Handbook

Venus-1
Command language

for

Corvus

high resolution positioning controller

SMC Corvus
SMC Corvus eco
SMC PCI

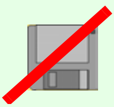
About this documentation

This handbook provides detailed information on the Venus-1 command language for the positioning controllers Corvus TT, Corvus eco, Corvus PCI
The commands are separated in functional groups to improve the overview.

The last chapter lists each command and gives a brief description of the command's function.

Symbols in this documentation

To clarify the content following symbols are used.

Symbol	Description
	Warning. This information must be observed strictly.
	Important information
	Indicates that this function can be enabled with a release code.
	This function must be installed from the factory personal or experts.
Venus-1	Venus-1 commands are indicated with this formatting style.
	This configuration can not be stored into the flash memory. It is lost after power off.
	This configuration can be stored with command ***save*** into the controller flash memory.

Contents

Communication

Velocity and acceleration

Positioning commands

Limit Switch functions

Safety functions

position / origin / coordinate system

Status requests

Input / Output functions

Closed Loop commands

Trigger Output functions

Trigger-Input functions

Joystick / Handwheel

System commands

Position error correction

Macro functions

Introduction in Venus-1

Venus-1 is an interpreter language

Venus-2 commands consist of ASCII-signs which are interpreted in the controller and immediately executed.
A software development surrounding to produce the control programs is not needed.
The commands can be produced by any Host and whatever programming language you are using, on condition that there is an access to the RS-232 or Ethernet interface.
In the simplest way the commands are directly transmitted to the controller via an ASCII terminal.

Venus-1 history

Venus-1 for Corvus has been developed on the basis of the interpreter language Venus-1 for the controllers mc-compact, smc-compact, MC-2000 and MC-3000.
The fundamental command construction is identical.
All basic functions are compatible to the former version.

Command syntax for parametrisation

The parameterisation commands are assembled following this scheme:

[parameter] _ [axis index] _ [command] _

_ blank, (space) or (SP)

Parameter

The parameter transmits a value without any unit.
If several parameters are prescribed for one command, they have to be divided by a blank (SP).

The following numbers and characters are permitted for parameters:

Letters	not allowed
Numbers	0-9
Characters	+ -.

-1 Parameter

Most of the get-commands allow the combination -1 to read out the settings of all axes.

For example:

With the command ***2 getpitch*** the spindle pitch of Axis-2 is asked.
The command ***-1 getpitch*** returns the pitch setting of all axes.

Axis index

With the axes index the target axis is addressed. The number of the index is equal with the labeling at the motor connector.

Axis label	Axis index
Axis-1	1
Axis-2	2
Axis-3	3

Commands

For the parametrization the commands are named with get or set. It consists of several ASCII characters, capitalization is distinguished.

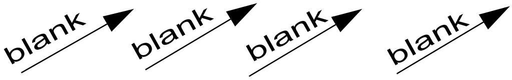
The following letters are allowed for commands:

ASCII-Charac-ters	a-z A-Z
Umlauts	not allowed
Numbers	not allowed

Command syntax for positioning commands

The positioning commands are assembled following this scheme:

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



_ blank, (space) or (SP)

Axis-1, Axis-2, Axis-3

For the positioning, absolute or relative coordinates are transferred to the controller.
The values must be separated by a blank (SP).
The number of position values to be transferred depends on the setting of ***setdim.***

setdim	Axis values that must be transferred
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

If insufficient coordinates are transferred, the move command will not be executed.
Useless coordinates will remain on the stack
The following letters are allowed for position coordinates:

Letters	no
Numbers	0 - 9
Characters	+ -.

Command ending character for transmitting

In the **host mode** data which are transmitted have to be completed with a blank
[parameter] _ [axis index] _ [command] _

In the **terminal mode** the command ending is executed by [CR] (carriage returns).
[parameter] _ [axis index] _ [command] [CR]

Command ending character for receiving

[1st parameter] _ [2nd parameter] _ [n-parameter] [CR][LF]

Data which are transmitted from the controller are always completed with ASCII [CR] and [LF]. Some data requests return parameters in several lines. In these cases each line is also completed with [CR] and [LF].
How many lines a request returns is mentioned in the command description.

Table of important ASCII signs for programming

ASCII Code	**Sign**	**Dez**	**HEX**
CR	Ctrl-M	13	0xD
LF	Ctrl-J	10	0xA
SP		32	0x2 0
ETX	Ctrl-C	3	0x3

Command execution

For the correct programming it is important to know the internal courses during the execution of the interpreter commands.

The ASCII data transmitted by a host run through the following areas of the controller:

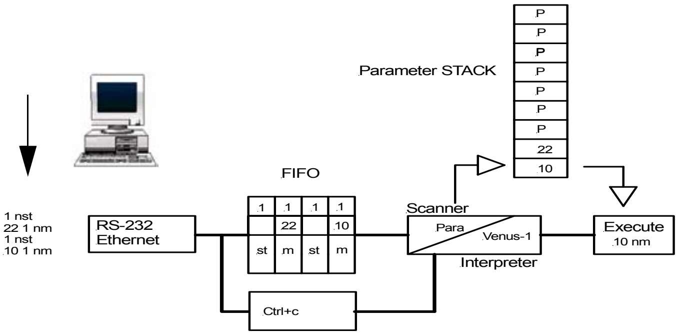
- command input FIFO
- scanner and stack
- interpreter

Command input FIFO

The ASCII commands are transferred from the communication interface (Host) to the data input memory and remain there until they are processed. The memory possesses a FIFO structure. This FIFO is able to accept up to 256 ASCII signs.
There is no data flow control during the transmission of the data, i.e., an overflow of the FIFO would not be recognized. For that reason not to much data should be transmitted to the controller.
With the controller switched off, the FIFO is cleared.

Scanner - Interpreter - Stack

The content of the command FIFO will be read by the scanner and divided in parameters and commands.
The parameters are transmitted to a stack which can accept up to 99 values.
The Commands are directly passed to the interpreter, as soon as it is free.



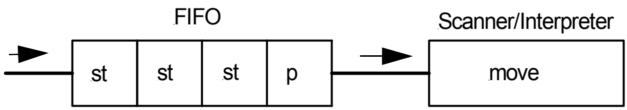
Blocking and non blocking commands

During the interpreter executes a move it is able to execute several other commands parallel to it. These commands are called non blocking commands.

On the other side there are commands that will be only executed until the move is finished.
If such a command is stored in the FIFO all other commands behind it will be blocked until the blocking command is executed and removed from the FIFO.
These commands are called blocking command.

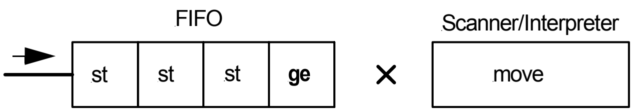
Examples of blocking and non blocking commands

The interpreter is able to execute several commands at the same time.
Below mentioned, the interpreter executes the instruction ***p*** and is also free to process 3x ***st***.



The interpreter has been blocked with the command ***ge***.

The interpreter is executes the command ***move***. The FIFO contains the command ***ge***, this blocks the interpreter for the execution of further commands until ***move*** is completed
After ***ge*** is executed the commands ***st*** are processed.



Examples of non blocking commands

The commands below do not block the interpreter. These commands are also executed if the interpreter processes a ***move*** command.

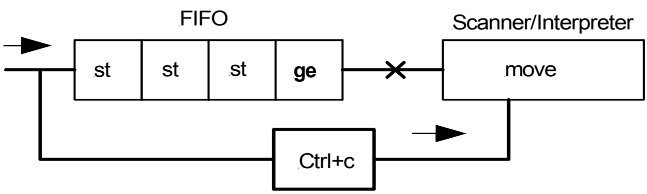
Command	Description
st	Status
p	Current position
getin	Read digital Input
setout	Write digital Output
abort	Aborts the current command Attention: This command has to pass through the FIFO it's execution could be delayed with a blocking command.
Ctrl-C	Aborts the current command This commands has **not** to pass the FIFO and could not be delayed with a blocking command.

Unlock interpreter

A lasting blockade of the interpreter is not possible, because all commands are finally processed this will unlock the interpreter.
To accelerate Interpreter unlocking, command ***Ctrl-C*** can be used to abort the commands.

Terminate command execution

Ctrl-C has a direct access to the interpreter and will abort the current command in the interpreter.
It is not possible to erase the FIFO totally.



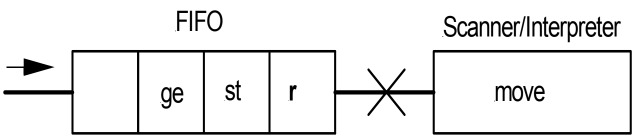
Generate an automatic status reply message

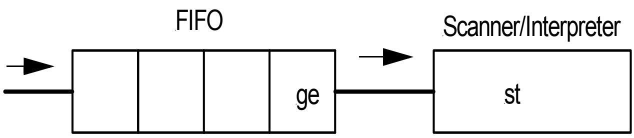
With the following sequence of instructions the blocking effect of the interpreter can be used to generate an automatic status reply.

