem ausgestattet. Das Messsystem hat eine Teilungsperiode von 20µm und liefert eine Ausgangs-spannung von 1Vss.
Antriebsmechanik und Messsystem sind hier direkt gekoppelt. Die Strecke der Maßstabsperioden entspricht damit der resultierende Strecke des Antriebs.

Die Einstellung von ***setclperiod*** ist folglich: 0.020 mm

Hat die gleiche Anordnung zusätzlich ein Getriebe 5:1, muss lediglich die Spindelsteigung auf 2mm eingestellt werden. Der Wert für ***setclperiod*** ändert sich nicht, da die resultierende Strecke pro Massstabsperiode gleich bleibt

.

Beispiel: Drehgeber

Ein Spindelantrieb mit 10mm Spindel ist mit einem analogen Drehgeber zur Positionsrückmeldung ausgestattet. Der Drehgeber ist direkt auf der Motorwelle montiert und erzeugt 1000 Schwingungsperioden pro Umdrehung:

Die von dem Spindelantrieb erzeugte Strecke stimmt hier nicht mit der Strecke der Maßstabsperioden überein sondern ist abhängig von der Spindelsteigung bzw. Getriebeübersetzung.

Abgeleitet daraus einige Berechnungsbeispiele:

Motor mit Spindel und Drehgeber auf Motorwelle:

setpitch = 10mm
setclperiod = 10mm / 1000 Perioden = 0.01mm

Motor mit Getriebe 120: 1, Drehgeber auf Motorwelle:

setpitch = 1 / 120 = 0.00833mm
setclperiod = 0.00833 / 1000 = 0.00000833mm

Motor mit Getriebe 120: 1, Drehgeber am Getriebe:

setpitch = 1 / 120 = 0.00833mm
Durch die direkte Kopplung von Messsystem und Getriebeabtrieb ist die resultierende mechanische Strecke und die von den Maßstabsperioden erzeugte Strecke gleich.
setclperiod = 1 / 1000 * 10 = 0.001mm

Beispiel: Rundtisch mit Drehgeber in der Einheit Grad (°)

Ein Rundtisch mit Getriebe soll in der Einheit Grad(°) positioniert werden.
Der Drehgeber ist auf der Motorwelle montiert und liefert 18000 Perioden pro Umdrehung. Das Getriebe hat eine Untersetzung 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 Maßstabsperioden.
Resultierender Winkel = 3° (360 / 120)

Winkel pro Massstabsperiode:
Signalperiode = 3° / 18000 = 0.0001666

setclperiod = 0.0001666

getclperiod

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

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

Syntax:

[Achse] ***getclperiod***

Parameter	Wertebereich
[Achse]	1, 2, 3

Rückmeldung

[Strecke / Massstabsperiode]

	Wertebereich	Einheit
[Strecke/Massstabsperiode]	0.0000001-1.999999	mm

Beispiel:

1 getclperiod

Rückmeldung:

0.0001

refmove

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***refmove*** positioniert alle aktiven Achsen zur Referenzmarkierung des Längenmesssystems.
Mit dem 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***.

Für diese Funktion muss die Steuerung mit der Option "Closed Loop" ausgestattet sein.
Details dazu finden Sie in der Betriebsanleitung.

Notwendige Closed Loop Einstellungen:

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

refmove ist ein sperrendes Kommando.

Syntax:

[Richtung] [Strecke] ***refmove***

	Wertebereich	Einheit
[Richtung]	+ / -	
[Strecke]	+/-1000	Umdrehungen

Verwandte Befehle:

setref, setrefvel, getrefst

Beispiele:

100 refmove

Die Steuerung bewegt alle aktiven Achsen 100 Motorumdrehungen in positive Richtung. Die Achse stoppt mit der eingestellten Systembeschleunigung wenn die Referenzmarke erkannt wurde.

-7 refmove

Die Steuerung bewegt alle aktiven Achsen Achse 7 Motorumdrehungen in negative Richtung. Die Achse stoppt mit der eingestellten Systembeschleunigung wenn die Referenzmarke erkannt wurde.

setclwindow

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***setclwindow*** definiert ein Positionszielfenster für den Closed Loop Betrieb.
Innerhalb dieses Fensters wird keine Positionsregelung ausgeführt. Bewegt sich die Achse aus dem Zielfenster hinaus, setzt die Regelung wieder ein.
Befinden sich alle beteiligten Achsen innerhalb ihres festgelegen Zielfensters, wird Statusbit D5 gesetzt.

So lange sich die Position einer Achse ausserhalb ihres Zielfensters befindet, blinkt die der Achse zugeordnete LED in der Diagnoseanzeige.
Wird das Fensterbreite auf 0 gesetzt, ist die Funktion abgeschaltet und die Achse wird permanent geregelt.

Syntax:

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

	Wertebereich	Einheit	Funktion
[Window]		User	+/- Fenster aktiv
[Window]	0 *	User	Funktion inaktiv
[Achse]	1, 2, 3	-	

*werkseitige Einstellung

Partnerbefehl:

getclwindow

Beispiel:

0.001 1 setclwindow
User Einheit der Achse ist mm.
Für Achse-1 wird ein Positionszielfenster von +/-1µm eingestellt.

getclwindow

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

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

[

Syntax:

[Achse] ***getclwindow***

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

Rückmeldung:

[Wert]

	Wertebereich	**Einheit**
[Wert]		User Einheit

Beispiel:

1 getclwindow

Rückmeldung:
0.001

-1 getclwindow

Rückmeldung
0.001 0.002 0.000

setref

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setref*** schaltet die Auswertung des Referenzsignals aktiv und bestimmt die Flankenauswertung.

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, setrefvel

Beispiel:

0 1 setref

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

getref

Corvus TT	Corvus eco	Corvus PCI

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

Trigger-Output-Funktionen

getrefst

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getrefst*** liefert das Ergebnis der Referenzfahrt (***refmove***).
Diese Abfrage sollte immer nach der Referenzfahrt ausgeführt werden.
In der binären Darstellung der Rückmeldung sind folgende Betriebszustände geschlüsselt.

- D0 = Referenzmarke gefunden oder nicht gefunden
- D1 = Endschalter betätigt oder nicht betätigt
- D2 = Aktueller Signalpegel des Referenzsignals

Kodierung	Wertebereich
D0 = 0	Referenzmarke gefunden
D0 = 1 (1)	Referenzmarke nicht gefunden
D1 = 0	Endschalter nicht betätigt
D1 = 1 (2)	Endschalter betätigt
D2 = 0	Positiver Signalpegel
D2 = 1 (4)	Nullpegel

Syntax:

[Achse] ***getrefst***

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

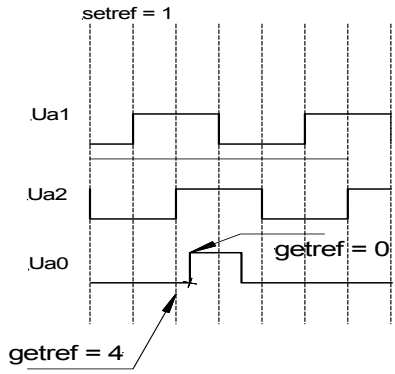
Rückmeldung:

[dezimaler Wert]

Typische Werte
0, 1, 3, 4, 5

Beispiele:

	Beschreibung
0	Referenzmarke gefunden, positiver Pegel
1	Referenzmarke nicht gefunden
3	Referenzmarke nicht gefunden, Endschalter betätigt
4	Referenzmarke gefunden, Nullpegel
5	Referenzmarke nicht gefunden, Nullpegel



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

Beispiel:

2 getrefst
Liefert das Ergebnis der Referenzfahrt von Achse-2.

outtrig (ot)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***outtrig*** besteht die Möglichkeit einen Triggerimpuls an einem der drei digitalen Ausgänge zu erzeugen.
Bei mehreren ***outtrig*** Kommandos in Folge, werden diese in einem FIFO gespeichert und nacheinander ausgeführt.

Für diese Funktion wird die Option "Digital I/O" benötigt.

Syntax:

[time] [pol] [output] ***outtrig***

	Beschreibung
[time]	Pulsbreite Triggersignal
[pol]	Polarität Triggersignal
[output]	Ausgang Nr.

	Bereich	**Einheit**	**Funktion**
[time]	1-1000	ms (ganzzahlig)	
[pol]	0, 1		0 = aktiv 1 = aktiv
[output]	1, 2, 3		Ausgang Nr.

Beispiel:

100 1 1 ot
Es wird ein Triggerimpuls mit einer Pulsbreite von 100ms am digitalen Ausgang 1 erzeugt.

waitposot (wpot)

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***wpot*** ermöglicht die Generierung von Triggerimpulsen an beliebigen absoluten Positionen.
Eine komplette Triggersequenz besteht aus einem move Kommando und der Angabe der absoluten Triggerkoordinaten, durch ein oder mehrere ***wpot*** Kommandos.

Die maximale Ausgangsfrequenz beträgt 2 kHz. Der Ortsfehler des Triggerimpulses ist davon abhängig mit welcher Geschwindigkeit die Achse bewegt wird.
Der maximale Ortsfehler ergibt sich aus der Strecke, welche die Achse innerhalb von 500 µs zurücklegt.

Die Funktion arbeitet auch im manuellen Betrieb.

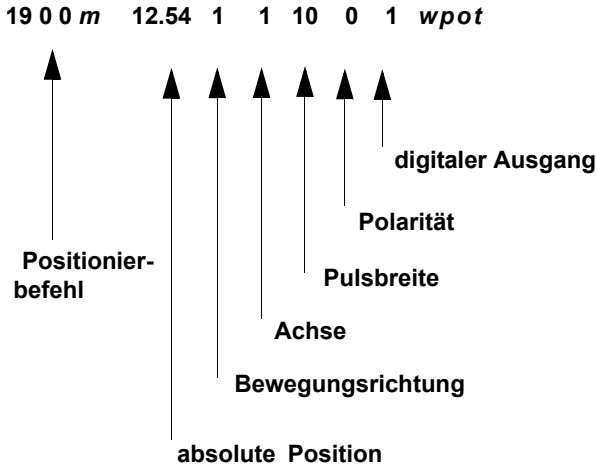
Für diese Funktion muss die Steuerung mit der Option "Digital I/O" ausgestattet sein.

Mit dem Kommando ***setotmode*** kann der wpot Trigger auf die Ist- oder Sollposition bezogen werden.
Zusätzlich ist es möglich den Triggerausgang als Eingang für das Abspeichern von Positionsdaten (siehe Kommando ***setpc***) zu konfigurieren.

.

Syntax:

X Y Z move [pos] [dir] [axis] [time] [pol] [output] ***wpot*** *[crlf]*



	Beschreibung
[pos]	absolute Triggerposition
[dir]	Bewegungsrichtung
[axis]	Achse
[time]	Pulsbreite des Triggersignals
[pol]	Polarität des Triggersignals (active low, active high)
[output]	Nummer des digitalen Ausgangs

	Bereich	Einheit	Funktion
[pos]	+/-16383	User	
[dir]	0, 1		0 = negative Richtung 1 = positive Richtung
[axis]	1, 2, 3		
[time]	1-1000	ms	
[pol]	0, 1		0 = active low 1 = active high
[output]	1, 2, 3		

Beispiele:

12_20_3_m_12.4_1_2_200_1_2_wpot_ *[cr lf]*

Die drei Achsen werden zur den Koordinaten 12/20/3 bewegt. Sobald die Achse-2 die Koordinate 12.4 erreicht hat, wird ein Triggerimpuls von 200ms Länge am digitalen Ausgang-2 erzeugt.

Die Funktion ***wpot*** innerhalb eines Makros:

```
beginmakro
0 100 0 m
12.4 1 2 200 1 2 wpot
22.2 1 2 200 1 2 wpot
30.9 1 2 200 1 2 wpot
41.0 1 2 200 1 2 wpot
53.2 1 2 200 1 2 wpot
60.0 1 2 200 1 2 wpot
77.9 1 2 200 1 2 wpot
89.9 1 2 200 1 2 wpot
90.8 1 2 200 1 2 wpot
endmakro
```

waitpos (wp)

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***waitpos*** wird die Ausführung der nachfolgenden Venus-1 Befehle so lange verzögert, bis die spezifizierte Achse eine angegebene Koordinate erreicht hat. Dieses Kommando wird vorzugsweise zur Makrosteuerung verwendet.

Mit ***Ctrl-C*** wird das Kommando abgebrochen.

.

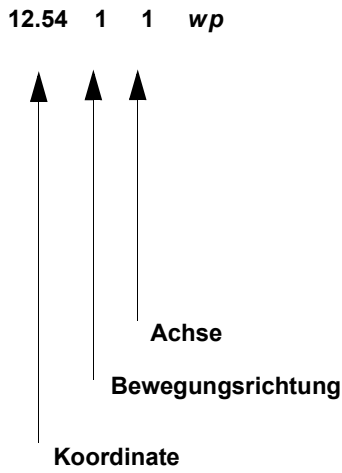
Syntax:

[pos] [dir] [Achse] ***waitpos***

	Beschreibung
[pos]	Koordinate
[dir]	Fahrtrichtung
[Achse]	Achse

	Bereich	**Einheit**	**Funktion**
[pos]	+/-16383	unit	Einheit der Achse
[dir]	0, 1		0 = negative Richtung 1 = positive Richtung
[Achse]	1, 2, 3		Achse

Beispiele:



Das Kommando wp innerhalb eines Makros:

beginmakro
100 10 m
12.54 1 1 wp
100 1 1 ot
endmakro

Mit dem Befehl *startmakro*
werden die Achsen zur den Koordinaten
100 mm (Achse-1) und 10 mm (Achse-2) bewegt.
Sobald Achse-1 die Koordinate 12.54 mm erreicht hat,
wird ein Triggerimpuls von 100ms am digitalen Ausgang-1
erzeugt. Siehe auch Kommando *wpot*.

oder:

100 10 m *[SP]* ***12.54 1 1 wp*** *[SP]* ***100 1 1 ot*** *[CRLF]*

waittime (wt)

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***wt*** sperrt den Kommandointerpreter für eine festgelegte Zeit.
Während der Sperrung , kann die Steuerung weiterhin Daten empfangen, diese werden aber erst nach Ablauf der Wartezeit ausgeführt.
Das aktuell ausgeführte Kommando wird mit ***waittime*** nicht unterbrochen.
Der manuelle Betrieb wird mit ***wt*** ebenfalls gesperrt.

Syntax:

[Wartezeit] [Zeiteinheit] ***wt***

	Beschreibung
[Wartezeit]	Der Interpreter ist für diesen Zeitraum gesperrt
[Zeiteinheit]	Einheit der Wartezeit

	Bereich	**Funktion**
[Wartezeit]	Integer	
[Zeiteinheit]	0	0 = Ticks (1 Tick = 250µs)
	1	1 = Sekunden (s)

Beispiele:

1000 0 wt *[SP]* ***ge*** *[SP]* ***st***
Das ***wt*** Kommando sperrt den Interpreter für 1000 x 250µs danach wird das Kommando ***ge*** und ***st*** ausgeführt.

1 1 wt *[SP]* ***st***
Das ***wt*** Kommando sperrt den Interpreter für 1s danach wird das Kommando ***st*** ausgeführt.

waitintrigot (witot)

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***witot*** (wait_in_trigger out trigger) ist eine schnell ausführbare Kombination aus den Kommandos ***wit*** und ***ot***.
Siehe Beschreibung der einzelnen Funktionen.

Für diese Funktion muss die Steuerung mit der Option "Digital Input/Output" ausgestattet sein.

Syntax:

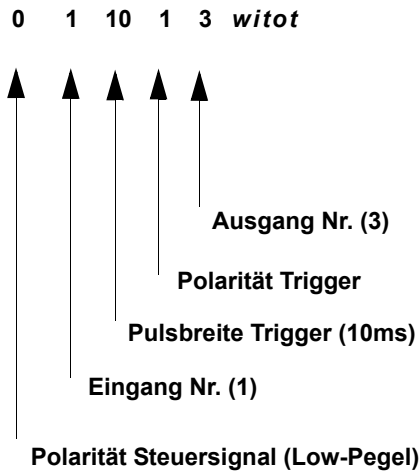
[pol_in] [input] [time] [pol_out] [output] ***witot***

	Beschreibung
[pol_in]	Polarität Steuersignal
[input]	Nr. digitaler Eingang
[time]	Pulsbreite Triggerausgangssignal
[pol_out]	Polarität Triggerausgangssignal
[output]	Nr. digitaler Ausgang

	Bereich	Einheit	Funktion
[pol_in]	0, 1		0= Funktion bei Low-Pegel aktiv 1 =Funktion bei High-Pegel aktiv
[input]	1, 2, 3		
[time]	1-1000	ms ganz-zahlig	
[pol_out]	0, 1		
[output]	1, 2, 3		

Beispiel:

0 1 10 1 3 witot



waittimeot (wtot)

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Mit dem Kommando ***wtot*** (wait_time out_trigger) wird ein Triggersignal verzögert ausgegeben.
wtot ist eine Kombination aus den Kommandos ***waittime*** (***wt***) und ***outtrig*** (***ot***). Diese wurden zusammengefasst um eine noch schnellere Befehlsausführung für Echtzeit-anwendungen zu erreichen.
Details finden Sie in der Beschreibung dieser Kommandos.

Syntax:

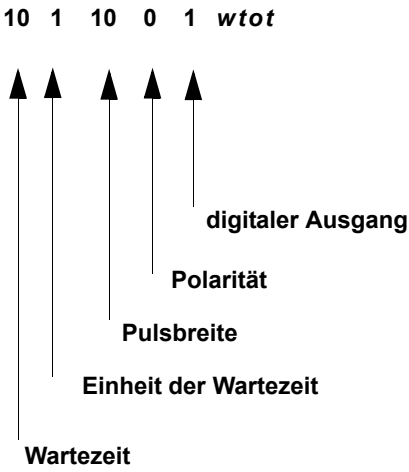
[time_wait] [wait_unit] [time_trigger] [pol_out] [output] ***wtot***

	Beschreibung
[time_wait]	Wartezeit
[wait_unit]	Einheit der Wartezeit
[time_trigger]	Triggerpulsbreite
[pol_out]	Polarität Triggersignal
[output]	Triggerausgang

	Bereich	**Einheit**	**Funktion**
[time_wait]	Integer value	wait_ unit	
[wait_unit]	0, 1		0 = Ticks 1 Tick = 250µs 1 = Sekunden (s)
[time_trigger]	0-1000	ms	
[pol_out]	0, 1		
[output]	1, 2, 3		Digit. Ausgang

Beispiel:

10 1 10 0 1 wtot



Ein 10ms Triggerimpuls wird mit einer Verzögerung von 10 Sekunden ausgegeben.

setrptdata

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Das Kommando ***setrptdata*** initialisiert den Positions-Intervall-Trigger mit dem Triggerausgangssignale in gleichen Abständen erzeugt werden können.
Die Triggerposition bezieht sich dabei wahlweise auf die kalkulierte Soll-Position oder auf die gemessene Ist-Position.

Für diese Funktion muss die Steuerung mit der Option "Digital I/O" ausgestattet sein.

.

Syntax:

[rpos] [axis] [time] [pol] [output] [selpos] ***setrptdata*** *[crlf]*

1.234 1 100 1 0 1 *setrptdata*

Triggerabstand

Achse

Pulsbreite

Polarität

Ausgang

Ist- od. Soll-Position

	Beschreibung
[rpos]	Triggerabstand in der eingestellten Achseinheit
[axis]	Bezugsachse
[time]	Triggerpulsbreite
[pol]	Polarität des Triggersignals
[output]	Digitaler Ausgang
[selpos]	Triggerposition intern oder vom Messsystem

	Bereich	Einheit	Funktion
[rpos]	+/-16383mm	user	
[axis]	1, 2, 3		
[time]	0,25-16383	ms	
[pol]	0, 1		0 = active low 1 = active high
[output]	1, 2, 3		
[selpos]	0, 1		0 = Soll-Position 1 = Ist-Position

Beispiel:

1.234 2 100 1 2 0 setrptdata

Der Positions-Intervall-Trigger wird hier wie folgt initialisiert:
Der Triggerabstand beträgt 1,234 units (Beispiel.: mm, µm, etc.) Die Bezugsachse für die Triggerimpulse ist Achse-2
Die Triggerimpulsbreite beträgt 100ms
Der Trigger wird am digitalen Ausgang-2 ausgegeben.
Die Triggerposition bezieht sich auf die Sollposition.

Vollständige Kommandosequenz zum Erzeugen eines aquidistanten Triggerausgangssignals:
1.234 2 100 1 2 0 setrptdata
0 10 startrpt 20 20 m

getrptdata

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Das Kommando ***getrptdata*** liefert die Parameterwerte der Funktion "Positions-Intervall-Trigger".

Syntax:

getrptdata

Rückmeldung:

[rpos] [axis] [time] [pol] [output] [selpos]

	Beschreibung
[rpos]	Triggerabstand in der eingestellten Achseinheit
[axis]	Bezugsachse
[time]	Triggerpulsbreite
[pol]	Polarität des Triggersignals
[output]	Digitaler Ausgang
[selpos]	Triggerposition Soll-Position oder gemessene Ist-Position

Beispiel:

getrptdata
Rückmeldung:
1.234 2 100 1 2 0

startrpt

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Das Kommando ***startrpt*** aktiviert den Positions-Intervall-Trigger und gibt die Start- und Stop-Koordinate des Triggerintervalles an.
Mit dem Erreichen der Stop-Koordinate wird der Trigger abgeschaltet. Für die Ausgabe weiterer Pulse muss dieser wieder mit neuer Start- u. Stop-Koordinate aktiviert werden.

Für diese Funktion muss die Steuerung mit der Option "Digital I/O" ausgestattet sein.

Syntax:

[Start] [Stop] ***startrpt*** *[crlf]*

	Beschreibung
[Start]	Koordinate der Bezugsachse an der die Triggerausgabe gestartet wird
[Stop]	Koordinate der Bezugsachse an der die Triggerausgabe beendet wird

	Bereich	Einheit
[Start]	+/-16383mm	user
[Stop]	+/-16383mm	user

zugehörige Kommandos:

setrptdata, getrptdata, setotmode

Beispiele:

10.234 12.56 startrpt

Die Triggersequenz beginnt mit Koordinate 10.234 und endet bei Koordinate 12.56.