10 10 2 move
0 0 0 r
st
ge

Effect:
An automatic status reply is generated, after the instruction ***10 10 2 move*** is executed.

Description:
The instruction ***0 0 0 r*** has only the function to block the interpreter and prevent the execution of ***st***.
After ***move*** is executed, ***0 0 0 r*** is processed with no effect, because it is a relative positioning with 0mm.
Afterwards the command ***st*** produces the desired status feedback.

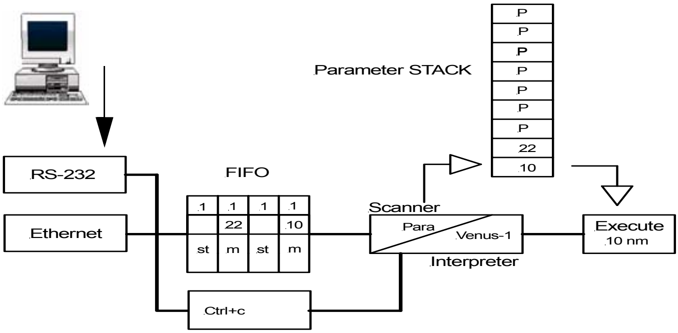




Corvus communication concept

Ethernet and RS-232

RS-232 is the standard communication interface of Corvus.
Optionally the Ethernet interface can be released.
Both interfaces are always ready to receive data.
The data feedback is automatically send to the interface from where the data inquiry comes.
Terminal and host mode are supported from both interfaces.



Basic settings

setpitch	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setpitch*** adapts the controller to the transmission ratio of the drive train.

$$\text{pitch} = \frac{\text{resulting move distance}}{\text{number of motor revolutions}}$$

Syntax:

[pitch] [axis] ***setpitch***

	Range	Unit
[pitch]	0.0001 to 4095	mm
[axis]	1, 2, 3	

Related command:

getpitch

Example:

4.0009 1 setpitch

Examples in the following page

Examples:

Drive mechanism with ball screw at Axis-1:

Lead screw with pitch = 2mm
Each motor revolution produces a move distance of 2mm
Pitch = 2mm / 1 rev. = 2

Setting : *2 1 setpitch*

Drive mechanism with lead screw and gear box

Lead screw pitch = 4mm
Gear = 120:1
120 motor revolutions produce a move distance of 4mm
Pitch = 4mm / 120 rev. = 0.0333

Setting : *0.0333 [axis] setpitch*

Rotation table (axis unit is normalized to degrees)

Each motor revolution produce a rotation angle of 360°
Pitch = 360° / 1 rev. = 360

Setting : *360 [Axis] setpitch*

Rotation table with gear 120:1 (axis unit is normalized to degrees)

120 motor revolutions produce a rotation angle of 360°
Pitch = 360° / 120 rev. = 3

Setting : *3 [Axis] setpitch*

getpitch

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getpitch*** returns the pitch setting of the axis.

Syntax:

[axis] ***getpitch***

	Range
[axis]	-1, 1, 2, 3

Reply:

[drive train transmission ratio]

	Unit
[drive train transmission ratio]	mm

Example:

2 getpitch

Reply:
4.000900

-1 getpitch

Reply:
4.000900
2.000000
2.000000

setunit

setunit	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setunit*** the physical units of the Axis-specific parameters are defined.

The units of velocity and acceleration are determined from the unit setting of Axis-0.
The unit of the commands ***setcalvel, setncalvel, setrmvel, setnrmvel*** and ***setrefvel*** are fixed to rev./s (r/s).
For the reason of compatibility with older controllers, the unit microstep is emulated from Corvus.
In this case the positioning resolution is reduced.

1 Microstep = 1 motor revolution / 40000 steps

Syntax:

[index] [axis] ***setunit***

	Range
[index]	0, 1, 2, 3, 4, 5, 6
[axis]	0, 1, 2, 3

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

Related command:

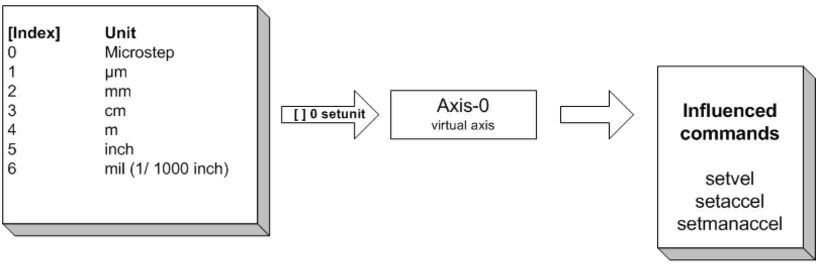
getunit

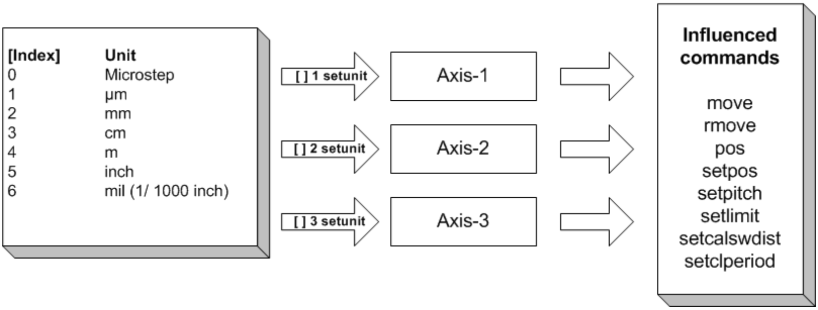
Example:

2 0 setunit

The unit of the virtual 0-Axis is set to **mm**.
As a result the settings of velocity (*sv*) and acceleration (*sa*) are referenced to mm/s resp. mm/s².

1 1 setunit
The physical unit of axis-1 is set to **µm**.





getunit

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getunit*** returns the settings the physical units.

Syntax:

[axis] ***getunit***

	Range
[axis]	-1, 0, 1, 2, 3

Reply:

[index]

	Range
[index]	0, 1, 2, 3, 4, 5, 6

Example:

1 getunit

Reply:
1

-1 getunit

Reply:
2 1 1 1

setumotmin

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setumotmin*** determines the motor phase voltage at stand still and lower speeds.

A greater index value, will increase the motor voltage. This implies an increased motor phase current and will produce more holding torque at stand still.

Attention: It will also increase the power consumption at the motor and motor driver.

Syntax:

[index] [axis] ***setumotmin***

	Index range	Unit
[index]	0 - 3000	mV
[axis]	1, 2, 3	

Related command:

getumotmin, setumotgrad

Example:

2000 1 setumotmin

getumotmin

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getumotmin*** returns the setting of *umotmin*.

Syntax:

[axis] ***getumotmin***

	Range
[axis]	-1, 1, 2, 3

Reply:

[index]

	Range
[index]	0 - 3000

Example:

1 getumotmin

Reply
2000

-1 getumotmin

2000
1000
750

setumotgrad

setumotgrad	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setumotgrad*** determines the motor voltage in the middle and upper speed range.

A greater index value, will increase the motor voltage and implies an increased motor phase current and torque during the move.

Attention: This will produce more power consumption at the motor and motor driver.

Syntax:

[index] [axis] ***setumotgrad***

	Range
[index]	0 - 300
[axis]	1, 2, 3

Related commands:

getumotgrad, setumotmin

Example:

70 1 setumotgrad

getumotgrad

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getumotgrad*** returns the setting of umotgrad.

Syntax:

[axis] ***getumotgrad***

	Range
[axis]	-1, 1, 2, 3

Reply:

[index]

	Range
[index]	0 - 300

Example:

1 getumotgrad

Reply:
50

-1 getumotgrad

Reply:
50
40
100

setpolepairs	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setpolepairs*** adapts the controller to the number of the stepper motor pole-pairs.

The relationship between motor type and pole-pairs is shown in the following table.

	pole pairs
Hybrid Stepper Motor with full step size=1.8°	50
Hybrid Stepper Motor with full step size= 0.9°	100

Syntax:

[pole-pairs] [axis] ***setpolepairs***

	Range
[pole pairs]	50, 100 other on request
[axis]	1, 2, 3

Related command:

getpolepairs

Example:

50 1 setpolepairs

Controller Axis-1 is configured to a Stepper Motor with 100 poles or 50 pole-pairs (full step size 1.8°).

# ***getpolepairs***	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getpolepairs*** returns the configured number of pole-pairs.

Syntax:

[axis] ***getpolepairs***

	Range
[axis]	-1, 1, 2, 3

Reply:

[value]

	Range
[value]	50, 100

Example:

1 getpolepairs

Reply:
50

-1 getpolepairs

Reply:
50 100 50

setaxis	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setaxis*** enables or disables the specified axis for positioning tasks.
setaxis also has an effect on the commands ***pos***, ***setpos*** ***cal, rm*** and the hardware limits
The settings are significant for the programmable and manual move.

Syntax:

[index] [axis] ***setaxis***

	Range
[index]	0, 1, 2, 3, 4
[axis]	1, 2, 3

[index] = 0:

The axis is disabled for all moves.
The commands ***cal***, ***rm*** and ***0 0 0 setpos*** will clear the actual position, but will not change the hardware limits of the axis.

[index] = 1:

The axis is enabled for all moves.
The commands ***cal***, ***rm*** and ***0 0 0 setpos*** will clear the actual position to zero and reset the hardware limits of the axis.

[index] = 2:

The axis is restricted enabled because the limit switch moves cal / rm wil not be executed.
The commands ***cal***, ***rm*** and ***0 0 0 setpos*** clear the actual position to zero but will not change the hardware limits of the axis.

[index] = 3:

The axis is disabled for all moves.
The commands ***cal***, ***rm*** and ***0 0 0 setpos*** will **not** clear the actual position and **not** change the hardware limits of the axis.

[index] = 4:

The axis is restricted enabled because the limit switch move procedure cal / rm will not be executed.
The commands ***cal***, ***rm*** and ***0 0 0 setpos*** will **not** clear the actual position and **not** change the hardware limits of the axis.

Related command:

getaxis

Example:

1 3 setaxis

getaxis

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getaxis*** returns the setting of ***setaxis.***

Syntax:

[axis] ***getaxis***

	Range
[axis]	-1, 1, 2, 3

Reply:

[index]

	Range
[index]	0, 1, 2, 3, 4

Example:

2 getaxis

Reply:

2

-1 getaxis

Reply:

1 2 2

setpowerup

Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setpowerup*** it is possible to execute fixed commands automatically after power up.

Each single power up command is assigned to a binary state (D0-D4). To combine several power up commands to a command sequence, their binary states must be added.

Syntax:

[Parameter] ***setpowerup***

	Meaningful combinations
[Parameter]	0, 1, 2, 3, 4, 5, 6, 7, 15, 16

bin (dec)	**Command**	**Description**
D0 (1)	*1j*	Joystick On/Off With power up = 0 always the before stored Joystick setting is valid.
D1 (2)	*cal*	The axes are moving to the cal limit switches
D2 (4)	*rm*	The axis are moving to the rm limit switches
D3 (8)	*randmove*	random move of all enabled axes within the limits The cal- and rm-limits must be determined first to prevent a hard limit crash
D4 (16)	*cal/rm/* *0 0 0* move* *depends on setdim	The axes first move to the limits (cal/rm) and than to the origin.

Example:

1 setpowerup
Joystick is enabled after power up.

15 setpowerup
After power up, the controller determines the limits of all active axes, than the axes are moved to randomized coordinates.

# getpowerup	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getpowerup*** returns the Power up command settings of the controller.

Syntax:

getpowerup

Reply:

[Parameter]

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

Example:

getpowerup

Reply: 15

setphaseares

setphaseares	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 3.6.3

Description:

With command ***setphaseares*** it is possible to reduce the resolution of the motor drivers in incremental steps.
In the lowest resolution (2 Bit), the motor drivers are working in the stepper macro step mode.
The resolution depends also on the settings of *setpolepairs.*

Syntax:

[resolution (Bit)] [axis] ***setphaseares***

	Range
[resolution] *	2....16
[axis]	1, 2, 3

* factory setting = 16

Polepairs = 50

Bit	**Resolution (steps/rev.)**
2	200
16	>600.000

Polepairs = 100

Bit	**Resolution (steps/rev.)**
2	400
16	>1.200.000

** see command *setpolepairs*

Example:

2 1 setphaseares
Makro step resolution setting for Axis-1.

getphaseares

Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 3.6.3

Description:

Command ***getphaseares*** returns the motor resolution value of the selected axis.

Syntax:

[Axis] ***getphaseares***

Reply:

[Resolution (Bit)]

	Range
[Resolution]	2......16

Polepairs = 50

Bit	Resolution (steps/rev.)
2	200
16	>600.000

Polepairs = 100

Bit	Resolution (steps/rev.)
2	400
16	>1.200.000

** see command *setpolepairs*

Example:

2 getphaseares

setmotiondir

setmotiondir	Corvus TT	Corvus eco	Corvus PCI

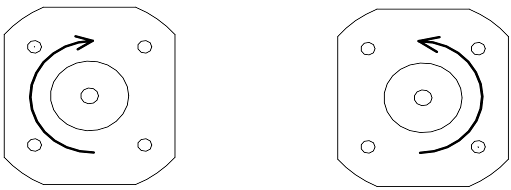
Valid from firmware version 4.4.0

Description:

With command ***setmotiondir*** the factory assigned relationship between the direction of motor rotation and the motion direction can be reversed.

In the case of a reversed assignment between motor direction and motion direction, the function of the limit switch inputs are changed.
This means, during the calibration move cal, the controller expects the limit switch activity at the rm input and during the rm move at the cal input.

Motor direction if a move to positive coordinates is executed.



Factory setting

Motor direction is changed with ***setmotiondir***

Syntax:

[Function] [axis] ***setmotiondir***

[Function]	Description
0	factory assigned motor direction and motion direction
1	Relationship between motor direction and motion direction is changed.

Example:

1 1 setmotiondir

getmotiondir

Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.4.0

Description:

Command ***getmotiondir*** indicates, if the relationship between motor direction and motion direction differs from the factory settings.

.

Syntax:

[Axis] ***getmotiondir***

	Range
[Axis]	1, 2, 3

Reply:

[0,1]

	Description
[0]	Factory settings
[1]	Motor direction and the function of the limit switch inputs are changed.

Example:

1 getmotiondir

Communication

mode	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***mode*** enables Terminal or Host Mode.

In **Terminal Mode** a terminal mask is transmitted from the controller to the Host or Terminal.
The mask provides a Venus command input line
and displays the actual position.

In **Host Mode** the controller returns data only after a command request.

Alternatively it is possible to enable the mode with
DIP-Switch 6 directly at the controller, see Corvus Manual.

Syntax:

[index] ***mode***

[index]	Description
0	Host Mode
1	Terminal Mode

Example:

1 mode

Corvus is switched to Terminal Mode

```
VENUS-1 (Corvus) Interpreter Version: 4.55 Copyright 2008 by ITK Dr.Kassen
5  X:   0.00000
5  Y:   0.00000


Command[ 0]:
```

setipadr

setipadr	Corvus TT		

Description:

With the command ***setipadr*** the controller Ethernet IP-Address can be defined.

The following Ethernet settings are fixed:

Subnet mask: 255.255.255.0
Port: 23
Socket: TCP/IP (Winsocket)

Syntax:

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

The address elements have to be divided by a blank.

Related command:

getipadr

Example:

192_168_128_0_setipadr

getipadr

getipadr	Corvus TT		

Description:

The command ***getipadr*** returns the controller IP-Address.

Syntax:

getipadr

Reply:

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

Related command:

getmacadr

Example:

getipadr

Reply:

192.168.128.0

The replied address elements are separated with a dot.

Velocity and acceleration

setvel (sv)

setvel (sv)	Corvus TT	Corvus eco	Corvus PCI

Description:

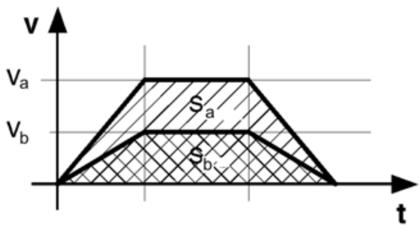
Command ***setvel*** configures the programmed move velocity **v**a.

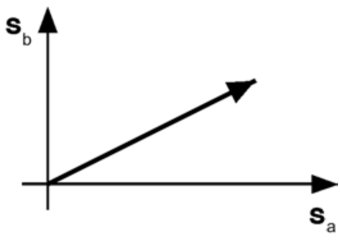
In consideration to the given move distances of all active axes, the controller calculates an individual velocity profile for each axis.
The setting of ***setvel*** relate to the axis, that moves the longest distance, see diagram.

The maximum velocity **V**b or **V**c depends on the distance ratio to the axis with the longest travel.

The motor rotation speed is determined from the ***setvel*** value and the *setpitch* value.

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





In the programmed move, all axes are starting and ending the move simultaneously.

Syntax:

[velocity] ***setvel***

	Range
minimum velocity	15,26 nm/s
maximum velocity	45 rev./s, pitch =4 mm -> 180mm/s 60 rev./s (option)

	Unit
[velocity]	Unit of the virtual 0-Axis

Example:

100 sv

getvel (sv)

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getvel (gv)*** returns the setting of ***setvel***.

Syntax:

getvel

Reply:

[velocity]

	Unit
[velocity]	selected unit

Example:

gv

Reply:
180.000000

setaccel (sa)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setaccel (sa)*** defines the acceleration ramp with which the controller executes the programmed move.
The axes are linear interpolated, this means the controller starts and stops all axes simultaneously.
The value of ***setaccel*** relates to the axis which must travel the longest distance.
The maximum acceleration of the other axes depends on the ratio to the axis with the longest travel.