Komplette Kommandosequenz für die Ausgabe eines Positions-Intervall-Trigger am digitalen Ausgang-1 mit einem Abstand von 10µm.
Bezugsachse für den Trigger ist Achse-2:

0.010 2 0.5 0 1 0 setrptdata
12.234 15.23 startrpt
4.5 16.0 move

Sobald Achse-2 die Koordinate 12.234 erreicht hat, werden im Abstand von 10µm Triggerimpulse mit der Breite 0.5ms ausgegeben. Mit Erreichen der Koordinate 15.23 wird die Ausgabe beendet.
Die Triggerkoordinaten beziehen sich auf die Sollposition.

Trigger-Input Funktionen

setotmode

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Mit dem Kommando ***setotmode*** wird für das Kommando *wpot* die Soll- oder Ist-Position als Triggerquelle festgelegt.

Zusätzlich besteht die Möglichkeit den mit *wpot* initialisierten Absoluttrigger als Eingangstrigger für die Funktion *setpc* zu verschalten. Damit wird die zum Zeitpunkt der Triggerausgabe gültige Soll- oder Ist-Position geloggt.

Syntax:

[mode] ***setotmode***

[mode]	Triggerposition	Logging
0	Soll-Position	aus
1	Ist-Position *	aus
2	Ist-Position *	ein
3	Soll-Position	ein

* Die Steuerung muss mit der Option Closed Loop ausgestattet sein

Beispiel:

3 setotmode

getotmode

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getotmode*** liefert die mit setotmode eingestellten Einstellungen für die Funktionen ***wpot*** und setpc zurück.

Syntax:

getotmode

Rückmeldung:

[mode]

	Wertebereich
[mode]	0, 1, 2, 3

Beispiel:

getotmode

setpcin

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Das Kommando ***setpcin*** initialisiert den Triggereingang für die Funktion "position capture".
Mit dieser Funktion werden aktuelle Positionswerte, gesteuert von einem Triggereingangssignal oder einem Triggerausgangssignal (siehe *setotmode*) in den Positionsspeicher übernommen.

Für diese Funktion muss die Steuerung mit der Option "Digital Input/Output" ausgestattet sein.
Details dazu finden Sie in der Betriebsanleitung.

Syntax:

[Flanke] [Eingang] ***setpcin***

[Flanke]	Beschreibung
0	Die Daten werden mit der steigenden Flanke übernommen
1	Die Daten werden mit der fallenden Flanke übernommen

[Eingang]	Beschreibung
1	Eingang DIN 1 (Pin 6)
2	Eingang DIN 2 (Pin 2)
3	Eingang DIN 3 (Pin 7)

Beispiel:

1 3 setpcin
Für die Funktion "position capture" wurde Eingang 3 festgelegt. Die Daten werden mit der steigenden Flanke in das "capture memory" übernommen.

getpcin

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getpcin*** liefert die Einstellungen der Funktion "position capture".

Syntax:

getpcin

Rückmeldung: [Flanke] [Input]

	Wertebereich
[Flanke]	0,1
[Input]	1, 2, 3

Beispiel:

getpcin
Rückmeldung:
1 3

setpc

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setpc*** aktiviert oder deaktiviert die Funktion "position capture",

Mit dem Kommando *setnselpos* kann festgelegt werden, ob die intern berechnete Position (Nominalposition) oder die von einem Messsystem gelieferte Ist-Position abgespeichert wird.

Wird die Einstellung von *setnselpos* geändert, muss das Kommando *setpc* erneut ausgeführt werden.

Syntax:

[Ein/Aus] ***setpc***

[Ein/Aus]	Beschreibung
0	Positionsdatenerfassung aus
1	Positionsdatenerfassung ein

Beispiel:

1 setpc

getpc

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getpc*** liefert den Status der Funktion "position capture". Zusätzlich wird die Anzahl der empfangenen Triggersignale angezeigt.

Mit jedem Triggerimpuls wird die Position aller aktiven Achsen, sowie die internen Systemzeit abgespeichert.
Für diesen Datensatz wird im "capture memory" ein Speicherplatz verbraucht.
Die Datensätze werden mit dem Zählerstand des Triggerzählers indiziert und sind mit diesem Index abrufbar.
Insgesamt können 65000 Werte indiziert werden.
In den Speicher können maximal 1000 Datensätze übernommen werden. Bei mehr als 1000 Werten wird der Speicher-
inhalt überschrieben.

Syntax:

getpc
Rückmeldung:
[Zählerstand] [Status]

	Wertebereich
[Zählerstand]	0-65000
[Status]	0,1 0 = Aus, 1 = Ein

Beispiel:

getpc
Rückmeldung:
1204 1
Die Funktion "position capture" ist eingeschaltet, es wurden 1204 Triggersignale empfangen.
Die ersten 204 Werte wurden dabei überschrieben.

waitintrig (wit)

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Mit dem Kommando ***wit*** (wait_in_trigger) wird die Kommandoausführung so lange unterbrochen, bis ein digitales Eingangsignal anliegt und die Freigabe erzeugt.

Hinweise zum ***wit*** Kommando:

- Die ***wit*** Einstellung wird nach erfolgter Freigabe wieder zurückgesetzt.
- Mit Ctrl-C wird die ***wit*** Funktion abgebrochen.
- Das aktuell ausgeführte Kommando wird mit ***wit*** nicht unterbrochen.
- Die manuelle Positionierung mit Joystick oder Handrad wird von ***wit*** nicht beeinflusst.

Ab Firmwareversion 4.5.0 kann diese Funktion mit einem timeout versehen werden, der die Freigabe des Interpreters nach Ablauf einer vorgegebenen Zeit erzwingt.
Siehe Kommando ***setintrigtimeout (sitto)***

Für diese Funktion muss die Steuerung mit der Option "Digital Input/Output" ausgestattet sein.

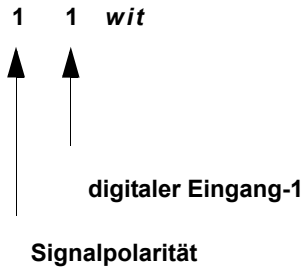
Syntax:

[pol_in] [input] ***wit***

	Beschreibung
[pol_in]	Signalpegel um den Interpreter freizugeben
[input]	digitaler Eingang

	Wertebereich	**Funktion**
[pol_in]	0, 1	0 = Low-Pegel 1 = High-Pegel
[input]	1, 2, 3	

Beispiel:



1 1 wit *[SP]* ***st***
High Pegel am digitalen Eingang-1 sperrt den Kommando-interpreter, deshalb wird das Kommando ***st*** nicht ausgeführt. Wechselt der Eingang auf Low-Pegel, wird das Kommando ***st*** ausgeführt und die ***wit*** Einstellung wieder zurück-genommen.

getpcdata (gpd)

Corvus TT, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***getpcdata*** liest die Daten aus dem "capture memory".

Erläuterung zur Funktion "position capture":
Mit jedem Triggerimpuls wird immer ein Datensatz abgespeichert. Jeder Datensatz enthält die Position aller aktiven Achsen, sowie die Systemzeit in Ticks. Ein Datensatz verbraucht damit im "capture memory" einen Speicherplatz.
Die abgespeicherten Datensätze werden mit dem Zählerstand des Triggerzählers indiziert und sind über diesen Index abrufbar, so lange sie nicht überschrieben wurden.
Insgesamt können 65000 Werte indiziert werden.

Im "capture memory" können 1000 Datensätze gespeichert werden. Das kleinste Zeitraster der Ticks beträgt 250 µs. Die Triggerfrequenz beträgt maximal 2 kHz.
Der Triggereingang wird mit einer Frequenz von 4 kHz abgefragt.

Syntax:

[Datensatzindex A] [Datensatzindex B] ***getpcdata***

	Wertebereich
[Datensatzindex A]	1-65000
[Datensatzindex B]	1-65000

Wobei A<B.

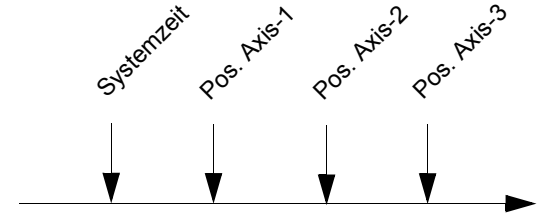
Rückgabe:
[Tick] [Pos. Achse-1] [Pos. Achse-2] [Pos. Achse-3]

	Wertebereich
[Tick]	1 Tick = 250 µs Die maximale Laufzeit der internen Uhr beträgt 298 Stunden.
[Pos. Achse-1] [Pos. Achse-2] [Pos. Achse-3]	Das Format ist abhängig von den Einstellungen mit *setunit*, *setdim* und *setnselpos*

Beispiel:

100 130 gpd

Es wird der Datensatz mit dem Index 100 bis 130 ausgelesen.




```
296149 9.749368 9.749368 0.000000
296151 9.749868 9.749868 0.000000
296153 9.750368 9.750368 0.000000
296155 9.750868 9.750868 0.000000
296157 9.751368 9.751368 0.000000
296159 9.751868 9.751868 0.000000
296161 9.752368 9.752368 0.000000
296163 9.752868 9.752868 0.000000
296165 9.753368 9.753368 0.000000
296167 9.753868 9.753868 0.000000
```

Einstellungen bei diesem Beispiel:
Geschwindigkeit = 1mm/s,
Triggerfrequenz = 2 kHz, Einheit = mm

296155 - 296153 = 2 Ticks = (2*250 µs) = 500 µs = 2 kHz

9.750868 mm - 9750368 mm = 0,5 µm

Bei einer Verfahrgeschwindigkeit der Achsen von 1mm/s und einer Triggerfrequenz von 2 kHz werden die Positionen in einem Raster von 0,5 µm abgespeichert.

clearpcdata (cpd)

Corvus TT, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***clearpcdata (cpd)*** löscht das "capture memory" sowie den Triggerzähler.

Syntax:

clearpcdata

oder ***cpd***

Beispiel:

cpd

setintrigtimeout

Corvus TT, Corvus eco, Corvus PCI

Verfügbar ab Firmwareversion 4.5.0.

Beschreibung:

Mit dem Kommando ***setintrigtimeout*** (***sitto)*** wird eine Wartezeit (timeout) für den bei Kommando *waitintrig* erwarteten Eingangstrigger festgelegt.
Erscheint das Triggersignal nicht innerhalb dieser Zeit, so wird nicht weiter gewartet, sondern die Befehlsabarbeitung fortgeführt.

Wird die Wartezeit auf 0 eingestellt, ist die Funktion abgeschaltet und eine unendliche lange Wartezeit ist aktiv. (Werkseinstellung)

Syntax:

[time] setintrigtimeout

	Beschreibung
[time] *	Wartezeit [s]
time = 0	Wartezeit unendlich

	Wertebereich
[time]	0.01 bis 100s

* Werkseinstellung = 0

Beispiel:

10 sitto

Es wird eine Wartezeit von 10s wird eingestellt.

getintrigtimeout

Corvus T, Corvus eco, Corvus PCI

Verfügbar ab Firmwareversion 4.5.0.