Acceleration and deceleration ramp are identical.

Syntax:

[Acceleration] ***setaccel***

	Range	Unit
[Acceleration]	0 - $\frac{100000}{s^2}$ * $\frac{\text{pitch [unit]}}{\text{polepairs *}}$	unit/s²

* polepairs see command ***polepairs***

Related commands:

getaccel / setmanaccel

Example:

500 sa

getaccel (ga)

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getaccel (ga)*** returns the setting of *setaccel.*

Syntax:

getaccel

Reply:

[Acceleration]

	Range	unit
[Acceleration]	$0 - \frac{100000}{s^2} * \frac{\text{pitch [unit]}}{\text{pole pairs}}$	unit/s²

Pole pairs: 50 or 100 (see command *setpolepairs*)

Example:

ga

Reply:
2400000.000000 (unit = µm)

setaccelfunc

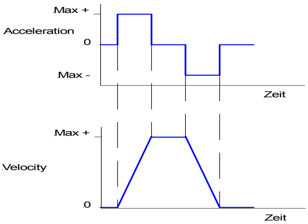
setaccelfunc	Corvus TT	Corvus eco	Corvus PCI

Description:

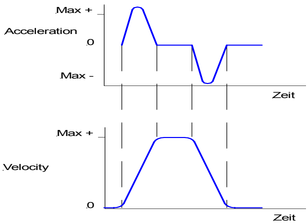
The command ***setaccelfunc*** defines the acceleration function with which the positioning task is executed.

Following functions are possible:

- Linear acceleration (trapezoidal)
- Sin² acceleration (S-curve)



Linear acceleration



sin² acceleration

Syntax:

[index] ***setaccelfunc***

[index]	Description
0	Linear acceleration
1	Sin² acceleration

Related command:

getaccelfunc

Example:

1 setaccelfunc

getaccelfunc

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getaccelfunc*** returns the adjusted acceleration function.

Syntax:

getaccelfunc

Reply:

[index]

	Range
[index]	0, 1

[index]	**Description**
0	Linear acceleration
1	Sin^2 acceleration

Example:

getaccelfunc

setmanaccel

setmanaccel	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setmanaccel*** defines the acceleration ramp for the manual operation with Joystick or Handwheel.

Syntax:

[Acceleration] ***setmanaccel***

	Range	Unit
[Acceleration]		unit 0-Axis /s²

Related commands

getmanaccel / setaccel

Example:

100 setmanaccel

# *getmanaccel*	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getmanaccel*** returns the setting of *setmanaccel.*

Syntax:

getmanaccel

Reply:

[Value]

	Range	Unit
[Value]		unit 0-Axis /s²

Example:

getmanaccel

Reply:
2400.000000

setcalvel

setcalvel	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setcalvel*** defines two velocities for the cal limit-switch move. The setting is significant for all axes.

1. velocity to move in a negative direction to the switch
2. velocity to move in a positive direction out of the switch

For the reason of compatibility with the older controllers, the unit of this velocity is defined in revolutions/s.
The resulting velocity in mm/s depends on the pitch value of the virtual 0-Axis (see command ***setpitch***).

Syntax:

[velocity] [index] ***setcalvel***

[index]	Description
1	Velocity to the limit-switch
2	Velocity out of the limit-switch

	Range	Unit
[Velocity]	0 - 45	revolution/s
[index]	1, 2	-

Related commands:

getcalvel, setrmvel

Example:

2 0 setpitch (virtual 0-Axis)
2 1 setcalvel
1 2 setcalvel
The pitch of 0-Axis is adjusted to 2mm.
The controller is moving to the cal limit-switch with 2 rev./s (4 mm/s) and with 1 rev./s (2 mm/s) out of the switch.

getcalvel

getcalvel	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getcalvel*** returns the adjusted velocities for cal limit-switch move.

Syntax:

getcalvel

Reply:

[velocity-1] cr
[velocity-2] cr

	Range	Unit
[velocity-1]	0 - 45	rev. /s
[velocity-2]	0 - 45	rev. /s

Example:

getcalvel

Reply:

2.000000
0.250000

setncalvel	Corvus TT	Corvus eco	Corvus PCI

Valid with firmware version 4.0

Description:

Command ***setncalvel*** defines the two velocities for the ncal limit-switch move.

1. Velocity to move in negative direction into the switch
2. Velocity to move in positive direction out of the switch

For the reason of compatibility with the older controllers, the unit of this velocity is defined in revolutions/s.
The resulting velocity in mm/s depends on the pitch value of the virtual 0-Axis (see command ***setpitch***).

Syntax:

[Velocity] [index] [axis] ***setncalvel***

	Range	Unit
[Velocity]	0 - 45	revolution/s
[index]	1, 2	-
[axis]	1, 2, 3,	-

[index]	Description
1	Velocity to the limit-switch
2	Velocity out of the limit-switch

Related commands:

getncalvel, setnrmvel

Example:

2 1 2 setncalvel
0.1 2 2 setncalvel

With command 2 ncal, the controller is moving Axis-2 with 2 rev./s. to the cal limit-switch and with 0.1 rev./s out of the switch.

getncalvel

getncalvel	Corvus TT	Corvus eco	Corvus PCI

Valid with firmware version 4.0

Description:

The command ***getncalvel*** returns the *ncal* limit-switch move velocities.

Syntax:

[axis] getncalvel

Reply:

[velocity-1] cr
[velocity-2] cr

	Range	**Unit**
[velocity-1]	0 - 45	rev. /s
[velocity-2]	0 - 45	rev. /s

Example:

2 getncalvel

Reply:

2.000000
0.250000

setrmvel	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***setrmvel*** defines the two velocities for the rm limit-switch move. The setting is significant for all axes.

1. Velocity: move in positive direction into the limit-switch
2. Velocity: move in negative direction out of the limit-switch

For the reason of compatibility with the older controllers, the unit of this velocity is defined in revolutions/s.
The resulting velocity (mm/s) depends on the pitch value of the virtual 0-Axis (see command ***setpitch***).

Syntax:

[velocity] [index] ***setcalvel***

[index]	Description
1	Velocity to the limit-switch
2	Velocity out of the limit-switch

	Range	Unit
[velocity-1]	0 - 45	rev./s
[velocity-2]	0 - 45	rev./s

Example:

2 0 setpitch (virtual 0-Axis)
2 1 setrmvel
1 2 setrmvel
The pitch of the 0-Axis is adjusted to 2mm
The controller is moving with 2 rev./s (4 mm/s) to the rm limit-switch and with 1 rev./s (2 mm/s) out of the rm limit-switch.

getrmvel

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getrmvel*** returns the two adjusted ***rm*** move velocities.

Syntax:

getrmvel

Reply:

[velocity-1]
[velocity-2]

	Range	Unit
[velocity-1]	0 - 45	rev. /s.
[velocity-2]	0 - 45	rev. /s.

Example:

getrmvel

Reply:

2.000000
0.250000

set<u>n</u>rmvel

Valid from firmware version 4.0

Description:

Command ***setnrmvel*** configures the two velocities for the ***nrm*** limit switch move.

1. velocity: move in positive direction into the limit-switch
2. velocity: move in positive direction out of the limit-switch

For the reason of compatibility with the older controllers, the unit of this velocity is defined in revolutions/s.
The resulting velocity in mm/s depends on the pitch value of the virtual 0-Axis (see command ***setpitch***).

Syntax:

[velocity] [index] [axis] ***setnrmvel***

	Range	Unit
[Velocity]	0 - 45	revolution/s
[index]	1, 2	-
[axis]	1, 2, 3,	-

[index]	Description
1	Velocity to the limit-switch
2	Velocity out of the limit-switch

Example:

2 1 1 setnrmvel
1 2 1 setnrmvel

The pitch of 0-Axis is adjusted to 2mm.
The controller is moving Axis-1 with 2 rev./s to the rm limit-switch and with 1 rev./s out of the switch.

get<u>n</u>rmvel

Valid from firmware version 4.0

Description:

Command ***getnrmvel*** returns the two adjusted ***nrm*** movement velocities.

Syntax:

[axis] getnrmvel

Reply:

[velocity-1]
[velocity-2]

	Range	Unit
[velocity-1]	0 - 45	rev. /s.
[velocity-2]	0 - 45	rev. /s.

Example:

2 getnrmvel

Reply:

2.000000
0.250000

setrefvel

Corvus TT | Corvus eco | Corvus PCI

Description:

Command ***setrefvel*** defines the velocity with which the move to a reference mark is executed.
See command ***refmove***.

To find the reference mark with a adequate accuracy, it is recommended to choose a slow refmove velocity.

Syntax:

[velocity] [index] ***setrefvel***

	Range	Unit
[velocity]	0 - 45	mm/s
[index]	1	fix value

Related commands:

getrefvel / setvel / refmove / setref

Example:

0.5 1 setrefvel

The refmove velocity is adjusted to 0.5mm/s.

getrefvel

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getrefvel*** returns the setting of *setrefvel*.

To be compatible to the older controllers the command ***getrefvel*** returns two parameters. The second parameter is irrelevant.

Syntax:

getrefvel

Reply:

[velocity]
[NF]

	Range	**Unit**
[velocity]	0 - 45	mm/s
[NF]	irrelevant	

Example:

getrefvel

Reply:

0.500000
0.010000

Positioning commands

move (m)

move (m)	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***move*** executes point to point positioning tasks to absolute coordinates based on the point of origin. The move profile is calculated in respect to the velocity/acceleration setup and the given hard or software limits.
The axes are linear interpolated, this causes the controller to start and stop all active axes simultaneously

Command ***status*** returns the actual state of the move procedure.

Ctrl-C or ***abort*** interrupts the actual move.

Syntax:

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

Absolute coordinates	Range	Unit
[Axis -1]	+/- 16383mm	axis unit
[Axis -2]	+/- 16383mm	axis unit
[Axis -3]	+/- 16383mm	axis unit

The number of expected coordinates depends on the setting of ***setdim***.

setdim	Expected number of coordinates
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

Related commands:

rmove, speed

Examples:

Dimension = 3
12.5 20.0 0.0001 m

Dimension = 1
12.5 m

Dimension = 2
12.5 20 m

rmove (r)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***rmove*** executes point to point positioning tasks relative to the current position.

The move profile is calculated in respect to the velocity/ acceleration setup and the given hard or software limits.
The axes are linear interpolated, this causes the controller to start and stop all active axes simultaneously

The command ***status*** returns the actual state of the move procedure.

Ctrl-C or ***abort*** interrupt the current executed move.

Syntax:

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

Relative coordinates	Range	Unit
[Axis-1]	+/- 16383mm	axis unit
[Axis-2]	+/- 16383mm	axis unit
[Axis-3]	+/- 16383mm	axis unit

The number of expected coordinates depends on the setting of command ***setdim***.

setdim	Expected number of relative axis coordinates
1 setdim	Axis-1
2 setdim	Axis-1_Axis-2
3 setdim	Axis-1_Axis-2_Axis-3

Related commands:

move, speed

Examples:

Dimension = 3
0.5 20 0.0001 r

Dimension = 1
12.5 rmove

Dimension = 2
12.5 20 r

speed

	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***speed*** starts a constant velocity move.

The move profile is calculated in respect to the velocity/ acceleration setup and the given hard or software limits.

Speed and direction can be changed on the fly.

With command ***stopspeed*** or ***Ctrl-C*** the move of all axes is stopped.
The speed mode is indicated in the status reply, see command ***status***.

Syntax:

[direction] [velocity] [axis] ***speed***

	Range	Unit
[direction]	+ / -	
[velocity]	0-60 *	rev./s.
[axis]	1, 2, 3	

* depends on the model and the released speed grade.

Example:

10 1 speed
Axis-1 is continuos moved with 10 rev./s in positive direction.
-0.1 2 speed
Axis-2 is continuos moved with 0.1rev./s in negative direction.
-0.5 2 speed
The speed of Axis-2 is changed on the fly to 0.5 rev./s.

stopspeed

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***stopspeed*** interrupts the constant velocity move of all axes with the adjusted acceleration.
See command ***sa***.

Syntax:

stopspeed

Example:

stopspeed

# test	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***test*** preforms a positioning test procedure.
This procedure start at the origin and moves stepwise to the maximum range, than stepwise back to the origin.

The procedure is aborted if the communication interface receives any ASCII character.

The command is only functional, if the limits of all axes are defined.

Syntax:

[Step size] [Axis] ***test***

	Range	**Unit**
Step size	any	axis units
Axis	1, 2, 3	

Example:

cal
rm
10 1 test

unit = mm / The limits are determined!

Axis-1 moves in 10mm steps to the upper limit.
If the limit border is reached, the axis moves in 10mm steps backwards to the origin.
This procedure is executed until a ASCII sign is received.

randmove

randmove	Corvus TT	Corvus eco	Corvus PCI

Description

Command ***randmove*** moves all active axes to randomized coordinates with a randomized velocity/acceleration setup.

The randmove procedure is terminated if the controller receives any ASCII character.

The command is only functional if the limits of all axes are defined.

Syntax:

randmove

Example:

cal
rm
randmove

The limits are defined with *cal* and *rm*.
All active axes are moving to randomized positions within the limits.

Limit Switch functions

calibrate (cal)

calibrate (cal)	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***cal*** executes the limit-switch move to the cal limit-switches. All active axes are simultaneously moved in negative direction, until the cal-switches are in ON state.
After that, the controller moves each axis in positive direction, as long as the limit-switches switch in OFF state.
At the OFF state coordinate, the position of the axes are set to zero (depends on the setup *setaxis*). Further on it is not possible to move the axes to coordinates less than zero.

With ***Ctrl-C*** the cal limit-switch move is immediately aborted and the origin and lower limit is set at this coordinate.

The origin and the lower limits are not permanently stored. After power OFF these values are lost.

During the ***cal*** procedure the command interpreter is blocked and no other commands can be executed.
The received commands are temporary stored in the communication FIFO and executed after the cal procedure is finished.

The proceedings to generate an automatic status reply after the completion of the limit-switch move, is described in the Venus-1 introduction.

Syntax:

calibrate or ***cal***

Example:

cal

rangemeasure (rm)

Corvus T, Corvus eco, Corvus PCI

Description:

The command ***rm*** executes the limit-switch move to the rm limit-switches. All active axes are simultaneously moved in positive direction, until the rm-switches are in ON state. After that, the controller moves each axis in negative direction, as long as the limit-switches switch in OFF state. At the OFF state coordinate, the position of the axes are set to zero (depends on the setup *setaxis*). Further on it is not possible to move the axes behind these coordinates

With ***Ctrl-C*** the rm limit-switch move is immediately aborted and the origin and upper limit is set at this coordinate.

The origin and the upper limits are not permanently stored. After power OFF these values are lost.

During the ***rm*** procedure the command interpreter is blocked and no other commands can be executed. Nevertheless the received commands are temporary stored in the communication FIFO and executed after the cal procedure is finished.

The proceedings to generate an automatic status reply after completion of the limit switch movement, is described in the Venus-1 introduction.

Syntax:

rangemeasure or ***rm***

Example:

rm

getcaldone

Corvus TT | Corvus eco | Corvus PCI

Valid from firmware version 4.42

Description:

With command ***getcaldone*** it can be determined if the calibration moves to the limit-switch *cal* and *rm* are executed or not.

The status of the *rm*-limit switch move is cleared if an *cal*-limit switch move is executed.

Syntax:

[axis] getcaldone

Reply:

[decimal value]

	Range
[value]	0 - 3

[value]	***cal* done**	***rm* done**
0	no	no
1	yes	no
2	no	yes
3	yes	yes

Example:

1 getcaldone

Reply:
1

setsw

setsw	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***setsw*** adapts the specified limit-switch input to the connected switch type of the cal/rm-switch.

Following settings are possible:

- normally open (no)
- normally closed (nc)
- limit-switch disabled

If a limit-switch input is disabled, it can not accomplish a safety function.

Syntax:

[function] [limit-switch] [axis] ***setsw***

[function]	NPN Type *	PNP Type **
0	closer (nc)	opener (no)
1	opener (no)	closer (nc)
2	disabled	disabled

* NPN-Switch is switched to GND

** PNP-Switch is switched to VCC

[limit-switch]	description
0	cal-input
1	rm-input

Examples:

0 0 1 setsw

cal limit-switch input of Axis-1 is prepared for a NPN-closed or PNP-open switch.

2 1 2 setsw

rm limit-switch input of Axis-2 is disabled.

getsw

Description:

The command ***getsw*** returns the setting of the limit-switch inputs.

Syntax:

[axis] ***getsw***

	Range
[axis]	-1, 1, 2, 3

Reply:

[cal-input] [rm-input]

[Input]	NPN-Switch *	PNP-Switch **
0	closer (nc)	opener (no)
1	opener (no)	closer (nc)
2	disabled	disabled

Example:

3 getsw

Reply:

0 0

-1 getsw

0 0 1 0 2 2

Axis-1

getswst

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getswst*** returns the current activity of the limit-switch inputs.

Syntax:

[axis] ***getswst***

	Range
[axis]	-1, 1, 2, 3

Reply:

[cal-input]] [rm-input]

[cal-input]	Description
0	cal limit-switch is in OFF state
1	cal limit-switch is in ON state

[rm-input]	Description
0	rm limit-switch is in OFF state
1	rm limit-switch is in ON state

Example:

3 getswst
Reply: 0 0
The cal and rm limit-switches of Axis-3 are in OFF state.

-1 getswst
Reply: 00 10 00
The cal limit-switch of Axis-2 is in ON state, others are OFF.

setcalswdist

Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setcalswdist*** an additional distance out of the limit-switches can be defined.
This setting reduces the working area on each side of an axis.