Beschreibung:

Das Kommando ***getintrigtimeout (gitto)*** liefert die eingestellte Wartezeit für das Triggereingangssignal zurück.

Syntax:

gitto

Rückmeldung:

[time]

	Wertebereich
[time]	0 0.01 bis 100s

Beispiel:

gitto

Joystick / Handrad

setjoysticktype

Corvus TT, Corvus eco, Corvus PCI

Beschreibung:

Mit dem Kommando ***setjoysticktype*** wird die Steuerung an das manuelle Bediengerät (Joystick oder Handrad) angepasst.

Syntax:

[Index*] ***setjoysticktype***

*Werkseinstellung = 3

	Wertebereich
[Index]	2, 3, 8

	Beschreibung
2	analoger Joystick
3	analoger Joystick
8	digitales Handrad

Partnerbefehl:

getjoysticktype

Beispiel:

8 setjoysticktype

getjoysticktype

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

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

Syntax:

getjoysticktype

Rückmeldung:

[Index]

	Beschreibung
2	analoger Joystick
3	analoger Joystick
8	digitales Handrad

Beispiel:

getjoysticktype

joystick (j)

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Der Befehl ***joystick*** aktiviert oder deaktiviert den manuellen Betrieb (Handrad od. Joystick) Bei eingeschaltetem manuellen Betrieb wird Statusbit D1 gesetzt.
Bei Corvus TT wird der Status zusätzlich mit einer LED angezeigt.

Automatischer Abgleich der Nullstellung des Joysticks
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***

	Wertebereich
[Index]	0, 1

Funktion

Index	Funktion
0	Joystick oder Handrad ausgeschaltet
1	Joystick oder Handrad eingeschaltet

Partnerbefehl:

getjoystick (ab Version 4.50),

Beispiel:

1 j
Der manuelle Betrieb wird eingeschaltet.

getjoystick (gj)

Corvus TT | Corvus eco | Corvus PCI

Verfügbar ab Firmwareversion 4.5.0

Beschreibung:

Der Befehl ***getjoystick*** liefert die Information ob der manuelle Betrieb aktiv ist.

Dieser Betriebszustand kann auch mit dem Kommando ***status*** festgestellt werden.

Syntax:

getjoystick

Rückmeldung:

[Wert]

Wert	Funktion
0	Handrad/Joystickbetrieb ausgeschaltet
1	Handrad/Joystickbetrieb eingeschaltet

Beispiel:

getjoystick

setjoyspeed (js)

Corvus TT, Corvus eco, Corvus PCI

Beschreibung:

Das Kommando ***setjoyspeed*** definiert die Geschwindigkeit mit der alle Achsen bei maximaler Auslenkung des Joysticks positioniert werden.
Die Motordrehzahl der einzelnen Achsen ergibt sich aus dem Verhältnis der eingestellten manuellen Geschwindigkeit zur eingetragenen Spindelsteigung

Eine achsspezifische Einstellung der Geschwindigkeit ist mit dem Kommando *setnjoyspeed* möglich.

Syntax:

[Geschwindigkeit] ***setjoyspeed***

	Einheit
[Geschwindigkeit]	Unit

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

Partnerbefehl:

getjoyspeed, setnjoyspeed

Beispiel:

20 setjoyspeed
unit = mm
Die Joystickgeschwindigkeit wird global auf 20 mm/s eingestellt.

getjoyspeed

Corvus TT, Corvus eco, Corvus PCI

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

setnjoyspeed (njs)

Corvus TT, Corvus eco, Corvus PCI

Verfügbar ab Firmwareversion 3.73

Beschreibung:

Das Kommando ***setnjoyspeed*** ermöglicht es die Joystickgeschwindigkeit der Achsen individuell festzulegen. Die Einheit dieser Geschwindigkeit wird von der eingestellten Einheit der 0-Achse bestimmt, siehe Kommando ***setunit***
Die Einstellungen von ***setnjoyspeed*** werden überschrieben wenn danach das Kommando ***setjoyspeed*** ausgeführt wird.

Eine kleine Spindeleinstellung kann dazu führen, dass der Motor eine sehr hohe Drehzahl erzeugen muss um die vorgegebene Geschwindigkeit zu erreichen. Hierbei besteht die Gefahr, dass der Motor stehen bleibt.
In solchen Fällen ist es notwendig die maximale Joystickgeschwindigkeit zu reduzieren.

Syntax:

[Geschwindigkeit] [Achse] ***setnjoyspeed***

	Unit
[Geschwindigkeit]	Die Einheit wird von der 0-Achse bestimmt.

[axis]	1, 2, 3

Beispiel:

20 1 setnjoyspeed

Die maximale Geschwindigkeit mit der Achse-1 mit dem Joystick bewegt werden kann beträgt 20 mm/s

getnjoyspeed (njs)

Corvus TT, Corvus eco, Corvus PCI

Verfügbar ab Firmwareversion 3.73

Beschreibung:

Das Kommando ***getnjoyspeed*** liest die Einstellung der achsspezifischen Joystickgeschwindigkeit zurück.

Syntax:

[Achse] ***getnjoyspeed***

Rückmeldung:

[Geschwindigkeit]

	Unit
[Geschwindigkeit]	Einheit der 0-Achse

Beispiel:

1 getnjoyspeed

Rückmeldung:

20.000000

setjoybspeed

Corvus TT, Corvus eco, Corvus PCI

Beschreibung:

Mit dem Kommando ***setjoybspeed*** kann eine zusätzliche Joystickgeschwindigkeit festgelegt werden. Diese wird mit einem Schalter oder Taster am Joystick aktiviert.

Syntax:

[Geschwindigkeit] ***setjoybspeed***

	Einheit
[Geschwindigkeit]	Unit

Geschwindigkeit	**Wertebereich**
minimal	15.62nm/s
maximal	60 Umdrehungen/s x Spindelsteigung

Partnerbefehl:

getjoybspeed

Beispiel:

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

getjoybspeed

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

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

Syntax:

getjoybspeed

Rückmeldung:

[Wert]

	Einheit
[Wert]	unit

Beispiel:

getjoybspeed

Rückmeldung:

0.010000

setjoyassign

Corvus TT, Corvus eco, Corvus PCI

Verfügbar ab Firmware Version 4.40

Beschreibung:

Mit dem Kommando ***setjoyassign*** kann die Wirkung der Joystickauslenkung auf die Motordrehrichtung eingestellt werden. Zusätzlich ist es möglich die Zuordnung der Joytickachsen zu den Motorachsen zu verändern.

Syntax:

[Zuweisung] [Motorachse] ***setjoyassign***

	Wertebereich
[Zuweisung]	ganzzahlig -3 bis +3
[Motorachse]	1, 2, 3

[Zuweisung]	**Richtung**	**Joystickachse**
0	Joystick aus	-
1	positiv	X-Achse
2	positiv	Y-Achse
3	positiv	Z-Achse

[Zuweisung]	**Richtung**	**Joystickachse**
0	Joystick aus	-
-1	negative	X-Axis
-2	negative	Y-Axis
-3	negative	Z-Axis

Werkseitige Einstellung:

1 1 setjoyassign
2 2 setjoyassign
3 3 setjoyassign

Beispiele:

2 1 setjoyassign
1 2 setjoyassign
-3 3 setjoyassign

Motorachse-1 wird durch Joystickachse Y und Motorachse-2 durch Joystickachse X bewegt.
Motorachse-3 durch Joystickachse Z, jedoch mit inverser Wirkrichtung.

3 1 setjoyassign
3 2 setjoyassign
0 3 setjoyassign

Motorachse-1 und Motorachse-2 werden simultan mit Joystickachse Z bewegt.
Motorachse-3 ist für den Joystick abgeschaltet.
Die Joystickachsen X und Y haben keine Wirkung.

getjoyassign

Corvus TT, Corvus eco, Corvus PCI

Verfügbar ab Firmware Version 4.40

Beschreibung:

Das Kommando ***getjoyassign*** liest die Zuordnung der Joystickachsen zu den Motorachsen bzw. deren Wirkung auf die Motordrehrichtung zurück.

Syntax

[Motorachse] ***getjoyassign***

	Wertebereich
[Motorachse]	1, 2, 3

Rückmeldung:

[Joystickachse]

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

Beispiel:

1 getjoyassign

Rückgabe:
3

setjoydiag

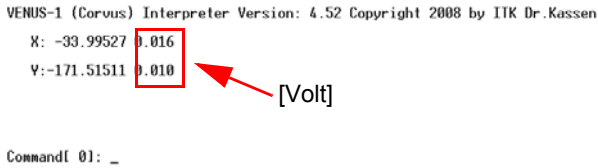
Corvus TT, Corvus eco, Corvus PCI

Verfügbar ab Firmwareversion 4.40

Beschreibung:

Das Kommando ***setjoydiag*** aktiviert die Joystickdiagnose, mit der die analogen Ausgangspegel des Joysticks im Terminal angezeigt werden.

Für diese Funktion muss der Terminal Betrieb (*1 mode*) eingeschaltet werden.



Syntax:

[Schalter] ***setjoydiag***

	Wertebereich
[Schalter]	0, 1

	Funktion
0	Joystickdiagnose aus
1	Joystickdiagnose ein

Beispiel:

1 mode
1 setjoydiag
Der Terminal Modus und die Joystickdiagnose werden eingeschaltet

getjoydiag

Corvus T, Corvus eco, Corvus PCI

Beschreibung:

Mit dem Kommando ***getjoydiag*** wird überprüft ob die Joystickdiagnose eingeschaltet ist.

Syntax

getjoydiag

Rückmeldung:

[Schalter]

	Wertebereich
[Schalter]	0, 1

Beispiel:

getjoydiag

Rückgabe:
0

setwheel

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

gespeichert nach "reset"

Kommando ***getwheel*** initialisiert den Handrad Modus.
Die Einstellung muss mit ***save*** gespeichert werden, danach ist ein ***reset*** auszuführen.
Das Handrad wird mit dem Kommando *setjoystick* ein- und ausgeschaltet.

Ein gleichzeitiger Betrieb von Joystick und Handrad ist nicht möglich. Die Umschaltung der beiden Betriebsmodi wird mit dem Kommando *setjoysticktype* durchgeführt.

Option 

Für die Handradfunktion werden alle drei Encodereingänge der Steuerung verwendet.
Der Closed Loop Betrieb ist dadurch nur mit dem analogen Messinterface (sin/cos Modul) möglich.

Syntax:

[Index] ***setwheel***

	Wertebereich
[Index]	0, 1

[Index]	**Beschreibung**
0	Encoderbetrieb (default)
1	Handradbetrieb

Beispiel:

1 setwheel [cr] ***save*** [cr] ***reset*** [cr]

Es wird der Handradmodus aktiviert.

getwheel

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getwheel*** überprüft ob der Handrad- oder Encoderbetrieb eingeschaltet sind.

Syntax

getwheel

Rückmeldung:

[Index]

	Wertebereich
[Index]	0, 1

[Index]	**Beschreibung**
0	Encoderbetrieb (default)
1	Handradbetrieb

Beispiel:

getwheel

setwheelres

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit Kommando ***getwheelres*** wird die Steuerung an die Anzahl der vom Handrad erzeugten Impulse bzw. der mechanischen Rastung pro Handradumdrehung (360°) angepasst.