With this setting, the limit-switch move procedure works as follows:
1. Move to the limit-switch until it is On state
2. Move out of the limit-switch until it is released (Off state)
3. Move out a additional distance defined with ***setcalswdist***

For the reason of compatibility with older controllers, the unit of ***setcalswdist*** is defined in motor revolutions.
The resulting distance in mm depends on the spindle pitch setting of the 0-Axis.

Syntax:

[revolution] [axis] ***setcalswdist***

	Range	**Unit**
[revolution]	0 - >1000	Motor revolutions
[axis]	1, 2, 3	

Related command:

getcalswdist

Example:

5 1 setcalswdist
An additional distance of 5 motor revolutions out of the cal and rm limit-switches is performed for the limit-switch procedure of Axis-1.

# *getcalswdist*	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getcalswdist*** returns the settings of *setcalswdist*.

Syntax:

[axis] ***getcalswdist***

	Range
[axis]	-1, 1, 2, 3

Reply:

[value]

	Unit
[value]	rev.

Example:

1 getcalswdist

Reply:

5.00000

-1 getcalswdist

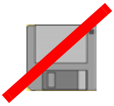
Reply:

5.00000
0.00000
0.00000

setlimit

Corvus TT	Corvus eco	Corvus PCI

Description:



With command ***setlimit*** the software limits are defined for the axes.
If the limits are determined, the controller is not able to move beyond it. All moves are ramped down and stopped at the limit border.
This scenario will produce error Code 1004 (see command *ge*)

With command *setnlimit,* software limits can be defined for each axis separately.

In manual mode the limits are determined with pressing the limit-switches.

Requirements to define software limits:

- If the hard limits are specified with the cal/rm limit switch moves, the software limits must be located between these limits.

- The value of the lower limit has to be less than the value of the upper limit.

- The current position has to be within these limits, otherwise the command is not executed.

- If an axis is disabled, no limit inputs are accepted.

- The limit-switch move ***cal*** and ***rm*** overwrites the software limits.

- With command reset, all limits are lost and switched to their maximum value +/-16383 mm

Syntax:

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

The number of axis parameters depends on the setting of *setdim*.

	Description
[-A1]	lower limit Axis-1
[-A2]	lower limit Axis-2
[-A3]	lower limit Axis-3

	Description
[A1+]	upper limit Axis-1
[A2+]	upper limit Axis-2
[A3+]	upper limit Axis-3

	Range	Unit
[-A1] [-A2] [-A3]	-16383mm	user unit
[A1+] [A2+] [A3+]	+16383mm	user unit

Related command:

setnlimit, getlimit, getnlimit

Example:

getdim = 3

0 0 0 12 25 30 setlimit

The axis limits are defined as follows:

Axis-1: 0 until 12
Axis-2: 0 until 25
Axis-3: 0 until 30

getlimit

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getlimit*** returns the limit coordinates of all axes. Dependent on the setting of *setdim* the controller returns the limit coordinates of 1, 2 or 3 axes.
With command *setnlimit* the limit of a single axis can be replied.
If no limit is specified the maximum limit value (16383mm) is returned:
I

Syntax:

getlimit

Reply:

		Unit	**Axis**
[lower limit]	[upper limit]	unit	1
[lower limit]	[upper limit]	unit	2
[lower limit]	[upper limit]	unit	3

* depends on the setting of *setdim*

Example:

getdim = 3
getlimit

0.000000 7.723750 [cr]
0.000000 7.723750 [cr]
-16383.000000 16383.000000 [cr}

ncal	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.0

Description:

The command ***ncal*** executes a single axis limit-switch move to the cal limit-switch. This procedure determines the origin and lower limit of the selected axis. The move procedure is similar to the function ***cal***.

Contrary to the cal limit switch movement the Venus-1 command interpreter is not blocked with *ncal*.
As a result the controller can execute all command requests during the *ncal* procedure.

The velocity of *ncal* movement is adjusted with the command *ncalvel*

With ***Ctrl-C*** the ncal limit-switch move is immediately aborted and the origin and lower limit is set at this coordinate.

Syntax:

[axis] ncal

Example:

1 ncal

Axis-1 is moved to cal limit-switch.

nrm

	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.0

Description:

The command ***nrm*** executes a single axis limit-switch movement to the rm limit-switch. This procedure determines the upper limit of the selected axis.

Contrary to the rm limit switch movement the Venus-1 command interpreter is not blocked with *nrm*.
As a result the controller can execute all command requests during the *nrm* procedure.

The other limit-switch movement functionality remain the same. See the *rm* command description.

The velocity of ***nrm*** movement is adjusted with the command *nrmvel*

With ***Ctrl-C*** the nrm limit-switch move is immediately aborted and the origin and lower limit is set at this coordinate.

Syntax:

[axis] nrm

Example:

1 nrm
Axis-1 is moved to rm limit-switch.

getnlimit

	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.5.0

Description:

The command ***getnlimit*** returns the current limits of a specified axes.
If no limits are specified, the following values are returned:
-16383.000000   16383.000000

With command *getlimit* the limits of all axes are replied.

Syntax:

[axis] ***getnlimit***

unit= user defined units

Reply:

[lower limit] [upper limit]

Related commands

getlimit

Example:

1 getnlimit

0.000000   12.00000

org

org	Corvus TT	-	-

Valid from firmware version 4.1.0

Description:

Command ***org*** a moves the specified axis a relative stroke until the org-switch is in ON state.
Similar to the cal/rm limit switch procedure the controller than moves in the reverse direction, as long as the org-switch is switched to the OFF state.

The function must be enabled with command *setorg.*
The speed values for the org moves are fixed with command *setcalvel.*

If the switch is not in ON state within the specified stroke, the move is stopped and error code 1011 is generated.

With command ***Ctrl-C*** the org procedure is aborted.

If the org switch is already in ON state when the command ***org*** is executed, the controller moves the axis immediately in the reverse direction, until the org switch is in OFF state.
In this case also error code 1011 is generated.

Syntax:

[direction] [relative stroke] [axis] ***org***

	range	unit
[direction]	+ / -	
[relative stroke]		unit
[axis]	1, 2, 3	

Example:

-10 1 org
Axis-1 is moved 10 [units] in negative direction until the org-switch is in ON state. After this, the axis is moved in positive direction until the org-switch is in OFF state.

# ***setorg***	Corvus TT	-	-

Valid from firmware version 4.1.0

Description:

The command ***setorg*** enables or disables the org-switch input.

Syntax:

[on/off] [axis] ***setorg***

	Range
[on/off]	0,1
[axis]	1, 2, 3

[on/off]	**Function**
0	org-input disabled
1	org-input enabled

Example:

1 1 setorg

Enables org-input for Axis-1.

# *getorg*	Corvus TT	-	-

Description:

Command ***getorg*** returns the org-input settings of the specified axis.

Syntax:

[axis] ***getorg***

	Range
[axis]	1, 2, 3

Reply:

[index]

[index]	Function
0	org-input disabled
1	org-input enabled

Example:

1 getorg

setorgsw	Corvus TT	-	-

Valid from firmware version 4.1.0

Description:

The command ***setorgsw*** adapts the specified org-switch input to the connected switch type.

Following settings are possible:

- normally open (no)
- normally closed (nc)

The input can be disabled with command *setorg*.

Syntax:

[function] [axis] ***setorgsw***

[function]	NPN-Switch *	PNP-Switch **
0	closed (nc)	open (no)
1	open (no)	closed (nc)

* NPN-switch is switched to GND
** PNP-switch is switched to VCC

Examples:

0 1 setorgsw
org limit-switch input of Axis-1 is prepared for a NPN-closed or PNP-open switch.

getorgsw

getorgsw	Corvus TT	-	-

Description:

The command ***getorgsw*** returns the setting of the org-switch inputs.

Syntax:

[axis] ***getorgsw***

	Range
[axis]	-1, 1, 2, 3

Reply:

[function]

[function]	NPN-Switch *	PNP-Switch **
0	closed (nc)	open (no)
1	open (no)	closed (nc)

Example:

3 getsw

3 getsw

Reply:
0 0

-1 getsw

Reply:
0 0 0 1 2 2

[Axis-1] [Axis-2] [Axis-3]

getorgswst

getorgswst	Corvus TT	-	-

Valid from firmware version 4.1.0

Description:

The command ***getorgswst*** returns the current activity of the org limit-switch input.

Syntax:

[axis] ***getorgswst***

	Range
[axis]	-1, 1, 2, 3

Reply:

[switch state]

[switch state]	Description
0	org-switch is in OFF state
1	org-switch is in ON state

Example:

1 getorgswst
Reply: 0
The org-switch of Axis-1 is in OFF state.

-1 getorgswst
Reply: 0 1 0
The org-switch of Axis-2 is in ON state, others are OFF.

Safety functions

Ctrl-C

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***Ctrl-C*** interrupts the current executed command. Moves will be stopped immediately with the acceleration setup, defined with command *sa.*
Subsequently the already transfered commands in the RS-232 input buffer will be executed

If a *cal* or *rm* limit-switch move is interrupted with ***Ctrl-C*** the current axes positions are taken over as the controller lower limits or maximum range limit.

Due to the *Ctrl-C command* has not to pass the command FIFO, it can not be delayed with a blocking command.

It is not permissible to use Ctrl-C in a quick succession.

Syntax:

ASCII sign	Decimal value	Hex value
Ctrl-C	3	0x3

Related command:

abort, Ctrl-B, Ctrl-D

Example:

Ctrl-C

Ctrl-B

	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***Ctrl+B*** the motor current of all axes is turned off.
The communication with the controller is further on possible.
To continue the normal motor driver operation, the controller has to be restarted with power ON/OFF or with the command *reset*.

Syntax:

Ctrl-B equal to decimal 4

Related commands:

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

Example:

Ctrl-B
The motor drivers are disabled.

abort

abort	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***abort*** interrupts the current executed command. All moves will be stopped immediately with the acceleration setup, defined with command *sa.*
The commands in the command FIFO are not cleared.

Due to ***abort*** has to pass the data input FIFO, it can be delayed with a blocking command.

For example: ***abort*** is blocked with the command ***ge***

1. ***100 0 0 move (current executed)***
2. ***ge***
3. ***abort***

Syntax:

abort

Related command:

Ctrl-C, Ctrl-B, Ctrl-D

Example:

abort

setinfunc

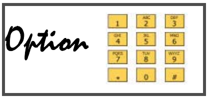
setinfunc	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setinfunc*** a safety function via the Digital Input/Output interface can be established.
This function can interrupt the current move or/and limiting the move direction of an axis.
The function is valid for programmed and manual move.

The following move limitations can be configured:

- move is limited for all directions
- move is limited for the negative direction
- move is limited for the positive direction



To support the function *setinfunc*, the controller must be equipped with the feature Digital Input/Output.
Details, see in the hardware manual

setinfunc procedure description

If a disable signal gets active at one of the configured inputs, while a move is executed, the move of all axes is stopped with the acceleration setup ***sa***.

After that, each axis follows the individual setting with command ***setinfunc***.
Axes which are not configured with a ***setinfunc*** function**,** can further on normally moved.
The *setinfunc* status is reflected in the status Bit D6, additionally the assigned Axis-LED in the Corvus diagnostic display is flashing.

If the disable signal is removed, all ***setinfunc*** limitations are canceled and the axes are free.

Syntax:

[action] [input] [axis] ***setinfunc***

	Range
[action]	0, 1, 2, 3
[input]	1, 2, 3
[axis]	1, 2, 3

	Description
[input]	digital inputs 1, 2, 3
[action]	setinfunc limitation

[action]	Function
0	no limitation
1	moving is limited for the positive direction
2	moving is limited for the negative direction.
3	moving is limited for all directions

Related command:

getinfunc

Example:

1 1 3 setinfunc
The following safety function is configured to Axis-3:

If Din-1 is active, all axes are stopped, Axis-3 is limited
in negative direction but can be moved in positive direction.
2 2 3 setinfunc
If Din-2 is active, all axes are stopped, Axis-3 is limited
in positive direction but can be moved in negative direction.
After the axes are stopped the Axis-1 and Axis-2 can be moved normally.

getinfunc

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getinfunc*** returns the setting of ***setinfunc***.

Syntax:

[input] [axis] ***getinfunc***

	Range
[input]	1, 2, 3
[axis]	1, 2, 3

Reply:

[action]

	Range
[action]	0, 1, 2, 3

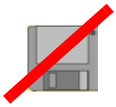
[action]	Function
0	no limitation
1	move is limited for the positive direction
2	move is limited for the negative direction.
3	move is limited for all directions

Example:

1 3 getinfunc

setmp	Corvus TT	Corvus eco	Corvus PCI

Description:



With command ***setmp*** the motor current from a specified axis can be switched off completely. All other functions remain active.

Depending on the rotor position, the motor will jerk if the motor current is switched on or off.

Syntax:

[switch] [axis] ***setmp***

	Range
[switch]	0, 1
[axis]	1, 2, 3

[switch]	**Function**
0	Motor driver is current less
1	Motor driver in standard condition

Example:

0 1 setmp

The motor current of Axis-1 is switched off.

getmp

	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getmp*** returns the setting of *setmp*.

Syntax:

[axis] ***getmp***

parameter	Range
[axis]	-1, 1, 2, 3

Reply:

[switch status]

	Range
[switch status]	0, 1

Example:

1 setmp
Reply:
0

-1 getmp
Reply:
0 1 1

position / origin / coordinate system

pos (p)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***pos*** return the current coordinate of all active axes.
The position value relates to the origin which is defined with command *cal* or *setpos*. The number of replied coordinates depends on the setting of *setdim*.

Syntax:

pos or ***p***

Reply:

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

	Description	Unit
[Pos Axis-1]	Position of Axis-1	unit
[Pos Axis-2]	Position of Axis-2	unit
[Pos Axis-3]	Position of Axis-3	unit

.

Example:

getdim = 2

pos

Reply:

1.00000 19.00000

setpdisplay

Description:

Command ***setpdisply*** the display format of the replied position value can be defined.
The positioning resolution is not affected by this setting.

Syntax:

[VK] [NK] [Axis] ***setpdisplay***

	Function	Range
[VK]	digits before the decimal point	
[NK]	digits after the decimal point	
[Axis]		1, 2, 3

Related command:

getpdisplay

Example:

1 3 1 setpdisplay
2 10 2 setpdisplay

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

getpdisplay

Description:

Command ***getpdisplay*** returns the setting of *setpdisplay*.

Syntax:

[Axis] ***getpdisplay***

	Range
[Axis]	-1, 1, 2, 3

Reply:

Host mode:
[VK] [NK]

Terminal mode:

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

Example:

1 getpdisplay

setpos

	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setpos*** the point of origin of all axes can be defined. The coordinates of the limits will be recalculated if the point origin changes.

The axes must be enabled.

For special cases the zero point can be defined with a relative offset.

Syntax:

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

	Range	unit
[Axis-1]	+/-16383mm	unit
[Axis-2]	+/-16383mm	unit
[Axis-2]	+/-16383mm	unit

Example:

0 0 0 setpos

The current coordinate is defined as the point of origin.

10 10 10 setpos / unit = mm

The current coordinate is defined as the point origin with an relative offset 10 mm each axis.
The command ***pos*** will reply the position value -10 -10 -10 if the previous coordinate was 0 0 0.

align

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***align*** rotates the orthogonal coordinate system of Axis-1 and Axis-2 (X/Y) around it‘s origin. Axis-3 is not affected by the rotation.
After rotation, the positioning commands *move* and *rmove* are using the new coordinate system.
The limits are further on checked. It is not possible to move to coordinates out of the limits.
Command *getlimit* or *getnlimit* returns the limit values.
With command *ico* the coordinate system is restored.

To execute the function properly the controller dimension setting *setdim* must be set to 2.

Syntax:

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

With [0] [0] [OrgX] [OrgY] the location of the related axis is defined.
[X/Y] defines if the related axis is X or Y.

	Description
[OrgX]	X-coordinate of related axis
[OrgY]	Y-coordinate of related axis
[X/Y]	Related axis 1 = X-axis, 2 = Y-axis

	Range	Unit
[OrgX]	+/- 16383mm	user unit
[OrgY]	+/- 16383mm	user unit
[X/Y]	1, 2	

Reply:

Command ***getico*** returns value=0 if the original coordinate system is rotated.

.

Example:

0 0 10 10 1 align

The X-axis is aligned to the coordinate 0 / 0 |10 / 10

The Y-coordinate follows automatically.

The sketch below shows a sample found on a microscope scanning stage. The sample is misaligned to the stage. With command align it is possible to adapt the controller coordinate system to the coordinates of the sample.

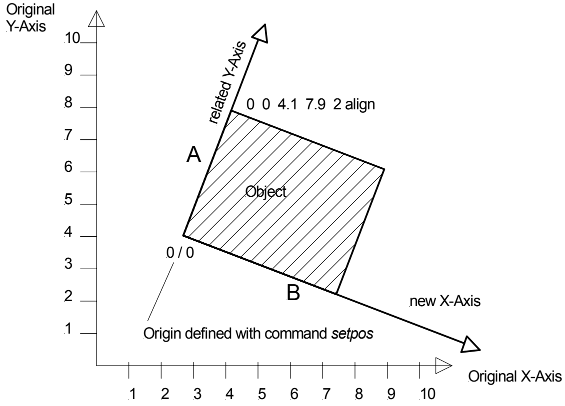
Border A of the object is defined with the coordinates 0 / 0 | 4.1 / 7.9

With command 0_0_ 4.1_ 7.9_ 2 ***align*** this border is aligned to the new Y-Axis. Therefore the coordinate system of the controller is rotated anti clockwise around its origin.

The coordinate system of the controller is than aligned to the coordinates of the object.

Alternatively the align procedure can be related to the coordinates of border B.

This will give the same results a the previous method.



ico

	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***ico*** restores the original coordinate system of the controller.