Werkseinstellung: 100 Impulse pro Umdrehung.

Syntax:

[Impulse] [Achse] ***setwheelres***

	Wertebereich
[Impulse]	1-65535
[Achse]	1, 2, 3

Beispiel:

200 1 setwheelres

getwheelres

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getwheelres*** liefert die Anzahl der Impulse die das Handrad mit einer Umdrehung erzeugt.

Syntax

[Achse] ***getwheelres***

Rückmeldung:

[Impulse]

	Wertebereich
[Impulse]	1 65535

Beispiel:

1 getwheelres

setwheelratio

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit Kommando ***setwheelratio*** wird die Positioniergeschwindigkeit bzw. die Positionierauflösung des Handradbetriebs eingestellt.
Eingegeben wird der gewünschte Weg bei einer Handradumdrehung. Daraus ergibt sich die Auflösung wie folgt

setwheelres = 100 Pulse/Umdrehung
setwheelratio = 1mm
Ergebnis:

Jede Rastung des Handrades erzeugt eine Strecke von
1mm / 100 Ticks = 0.01 mm

Mit dem Speed Schalter am Handrad kann die Auflösung auf einen zweiten, vorher festgelegten Wert, umgeschaltet werden. Die Parametrisierung dieser Einstellung wird mit dem Kommando *setwheelbratio* vorgenommen.

Syntax:

[Richtung] [Strecke] [Achse] ***setwheelratio***

	Wertebereich	**Einheit**
[Drehrichtung]	Positionierrichtung	+/-
[Strecke]	-32767...+32767mm	unit
[Achse]	1, 2, 3	

Beispiel:

-10 1 setwheelratio

getwheelratio

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getwheelratio*** liefert die Strecke die das Handrad mit einer Umdrehung erzeugt.

Syntax

[Achse] ***getwheelratio***

Rückmeldung:

[Richtung] [Strecke]

	Wertebereich	**Einheit**
[Strecke]	-32767mm...32767mm	unit

Beispiel:

1 getwheelratio

setwheelbratio

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit Kommando ***setwheelratio*** wird eine zweite Positioniergeschwindigkeit bzw. Positionierauflösung für den Handradbetrieb eingestellt.
Eingegeben wird auch hier der gewünschte Weg bei einer Handradumdrehung. Daraus ergibt sich die Auflösung wie folgt

setwheelres = 100 Ticks
setwheelratio = 1mm
setwheelbratio = 0.1mm

Ergebnis dieser Einstellung:
Jeder Tick des Handrades erzeugt eine Strecke von
1mm / 100 Ticks = 0.01 mm
Mit dem Speed Schalter am Handrad kann die Auflösung auf 0.001mm pro Tick umgeschaltet werden.

Syntax:

[Richtung] [Strecke] [Achse] ***setwheelratio***

	Wertebereich	**Einheit**
[Drehrichtung]	Positionierrichtung	+/-
[Strecke]	-32767...+32767mm	unit
[Achse]	1, 2, 3	

Beispiel:

0.1 1 setwheelbratio

getwheelbratio

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Das Kommando ***getwheelbratio*** liefert die Einstellung der zweiten Geschwindigkeit bzw. Tick-Auflösung des Handrades.

Syntax

[Achse] ***getwheelbratio***

Rückmeldung:

[Richtung] [Strecke]

	Wertebereich	**Einheit**
[Drehrichtung]	Positionierrichtung	+/-
[Strecke]	-32767...+32767mm	unit

Beispiel:

1 getwheelbratio

Systemkommandos

save

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***save*** speichert alle speicherbaren Parameter in den nichtflüchtigen Speicher der Steuerung. Diese Einstellungen bleiben auch nach dem Abschalten der Steuerung erhalten und sind nach dem Einschalten aktiv.

save beendet eine programmierte Positionierung und unterbricht die manuelle Positionierung während der Speicherphase.

Nach Beendigung der Speicherphase erfolgt im Terminal Mode eine Rückmeldung mit **OK**.
Mit der Befehlsfolge ***save status*** kann auch im Host Mode eine Rückmeldung erzeugt werden.

Speicherbare Parameter sind in diesem Handbuch mit untenstehenden Symbol gekennzeichnet.

Syntax:

save

Rückmeldung:

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

Beispiel:

save

restore

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Es erfolgt keine Rückmeldung nach Beendigung der Wiederherstellung.

Syntax:

restore

Rückmeldung:

keine Rückmeldung

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

Verwandter Befehl:

getfpara

Beispiel:

restore

getfpara

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Die aktuellen Einstellungen werden dabei überschrieben.
Diese können mit *restore* wieder hergestellt werden.

Syntax:

getfpara

Rückmeldung:

keine

Beispiel:

getfpara

clear

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

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

Bei nicht korrekt eingehaltener Syntax 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

reset

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***reset*** bewirkt einen Neustart und ist mit dem Aus- und Einschalten der Steuerung vergleichbar.

Während des Reset wird die LED-Diagnoseanzeige kurz abgeschaltet.
Ein kurzer Signalton signalisiert nach dem Neustart die Betriebsbereitschaft der Steuerung.

Syntax:

reset

Beispiel:

reset

beep

Corvus TT	-	-

Beschreibung:

Das Kommando ***beep*** aktiviert den internen Signalgeber. Der Geber erzeugt damit einen 1kHz Ton dessen Dauer durch das *beep* Kommando festgelegt werden kann.

Maximale Dauer des Tons = 10s

Syntax:

[Dauer] beep

	Bereich	**Einheit**
[Dauer]	1-10.000	ms

Beispiel:

1000 beep

version

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

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

Syntax:

version

Rückmeldung

[Versionsnummer]

Beispiel:

version

Rückgabe:

4.55

getmacadr	Corvus TT	-	-

Beschreibung:

Der Befehl ***getmacadr*** liefert die Ethernet MAC Adresse der Steuerung.

Syntax:

getmacadr

Rückmeldung

[Mac-Adresse]

Beispiel:

getmacadr

Rückgabe im Terminal Mode:

```
Ethernet MAC address: 00:50:C2:10:91:91
```

Rückgabe im Host Mode:

```
00:50:C2:10:91:91
```

identify

Corvus TT	-	-

Beschreibung:

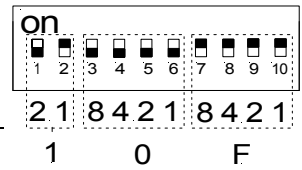
Der Befehl ***identify*** bei Corvus TT die Versionsnummer der Hard- und Firmware der Steuerung, sowie die Einstellung des auf der Rückseite der Steuerung befindlichen DIP-Schalters.

Syntax:

identify

Rückmeldung:

[Model] [HW-Rev] [SW-Rev] [Board-Sw] [DIP-Sw]

	Beschreibung
[Model]	Gerätebezeichnung
[HW-Rev]	Hardware Version
[SW-Rev]	Software Version
[Board-Sw]	Interner Schalter
[DIP-Sw]	Einstellung des Konfigurationsschalters. Die Schalterstellung ist in der Rückmeldung hexadezimal kodiert. 

Verwandter Befehl:

version

Beispiel:

identify

Rückmeldung: `Corvus 1 312 1 10F`

getoptions

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getoptions*** informiert über die freigeschalteten Optionen der Steuerung.
In der zurückgeliefeten dezimalen Zahl sind die Optionen kodiert.
Jeder Option wurde eine binärer Wert von D0 bis D9 zugeordnet.
Sind mehrere Optionen freigeschaltet, wird in der Rückmeldung der dezimale Wert dieser Stellen addiert.
Für die Auswertung ist es notwendig den dezimalen Wert in eine binäre Zahl zu wandeln und die entsprechenden Stellen auszumaskieren.

Syntax:

getoptions

Rückmeldung:

[Wert]

	Wertebereich
[Wert]	0 - 975

Bit	**Dezimal**	**freigeschaltete Option**
D0	1	3. Achse
D1	2	Ethernetschnittstelle TCP/IP
D2	4	Closed Loop / alle Achsen
D3	8	digitale Ein-Ausgänge, 3/3
D4	16	nicht genutzt
D5	32	nicht genutzt
D6	64	Closed Loop Achse-1
D7	128	Closed Loop Achse-2
D8	256	Closed Loop Achse-3
D9	512	Geschwindigkeit 60 U/s

Beispiel:

getoptions

Rückmeldung: 9

Achse-3 und die digitalen Ein-Ausgänge sind freigeschaltet.

Bit	Dezimal	freigeschaltete Option
D0	**1**	**3. Achse**
D1	2	Ethernetschnittstelle TCP/IP
D2	4	Closed Loop / alle Achsen
D3	**8**	**digitale Ein-Ausgänge, 3/3**
D4	16	nicht genutzt
D5	32	nicht genutzt
D6	64	Closed Loop Achse-1
D7	128	Closed Loop Achse-2
D8	256	Closed Loop Achse-3
D9	512	Geschwindigkeit 60 U/s
Summe	**9**	

getserialno

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Der Befehl ***getserialno*** liefert die Seriennummer der Steuerung zurück.

Syntax:

getserialno

Rückmeldung

JJ HW SERI

	Beschreibung	Stellen
JJ	Jahr	2
HW	Hardware Version	2
SERIAL	Forlaufende Nummer	4

Beispiel:

getserialno

Rückgabe:
01020105

Beschreibung:
Jahr: 2001
Hardware Version: 02
Fortlaufende Nummer: 0105

Fehlerkorrektur

setpcor

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setpcor*** aktiviert oder deaktiviert die Funktion "Positionsfehlerkorrektur".
Diese Funktion kann für jede Achse separat eingeschaltet werden.

Syntax:

[Funktion] [Achse] ***setpcor***

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

[Funktion]	**Beschreibung**
0	Fehlerkorrektur aus
1	Fehlerkorrektur ein

Partnerbefehl:

getpcor

Beispiel:

0 1 setpcor

Die Fehlerkorrektur der Achse-1 wird abgeschaltet.

getpcor

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***getpcor*** liefert den Status der Fehlerkorrektur.

.

Syntax:

[Achse] ***getpcor***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[0,1]

	Funktion
[0]	Fehlerkorrektur aus
[1]	Fehlerkorrektur ein

Beispiel:

1 getpcor

setpdat

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Das Kommando ***setpdat*** dient zur Eingabe der Daten für die Positionsfehlerkorrektur.

Warum Fehlerkorrektur?

Corvus wurde speziell dafür entwickelt Schrittmotore mit einer sehr hohen Positionsauflösung anzusteuern und laufruhig mit hoher Genauigkeit zu positionieren.
Die Genauigkeit der Positionierung ist dabei abhängig von der Bauart und der Fertigungsgenauigkeit des Schrittmotors, sowie von der angebauten Positioniermechanik.
Zu den Fehlern der Positioniermechanik gehören in erster Linie Fehler der Spindel un der Führungen. Dazu kommen noch belastungsabhängige Positionierfehler des Motors.
Durch den Anbau eines Längenmesssystems können wesentlichen Fehler erfasst und mit Hilfe einer Closed Loop Positionsregelung ausgeglichen werden.

Für viele Systeme ist die Ausstattung mit einem Messsystem aber zu teuer und aufwändig.
Als kostengünstige und wirksame Alternative bietet sich an Positionierfehler ohne Verwendung eines Messsystems im sogenannten offenen "Regelkreis" durch die Steuerung selbst ausgleichen zu lassen.
Dabei wird das Prinzip verfolgt, die einmalig gemessenen Fehlerwerte der Mechanik in die Steuerung zu speichern und die Stellpositionen entsprechend zu korrigieren.

Beschreibung der Corvus Fehlerkorrektur

Die Corvus Fehlerkorrektur ist Eindimensional, das heisst jede Achse muss separat korrigiert werden.
Die Fehlerkennlinie wird immer beginnend vom Startpunkt erfasst und mit dem Kommando ***setpdat*** in aufsteigender Reihenfolge an die Steuerung übertragen.
Das Stützstellenraster ist auf die Länge 1mm festgelegt.
Im Betrieb werden die Fehler an den Stützstellenpositionen zu 100% korrigiert. Zwischen den Punkten errechnet die Steuerung die Korrekturwerte eigenständig durch lineare Interpolation.

Mit dem Kommando ***save*** werden die Werte in der Steuerung gespeichert.

Die Korrektur arbeitet sowohl im programmierten als auch im manuellen Betrieb.
Maximal kann eine Strecke von 499mm korrigiert werden.

Syntax:

[$F_{0.}$.....F_{499}] [Start] [Anzahl] [Achse] ***setpdat***

	Beschreibung	Wertebereich	Einheit
[$F_{0.}$..F_{499}	Positionsfehler an den Stützstellen **(In aufsteigender Reihenfolge)**	+/- 100	µm
[Start]	Erste Messposition	0-499 (ganzzahlig) [Start] + [Anzahl] ≤ 499	mm
[Anzahl]	Anzahl der Korrektur-werte	1-500	
[Achse]	Achse	1, 2, 3	

Partnerbefehl:

getpdat

Beispiele:

0.5 0.1 0.5 1.2 -0.5 1.2 0 6 1 setpdat

0mm 1mm 2mm 3mm 4mm 5mm (Stützstellenpositionen)

Bei Achse-1 wird die Strecke 0-5mm korrigiert.
Es werden die Positionsfehler an den Stützstellen 0 bis 5 mm übertragen.

0.3 0.5 0.9 1.5 -0.5 6 5 2 setpdat

6mm 7mm 8mm 9mm 10mm

Für Achse-2 wird die Strecke 6-10mm korrigiert.
Es werden 5 Positionsfehlerwerte übertragen.

getpdat

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***getpdat*** werden die eingetragenen Korrekturwerte sequentiell ausgelesen.
Es werden immer zehn aufeinanderfolgende Werte ausgegeben.

.

Syntax:

[Startwert] [Achse] ***getpdat***

	Wertebereich	**Einheit**
[Startwert]	0-499	mm
[Achse]	1, 2, 3	

Rückmeldung:

[F_{Start} ... $F_{Start+9}$]

	Wertebereich	**Einheit**
F_{Start} ... $F_{Start+9}$	+/-100	µm

Beispiel:

12 1 getpdat

Rückmeldung:

```
0.992 1.999 2.991 3.998 4.990 0.000 0.000 0.000 0.000 0.900
```

12mm 13mm 14mm 15mm 16mm 17mm 18mm 19mm 20mm 21mm

Es werden 10 Fehlerwerte ab der Stützstelle 12mm zurückgeliefert.

setblc

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 3.66

Beschreibung:

Das Kommando ***setblc*** aktiviert oder deaktiviert die Funktion "backlash-compensation".
Mit dieser Funktion können mechanische Umkehrfehler, die beim Wechsel der Drehrichtung entstehen ausgeglichen werden.
Die Distanz der Kompensation wird mit dem Kommando *setblcd* eingestellt.

Syntax:

[Funktion] [Achse] ***setblc***

[Funktion]	Beschreibung
0	Kompensation aus
1	Kompensation ein

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

Partnerbefehl:

getblc, setblcd

Beispiel:

1 1 setblc

getblc

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 3.66

Beschreibung:

Das Kommando ***getblc*** liefert den Status der Funktion "backlash-compensation".

.

Syntax:

[Achse] ***getblc***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[0,1]

	Funktion
[0]	Kompensation aus
[1]	Kompensation ein

Beispiel:

1 getblc

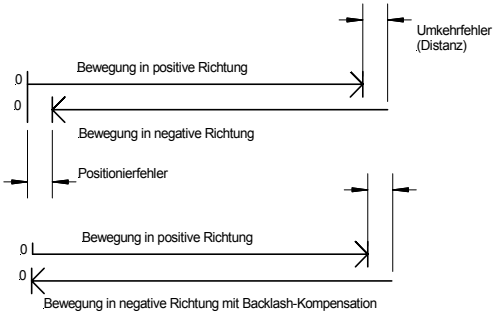
setblcd

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 3.66

Beschreibung:

Mit dem Kommando ***setblcd*** wird der Korrekturwert (Distanz) der Funktion "backlash compensation" eingestellt. Die Korrektur wird immer zu Beginn einer negativen Bewegung sprunghaft angewendet.



Syntax:

[Fehler] [Achse] ***setblcd***

	Beschreibung
[Fehler]	Distanz zur Kompensation des Umkehrfehlers

	Wertebereich
[Fehler]	0 - 0.1mm
[Achse]	1, 2, 3

Kleinster Wert: 0.000001mm

Beispiel:

0.001 1 setblcd

getblcd

Corvus TT	Corvus eco	Corvus PCI

Verfügbar ab Firmwareversion 3.66

Beschreibung:

Das Kommando ***getblcd*** liefert die Einstellung der backlash Kompensation zurück.

.

Syntax:

[Achse] ***getblcd***

	Wertebereich
[Achse]	1, 2, 3

Rückmeldung:

[0.000000 - 0.1mm]

Beispiel:

1 getblcd

Corvus Makros

Corvus Makro FAQ

Was ist ein Corvus Makro:

Ein Corvus Makro besteht im Prinzip aus mehreren Venus-1 Kommandos, die als Kommandoliste in der Steuerung abgespeichert und dort ausgeführt werden können.

Beispiel eines Corvus Makro:

```
beginmakro
cal
0 setout
20 sv
1 0 setunit
1 1 setunit
1 2 setunit
2 3 setunit
10000 sa
200 0 1 ot
10000 sv
1 setpc
clearpcdata
1 1 setnselpos
1 2 setnselpos
3 setotmode
1000 1000 m
100.1234 1 1 200 1 2 wpot
10 10 gpd
getpc
endmakro
```

beginmakro [SP]
[Venus-1 command] [SP]
[Venus-1 command] [SP]
[Venus-1 command] [SP]
endmakro [CR LF]

Beispiel:

beginmakro
0.01 1 1 0 2 1 setrptdata
0 0.1 startrpt
0.1 0 0 m
0 0 0m
endmakro

Welche Vorteile haben Makros

Mit Makros werden Venus-1 Kommandos direkt auf der Steuerung ausgeführt. Aufgrund des deutlich reduzierten Kommunikationsaufwandes wird die Host CPU entlastet und für das Anwenderprogramm bleibt mehr Zeit andere Aufgaben zu übernehmen.

Besonders Hilfreich ist die Makro Funktion bei schnellen Scanning Applikationen, die keine vom Host Betriebssystem verursachten "time lag´s" tolerieren.
Explizit die verschiedenen echtzeitnahen Corvus Triggerfunktionen werden mit Makros optimal nutzbar.

Wie wird ein Makro erzeugt und ausgeführt

Das Corvus Makro wird als einfacher Textfile erzeugt und über die RS-232 oder Ethernetschnittstelle in den Makro-Exe Puffer der Steuerung übertragen.
Mit einem Startkommando kann das Makro beliebig oft ausgeführt werden. Die Übertragung in den Makro-Exe Puffer erfolgt aufgrund der Steuerwörter *beginmakro* und *endmakro* automatisch.
Während der Ausführung eines Makros sind keine weiteren Kommandos, außer Abbruckkommandos, möglich.

Wieviele Kommandos können in einem Makro gespeichert werden

Die Größe eines Makros wird nicht in der Anzahl der darin enthaltenen Kommandos angegeben, sondern in der Anzahl der enthaltenen Symbole.
Eine Kommandozeile kann aus einem oder mehreren Symbolen bestehen.

Maximal können 4000 Symbole in den Makro-Exe Puffer übertragen werden.

Beispiele:

100$_1$ *100*$_2$ *move*$_3$
Diese Kommandozeile besteht aus drei Symbolen

100$_1$ *100*$_2$ *10*$_3$ *move*$_4$
Diese Kommandozeile besteht aus vier Symbolen

st$_1$
Diese Kommandozeile besteht aus einem Symbol

Das folgende Makro verbraucht insgesamt 13 Symbole

beginmakro

2 setdim 2 Symbole
cal 1 Symbol
rm 1 Symbol
1 setout 2 Symbole
1000 beep 2 Symbole
0 0 move 3 Symbole
2000 beep 2 Symbole

endmakro

Ist es möglich das Makro in einer Schleife arbeiten zu lassen

Der Makrobefehl ***startmakro*** kann innerhalb des Makros aufgerufen werden. Damit ist es möglich das Makro in einer Endlosschleife abzuarbeiten.

```
beginmakro
cal
0 setout
20 sv
100 sa
200 0 1 ot
50 50 0 m
10 1 1 200 1 2 wpot pos
20 1 1 200 1 2 wpot pos
30 1 1 200 1 2 wpot pos
40 1 1 200 1 2 wpot pos
50 1 1 200 1 2 wpot pos
ge
startmakro
endmakro
```

Kann ein Makro automatisch ausgeführt werden

Nein.

Mit ***setpowerup*** können Kommandos automatisch nach dem Einschalten der Steuerung ausgeführt werden.

Übersicht der Makro Kommandos

Für die Steuerung und Verwaltung der Makros sind folgende Kommandos zuständig.

beginmakro
Kennzeichnet den Anfang des Makros.

endmakro
Kennzeichnet das Ende des Makros.

startmakro
Führt das Makro im Makro-Exe Puffer aus.

listmakro
Liefert die Anzahl der Symbole im Makro-Exe Puffer.

Ctrl-D
Unterbricht die Ausführung des Makros o. Download.

Makro syntax

beginmakro [SP]
[Venus-1 command] [SP]
[Venus-1 command] [SP]
[Venus-1 command] [SP]
endmakro [CR LF]

[SP] = Space
[CR LF] = carriage return, line feed

Beispiel:

beginmakro
0.01 1 1 0 2 1 setrptdata
0 0.1 startrpt
0.1 0 0 m
0 0 0m
endmakro

Makrobefehle

beginmakro / endmakro

Corvus TT, Corvus eco, Corvus PCI

Beschreibung:

Die Kommandos ***beginmakro*** und ***endmakro*** sind Steuerwörter, die den Beginn und das Ende eines Makros kennzeichnen.

Syntax:

beginmakro / endmakro

Beispiele:

```
beginmakro
cal
100 sa
200 0 1 ot gt
2 sv
100 0 0 m
10 1 1 200 0 2 wpot gt
20 1 1 200 0 2 wpot gt
30 1 1 200 0 2 wpot gt
40 1 1 200 0 2 wpot gt
50 1 1 200 0 2 wpot gt
60 1 1 200 0 2 wpot gt
70 1 1 200 0 2 wpot gt
80 1 1 200 0 2 wpot gt
ge
0 0 0 m
endmakro
```

```
beginmakro
cal
rm
0 0 0 m
endmakro
```

startmakro

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem Kommando ***startmakro*** wird das im Makro-Exe Puffer befindliche Makro ausgeführt.
Das Kommando kann auch innerhalb des Makros selbst aufgerufen werden.

Syntax:

startmakro

Beispiel:

startmakro

listmakro

Corvus TT	Corvus eco	Corvus PCI

Beschreibung:

Mit dem Kommando ***listmakro*** wird die Anzahl der im Makro-Exe Puffer befindlichen Symbole ausgelesen. In den Puffer können maximal 4000 Symbole übertragen werden. Bei Überschreiten der maximalen Anzahl von Symbolen wird Fehlermeldung 1201 erzeugt.

Syntax:

listmakro

Beispiel:

listmakro

Rückmeldung:
1204

Ctrl-D

Corvus TT | Corvus eco | Corvus PCI

Beschreibung:

Mit dem ASCII Zeichen ***Ctrl-D*** wird das momentan ausgeführte Makro oder der Download des Makro abgebrochen.

Syntax:

Ctrl-D

ASCII Zeichen	Dezimalwert	Hex Wert
Ctrl-D	4	0x4

Beispiel:

Ctrl-D

Kurzbeschreibung der Venus-1 Kommandos

Grundeinstellungen

setpitch **29**
Mit dem Kommando ***setpitch*** wird die Steuerung an das Verhältnis von Motorumdrehung und resultierender linearer Bewegung angepaßt.
Beispiel: ***4.009 1 setpitch***

getpitch **31**
Das Kommando ***getpitch*** liefert die mit ***setpitch*** vorgenommene Einstellung zurück.
Beispiel: ***2 getpitch -1 getpitch***

setdim. **33**
Mit dem Kommando ***setdim*** wird die Achsdimension eingestellt.
Damit ist auch festgelegt, wieviele Achsparameter bei dimensionsabhängigen Kommandos von der Steuerung erwartet bzw. zurückgeliefert werden.
Beispiel: ***2 setdim***

getdim **34**
Das Kommando ***getdim*** liest die Einstellung der Dimension zurück.
Beispiel: ***getdim***

setunit **35**
Das Kommando ***setunit*** definiert die physikalische Einheit der Ein- und Ausgabewerte für Geschwindigkeit, Beschleunigung und Position.
Beispiel: ***2 0 setunit***

getunit **38**
Das Kommando ***getunit*** liefert die mit ***setunit*** vorgenommene Einstellungen zurück.
Beispiel: ***1 getunit -1 getunit***

setumotmin **39**
Mit dem Kommando ***setumotmin*** wird die Motorspannung für den Stillstand und unteren Drehzahlbereich eingestellt.
Beispiel: ***2000 1 setumotmin***

getumotmin **40**
Das Kommando ***getumotmin*** liest die durch ***setumotmin*** vorgenommene Einstellung zurück.
Beispiel: ***1 getumotmin -1 getumotmin***

setumotgrad **41**
Mit dem Kommando ***setumotgrad wird die Motorspannung bzw.*** der Phasenstrom für den mittleren und oberen Drehzahlbereich eingestellt.
Die Einstellung beeinflusst damit das Drehmoment bei mittlerer Drehzahl.
Beispiel: ***70 1 setumotgrad***

getumotgrad **42**
Der Befehl ***getumotgrad*** liest den durch ***setumotgrad*** eingestellten Wert

zurück.
Beispiel: ***1 getumotgrad -1 getumotgrad***

setpolepairs 43
Mit dem Kommando ***setpolepairs*** wird die Anpassung an die Polpaarzahl des Schrittmotors vorgenommen.
Beispiel: ***50 1 setpolepairs***

getpolepairs 44
Das Kommando ***getpolepairs*** liest die eingestellte Polpaarzahl zurück.
Beispiel: ***1 getpolepairs -1 getpolepairs***

setaxis 45
Das Kommando ***setaxis*** aktiviert oder deaktiviert die Achse für die Positionierung und die Endschalterfahrt. Zusätzlich wird die Wirkung der Kommandos cal, rm und setpos auf die Positionsanzeige und Limits beinflusst.
Beispiel: ***1 3 setaxis***

getaxis 47
Das Kommando ***getaxis*** liefert die mit ***setaxis*** vor-genommene Einstellungen zurück.
Beispiel: ***2 getaxis -1 getaxis***

setpowerup 49
Mit dem Kommando ***setpowerup*** können Power-Up Befehle, automatisch nach dem Einschalten der Steuerung ausgeführt werden.
Beispiel: ***1 setpowerup***

getpowerup 51
Das Kommando ***getpowerup*** liest die Power-Up Einstellung der Steuerung zurück.
Beispiel: ***getpowerup***

setphaseares 53
Mit dem Kommando ***setphaseares*** kann die Winkel-schrittauflösung der Motorendstufen stufenweise reduziert werden.
Beispiel: ***2 1 setphaseres***

getphaseares 55
Das Kommando ***getphaseares*** liefert die eingestellte Stufe der Schrittwinkelauflösung zurück.
Beispiel: ***2 getphaseares***

setmotiondir 57
Mit Kommando ***setmotiondir*** wird die Motordrehrichtung festgelegt.
Beispiel: 1 1 ***setmotiondir***

getmotiondir 58
Das Kommando ***getmotiondir*** zeigt an ob die Motordrehrichtung von der Standardeinstellung abweicht.
Beispiel: ***1 getmotiondir***

Kommunikation

Geschwindigkeit und Beschleunigung

Das Kommando ***getmanaccel*** liest die eingestellte Beschleunigung für den manuellen Betrieb.
Beispiel: ***getmanaccel***

setcalvel 77
Mit dem Kommando ***setcalvel*** werden zwei Geschwindigkeiten festgelegt, mit denen die Steuerung die cal-Endschalterfahrt ausführt.
2. Geschwindigkeit aus dem Endschalter heraus.
Beispiel: ***2 1 setcalvel***

getcalvel. 79
Das Kommando ***getcalvel*** liest die mit ***setcalvel*** eingestellte Geschwindigkeit der Endschalterfahrt in den cal-Endschalter zurück.
Beispiel: ***getcalvel***

setncalvel. 81
Mit dem Kommando ***setncalvel*** werden die Geschwindigkeiten festgelegt, mit denen die Steuerung die ncal Endschalterfahrt ausführt.
Beispiel: ***2 1 1 setncalvel***

getncalvel. 83
Das Kommando ***getncalvel*** liest die mit ***setncalvel*** eingestellte Geschwindigkeit der cal Endschalterfahrt zurück.
Beispiel: ***1 getncalvel***

setrmvel 85
Das Kommando ***setrmvel*** definiert zwei Geschwindigkeiten, mit denen die Steuerung die rm-Endschalterfahrt ausführt.
Beispiel: ***2 1 setrmvel***

getrmvel 87
Das Kommando ***getrmvel*** liest die mit ***setrmvel*** eingestellten Geschwindigkeiten für die Endschalterfahrt zurück.
Beispiel: ***getrmvel***

setnrmvel 89
Das Kommando ***setnrmvel*** definiert zwei Geschwindigkeiten, mit denen die Steuerung die nrm-Endschalterfahrt ausführt.
Beispiel: ***2 1 2 setnrmvel***

getnrmvel 91
Das Kommando ***getnrmvel*** liest die mit ***setnrmvel*** eingestellte Geschwindigkeit für die rm Endschalterfahrt zurück.
Beispiel: ***2 getnrmvel***

setrefvel 93
Der Befehl ***setrefvel*** legt die Geschwindigkeit fest, mit der die Positionierung zur Referenzmarke ausgeführt wird.
Beispiel: ***0.5 1 setrefvel***

getrefvel 94

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

Positionierkommandos

Das Kommando ***move*** positioniert die Achsen zu absoluten Koordinaten.
Beispiel: ***12.5 20 0.0001 m***
Beispiel: ***12.5 20 m***

Der Befehl ***rmove*** positioniert die Achsen relativ zu den aktuellen Koordinaten.
Beispiel: ***12.5 20 0.0001 r***
Beispiel: ***12.5 20 r***

Mit dem Kommando ***speed*** wird die Achse im sogenannten speed mode bewegt. Damit erfolgt die Positionierung durch Angabe einer Geschwindigkeit und Bewegungsrichtung.
Beispiel: ***10 1 speed***
Beispiel: ***-0.1 2 speed***

Mit dem Kommando ***stopspeed*** wird der speed mode
für alle Achsen mit der eingestellten Systembeschleunigung abgebrochen.
Beispiel: ***stopspeed***

Der Befehl ***test*** aktiviert eine Testroutine, mit der die ausgewählte Achse schrittweise zu den Limits positioniert wird. Die Schrittgrösse ist frei wählbar.
Beispiel: ***cal***
Beispiel: ***10 1 test***

Der Befehl ***randmove*** erzeugt für alle aktiven Achsen zufällige Positionsdaten innerhalb des gültigen Verfahrbereichs.
Beispiel: ***randmove***

Endschalterfunktionen

Das Kommando ***cal*** löst die Endschalterfahrt aller aktiven Achsen zum cal-Endschalter aus. Hierbei werden die aktiven Achsen gleichzeitig in negative Richtung positioniert. bis der cal-Endschalter betätigt ist. Die Steuerung positioniert danach wieder in Richtung positiver Positionswerte bis vor den Endschalter.
Beispiel: ***cal***

Der Befehl ***rm*** löst die Endschalterfahrt zum rm-Endschalter aus, hierbei werden

die ativen Achsen in positive Richtung positioniert, bis der rm-Schalter betätigt ist. Die Steuerung positioniert danach wieder in Richtung negativer Positionswerte bis vor den Endschalter.
Beispiel: ***rm***

getcaldone 115
Mit dem Befehl ***getcaldone*** wird abgefragt ob die Endschalterfahrt ***cal*** oder ***rm*** ausgeführt wurde.
Beispiel: ***1 getcaldone***

setsw 117
Mit dem Befehl ***setsw*** werden die Endschaltereingänge cal und rm an das Schaltverhalten der Endschalter angepasst. Die Einstellung "ignorieren" schaltet den Endschaltereingang ab. Es sind folgende Funktionen möglich:
Beispiel: ***0 0 1 setsw***

getsw 118
Das Kommando ***getsw*** liest die Einstellung der Endschaltereingänge zurück.
Beispiel: ***3 getsw -1 getsw***

getswst. 119
Das Kommando ***getswst*** zeigt den Schaltzustand der Endschaltereingänge cal und rm an.
Beispiel: ***3 getswst***

setcalswdist. 121
Mit dem Kommando ***setcalswdist*** kann jeder Achse ein zusätzlicher Abstand zu den Endschaltern vorgegeben werden.
Dieser Abstand wirkt auf beide Endlagen der Achse.
Beispiel: ***0.5 1 setcalswdist***

getcalswdist. 122
Das Kommando ***getcalswdist*** liest Einstellung von ***setcalswdist*** zurück.
Beispiel: ***-1 getcalswdist***

setlimit 123
Mit dem Kommando ***setlimit*** werden Softlimits für alle Achsen festgelegt, damit lässt sich der Verfahrbereich der Achsen beliebig einschränken.
Beispiel: ***0 0 -10 12 25 30 setlimit***

getlimit 125
Der Befehl ***getlimit*** ermittelt die Verfahrgrenzen, die durch die Befehle ***cal, rm***, ***setlimit*** oder den manuellen move in die Endschalter festgelegt wurde. Abhängig von der Einstellung ***setdim*** werden die Werte in 1, 2 oder 3 Zeilen zurückgeliefert.
Beispiel: ***getlimit***

ncal. 127
Mit dem Kommando ***ncal*** ist es möglich jede Achse separat zu ihrem unteren Endschalter zu bewegen.

Sicherheitsfunktionen

Position / Bezugspunkt / Koordinatensytem

Der Befehl ***ico*** setzt ein gedrehtes Koordinatensystem wieder auf den Standardwert zurück.
Beispiel: ***ico***

getico 169
Mit dem Befehl ***getico*** wird überprüft ob das Koordinaten-system der Steuerung durch das Kommando ***align*** gedreht wurde.
Beispiel: ***getico***

Statusabfragen

status (st) 173
Mit dem Kommando ***status*** wird der augenblickliche Betriebszustand der Steuerung abgefragt.
Beispiel: ***st***

geterror (ge) 177
Mit dem Kommando ***geterror*** wird die Steuerung auf allgemeine Systemfehler überprüft. Es wird immer der zuletzt aufgetretene Fehler angezeigt.
Beispiel: ***ge***

getmerror (gme) 179
Das Kommando ***getmerror*** liest den Maschinen-fehlerspeicher aus.
Beispiel: ***gme***

gsp 181
Der Befehl ***gsp*** ermittelt die Anzahl der Elemente auf dem Parameter Stack.
Beispiel: ***gsp***

getticks (gt) 183
Das Kommando ***gt*** liefert die Anzahl der ausgeführten Prozessorzyklen zurück. Jeder Zähler entspricht 250µs. Nach 298 Stunden wird der Zähler zurückgesetzt.
Beispiel: ***gt***

Input / Output Funktionen

setout 187
Das Kommando ***setout*** schaltet die Open Collector Ausgänge Dout1- Dout3.
Beispiel: ***7 setout***

getout 188
Das Kommando ***getout*** liest den mit ***setout*** eingestellten Ausgabewert zurück.
Beispiel: ***getout***

setaout 189
Das Kommando ***setaout*** erzeugt eine analoge Ausgangsspannung zwischen 0 und 1000mV mit einer Auflösung von 8-Bit.
Beispiel: ***100 1 setaout***

Closed Loop Kommandos

Mit dem Kommando ***getsp*** können die Reglerparameter einzeln ausgelesen werden

Beispiel: ***1 getsp***

setscaleinterface 211
Mit dem Kommando ***setscaleinterface*** wird das Closed Loop Interface ausgewählt.

Beispiel: ***2 1 setscaleinterface***

getscaleinterface 212
Das Kommando ***getscaleinterface*** überprüft welche der beiden Closed Loop Schnittstellen initialisiert ist.

Beispiel: ***1 getscaleinterface***

setscaletype 213
Mit dem Kommando ***setscaletype*** wird das Messinterface der Steuerung an den Typ des Messsystems angepasst.

Beispiel: ***1 1 setscaletype***

getscaletype 214
Das Kommando ***getscaletype*** überprüft für welchen Typ Messsystem das Messinterface initialisiert ist.

Beispiel: ***1 getscaletype***

setclfactor 215
Das Kommando ***setclfactor*** wird für die Anpassung an einen digitalen Drehgeber verwendet. Damit wird Strichauflösung und Zählrichtung des Gebers übernommen.

Beispiel: ***- 500 3 setclfactor***

getclfactor 216
Der Befehl ***getclfactor*** liest die Einstellungen von ***setclfactor*** zurück.

Beispiel: ***1 getclfactor***

setclperiod 217
Mit dem Befehl ***setclperiod*** wird das analoge und digitale Messinterface der Steuerung an lineare Messsysteme mit Ausgängen RS-422, 1Vss oder MR, sowie Drehgebern mit 1Vss Ausgang angepasst.

Beispiel: ***- 0.002 3 setclperiod***

getclperiod 221
Der Befehl ***getclperiod*** liest die Einstellungen von ***setclperiod*** zurück.

Beispiel: ***1 getclperiod***

refmove 223
Das Kommando ***refmove*** positioniert alle aktiven Achsen zur Referenzmarkierung des Längenmesssystems.

Beispiel: ***100 refmove***

setclwindow 225
Der Befehl ***setclwindow*** definiert ein Positionszielfenster

Trigger-Output-Funktionen

Mit dem Kommando ***wtot*** (wait_time out_trigger) wird ein Triggersignal verzögert ausgegeben.
Beispiel: ***10 1 10 0 1 wtot***

setrptdata 247
Das Kommando ***setrptdata*** initialisiert den Positions- Intervall-Trigger mit dem Triggerausgangssignale in gleichen Abständen erzeugt werden können.
Beispiel: ***0 10 startrpt 20 20 m***

getrptdata. 249
Das Kommando ***getrptdata*** liefert die Parameterwerte der Funktion "Positions-Intervall-Trigger".
Beispiel: ***getrptdata***

startrpt 251
Das Kommando ***startrpt*** aktiviert den Positions-Intervall-Trigger und gibt die Start- und Stop-Koordinate des Triggerintervalles an.
Beispiel: ***10.234 12.56 startrpt***

Trigger-Input Funktionen

setotmode 255
Mit dem Kommando ***setotmode*** wird für das Kommando wpot die Soll- oder Ist-Position als Triggerquelle festgelegt.
Beispiel: ***3 setotmode***

getotmode 256
Das Kommando ***getotmode*** liefert die mit setotmode eingestellten Einstellungen für die Funktionen ***wpot*** und setpc zurück.
Beispiel: ***getotmode***

setpcin 257
Das Kommando ***setpcin*** initialisiert den Triggereingang für die Funktion "position capture".
Beispiel: ***1 3 setpcin***

getpcin 258
Das Kommando ***getpcin*** liefert die Einstellungen der Funktion "position capture".
Beispiel: ***getpcin***

setpc. 259
Das Kommando ***setpc*** aktiviert oder deaktiviert die Funktion "position capture",
Beispiel: ***1 setpc***

getpc. 260
Das Kommando ***getpc*** liefert den Status der Funktion "position capture". Zusätzlich wird die Anzahl der empfangenen Triggersignale angezeigt.
Beispiel: ***getpc***

waitintrig (wit) 261
Mit dem Kommando ***wit*** (wait_in_trigger) wird die Kommandoausführung so lange

unterbrochen, bis ein digitales Eingangsignal anliegt und die Freigabe erzeugt.
Beispiel: ***1 1 wit [SP] st***

Joystick / Handrad

getjoyspeed 276
Das Kommando ***getjoyspeed*** liest die eingestellte maximale Geschwindigkeit für den Joystickbetrieb zurück.
Beispiel: ***getjoyspeed***

setnjoyspeed (njs). 277
Das Kommando ***setnjoyspeed*** ermöglicht es die Joystickgeschwindigkeit der Achsen individuell festzulegen.
Beispiel: ***20 1 setnjoyspeed***

getnjoyspeed (njs) 278
Das Kommando ***getnjoyspeed*** liest die Einstellung der achsspezifischen Joystickgeschwindigkeit zurück.
Beispiel: ***1 getnjoyspeed***

setjoybspeed 279
Mit dem Kommando ***setjoybspeed*** kann eine zusätzliche Joystickgeschwindigkeit festgelegt werden. Diese wird mit einem Schalter oder Taster am Joystick aktiviert.
Beispiel: ***0.01 setjoybspeed***

getjoybspeed 280
Mit dem Befehl ***getjoybspeed*** wird die Einstellung der zweiten Joystickgeschwindigkeit abgefragt.
Beispiel: ***getjoybspeed***

setjoyassign 281
Mit dem Kommando ***setjoyassign*** kann die Wirkung der Joystickauslenkung auf die Motordrehrichtung eingestellt werden. Zusätzlich ist es möglich die Zuordnung der Joytickachsen zu den Motorachsen zu verändern.
Beispiel: ***1 1 setjoyassign***
Beispiel: ***2 1 setjoyassign***

getjoyassign 283
Das Kommando ***getjoyassign*** liest die Zuordnung der Joystickachsen zu den Motorachsen bzw. deren Wirkung auf die Motordrehrichtung zurück.
Beispiel: ***1 getjoyassign***

setjoydiag 285
Das Kommando ***setjoydiag*** aktiviert die Joystickdiagnose, mit der die analogen Ausgangspegel des Joysticks im Terminal angezeigt werden.
Beispiel: ***1 setjoydiag***

getjoydiag 286
Mit dem Kommando ***getjoydiag*** wird überprüft ob die Joystickdiagnose eingeschaltet ist.
Beispiel: ***getjoydiag***

setwheel. 287
Kommando ***getwheel*** initialisiert den Handrad Modus.
Beispiel: ***1 setwheel [cr] save [cr] reset [cr]***

Systemkommandos

Fehlerkorrektur

Das Kommando ***setpdat*** dient zur Eingabe der Daten für die Positionsfehlerkorrektur.
Beispiel: ***0.5 0.1 0.5 1.2 -0.5 1.2 0 6 1 setpdat***

getpdat 328
Mit dem Kommando ***getpdat*** werden die eingetragenen Korrekturwerte sequentiell ausgelesen.
Beispiel: ***12 1 getpdat***

setblc 329
Das Kommando ***setblc*** aktiviert oder deaktiviert die Funktion "backlash-compensation".
Beispiel: ***1 1 setblc***

getblc 330
Das Kommando ***getblc*** liefert den Status der Funktion "backlash-compensation".
Beispiel: ***1 getblc***

setblcd 331
Mit dem Kommando ***setblcd*** wird der Korrekturwert (Distanz) der Funktion "backlash compensation" eingestellt.
Beispiel: ***0.001 1 setblcd***

getblcd 332
Das Kommando ***getblcd*** liefert die Einstellung der back-lash Kompensation zurück.
Beispiel: ***1 getblcd***

Makrobefehle

beginmakro / endmakro 345
Die Kommandos ***beginmakro*** und ***endmakro*** sind Steuerwörter, die den Beginn und das Ende eines Makros kennzeichnen.
```
Beispiel: beginmakro
Beispiel: cal
Beispiel: rm
Beispiel: 0 0 0 m
Beispiel: endmakro
```

startmakro 347
Mit dem Kommando ***startmakro*** wird das im Makro-Exe Puffer befindliche Makro ausgeführt.
Beispiel: ***startmakro***

listmakro 349
Mit dem Kommando ***listmakro*** wird die Anzahl der im Makro-Exe Puffer befindlichen Symbole ausgelesen. In den Puffer können maximal 4000 Symbole übertragen werden.
Beispiel: ***listmakro***

Ctrl-D 351

Mit dem ASCII Zeichen ***Ctrl-D*** wird das momentan ausgeführte Makro oder der Download des Makro abgebrochen.

`Beispiel:` ***Ctrl-D***