Syntax:

ico

Related command:

getico, align

Example:

ico

getico

	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getico*** verifies if the coordinate system is rotated with command ***align***.

Syntax:

getico

Reply:

[Index]

[Index]	Function
0	coordinate system is rotated
1	coordinate system is not rotated

	Range
[Index]	0, 1

Example:

getico

Status requests

status (st)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***status*** returns the current state of the controller.

Each state is assigned to a binary digit from D0 to D8.
If more states are active, the decimal values of the digits are added.
To decode the replied status, it is necessary to convert the decimal value into a binary pattern and mask the bits.

Binary	Decimal	Function
D0	1	Status current command execution
D1	2	Status Joystick or Handwheel
D2	4	Status Button A
D3	8	Machine error
D4	16	Status speed mode
D5	32	Status In-Window
D6	64	Status setinfunc
D7	128	Status motor enable, safety device
D8	256	Joystick button

D0: Current command execution or motion state

0	move finished
1	move in progress

D1: Joystick state

0	Manual mode not active
1	Manual mode is active

D2: Switch A state (only Corvus TT)

0	Button A not pressed
1	Button A pressed

D3: Machine error state

0	No machine error
1	Machine error occurred

D4: Speed mode state

0	Speed mode active.
1	Speed mode not active

D5: Closed Loop Window state

0	Current position out of the target window.
1	Current position within the target window.

D6: *setinfunc* safety function state

0	setinfunc limitation not active
1	setinfunc limitation active

D7: Motor disable state (disabled from safety device)

0	Motor driver enabled
1	Motor driver disabled from external device

D8: Joystick button state

0	Joystick button released
1	Joystick button pressed

Syntax:

status or ***st***

Reply:

[bit coded decimal value]

Example:

status
Reply:
2
Bit pattern: 01000000

Decoded states:
Move is finished, manual mode active

More examples on the following page.

Command execution

Manual mode

Button A

Speed mode

in-window

setinfunc mode

Motor driver status
Overcurrent
or motor disabled
from external device

1	2	4	8	16	32	64	128	
D0	D1	D2	D3	D4	D5	D6	D7	
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

status reply

geterror (ge)

Corvus TT	Corvus eco	Corvus PCI

Description:

With the command ***geterror*** the last occurred system error is returned. Afterwards the error code memory is cleared.

The occurrence of an system error is not reflected in the status reply.

Syntax:

geterror

Reply:

[Error code]

Error codes	Description
1....4	Internal error
1001	Wrong parameter
1002	Not enough parameter on the stack
1003 1007	Range of parameter is exceeded
1004	Move stopped working range should run over
1008	Not enough parameter on the stack (same as 1003)
1009	Not enough space on the stack
1010	Not enough space on parameter memory
1015	Parameters outside the working range
2000	Unknown command

Example:

ge

getmerror (gme)

Corvus T, Corvus eco, Corvus PCI

Description:

With the command ***getmerror*** the hardware errors from the machine error stack are returned.

The machine errors are put on a memory stack which can store a maximum of 10 error codes.
The occurrence of a machine error is reflected in the status reply and displayed in the Error-LED in the diagnostic panel.

With each ***gme*** command the error codes are replied in order of their appearance.
If all errors are returned, the error status is cleared and the Error-LED is deactivated after power up.

Syntax:

getmerror or ***gme***

Reply:

[machine error code]

Error code	Description
0	No machine errors occurred
1	Error memory is overflowed
10	Motor driver disabled (Motor disable function is active) or defective 12V power supply.
13	Maximum positioning error in Closed Loop mode is exceeded, see command ***setclpara*** (7th. parameter)
23	RS422 Encoder error

Example:

gme

# *gsp*	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***gsp*** returns the number of elements on the parameter stack.

Syntax:

gsp

Reply:

[Number]

	Range
[Number]	0 -99

Related command:

clear

Example:

gsp

Reply:
2

getticks (gt)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***gt*** (**g**et_**t**icks) returns the number of processor cycles, since the controllers was started.
Each count equals 250µs.
This function can be used as a time-stamp, to reference data or events.

After 298 hours this counter will overflow and start with zero.
.

Syntax:

gt

Example:

gt

returns: 10922835

(10922835 * 250µs = 2730,708 s)

Input / Output functions

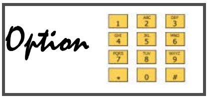
setout

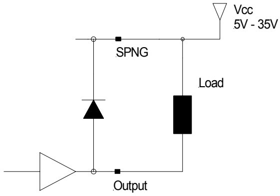
Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setout*** controls the digital outputs.
A similar function is command *outtrig*.

The outputs are open collector circuits. In ON state the output transistor is switched to DGND.





Syntax:

[value] ***setout***

Value	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

Related command:

getout

Example:

1 setout

getout

	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getout*** returns the state of the digital outputs as a decimal value.

Syntax:

getout

Reply:

[value]

	Range
[value]	0-7

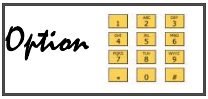
Example:

getout

# setaout	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setaout*** generates an analog output voltage between 0 and 1000mV with 8 Bit resolution.
Two outputs channels are provided.



A load resistance of 1 MOhm (or higher) is recommended for proper functionality. A smaller load resistance will reduce the output voltage.

Syntax:

[voltage] [channel] ***setaout***

	Range	Unit
[voltage]	0-1000	mV
[channel]	1,2	

Related command:

getaout

Example:

100 1 setaout
100mV output voltage is generated at channel-1.

getaout

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getaout*** returns the adjusted analog output voltage, generated with command ***setaout***.

Syntax:

[channel] ***getaout***

unit= [mV]

Reply:

[voltage]

	Range	Unit
[voltage]	0-1000	mV

Example:

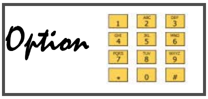
1 getaout

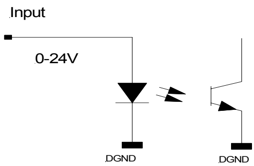
getin

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getin*** returns the current status of the three digital Inputs Din-1, Din-2, Din-3 as a decimal value.
Each input is assigned to a binary state from D0 to D2.
If more inputs are active, the decimal values of the states are added.
To get each input state separately, it is necessary to convert the decimal value into a binary and mask the bit pattern.





Syntax:

getin

Reply:

[decimal value]

	Range
[decimal value]	0-7

Example:

getin

[decimal value]	Input Din-1	Input Din-2	Input Din-3
0*	0	0	0
1**	1	0	0
2	0	1	0
4	0	0	1

*0: Input voltage 0-2V
**1: Input voltage 3-24V

Closed Loop commands

setnselpos

Corvus TT | Corvus eco | Corvus PCI

Description:

Command ***setnselpos*** determines whether the internal calculated position value or the actual position value, from a measurement system, is returned.

To display the actual position, the controller must be equipped with a digital or analog encoder interface.
See in manual chapter "functions".

Syntax:

[index] [axis] ***setnselpos***

	Range
Index	0,1
Axis	1, 2, 3

[index]	Description
0	returns the calculated position
1	returns measured position

Related command:

getnselpos

Example:

0 3 setnselpos
1 1 setnselpos

getnselpos

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getnselpos*** returns the settings of *setnselpos*.

Syntax:

[axis] ***getnselpos***

	Range
[axis]	-1, 1, 2, 3

Reply:

[index]

	Range
[index]	0, 1

Example:

3 getnselpos

setclpara

setclpara	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setclpara*** configurates the loop controller.

Following settings are possible:

- Adjustment of the proportional gain (P), integral gain (I) and derivative gain (D)
- Safety function
  Disables the axis if the position deviation of the actual position and a desired position exceeds a determined value.
- Limiting the adjustment speed caused from the I-Controller.
- Limiting the influence of the I-Controller (anti windup).

Syntax:

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

The commands exists of a maximum of 10 parameters separated with a space and followed with [np] the number of parameters.

If a single parameter should be changed all prefixed parameters must also be transmitted.

Valid from firmware version 4.50 it is also possible to transmit single closed loop parameters, see command *sp*.

Parameter	Description
[P]	Proportional gain (Only used to adjust linear motors)
[I]	Integral gain Important by the use of stepper motors
[D]	Derivative gain (Only used to adjust linear motors)
[16383]	This setting may not be changed.
[SP1]	Boost-factor (Only used to adjust linear motors)
[SP2]	Disables the axis if the load angle deviation exceeds a determined value. Command ***gme*** returns error code 13. With value = 0 the function is disabled. (Only used to adjust linear motors)
[dpos]	Disables the axis if the position deviation of the actual position and a desired position exceeds a determined value. Command ***gme*** returns error code 13. With value dpos = 0, the function is disabled.
[ivel]	This function limits the adjustment velocity caused from the I-Controller during moving and stand still. With value ivel = 0, the function is disabled.
[cutoff]	The cutoff parameter influences the smoothness of the motors in closed loop mode and optimizes the steady state error and settling time. With cutoff the I-value is dynamically reduced depending on the current velocity. The move is starting with a maximal I-value and is reduced dynamically to a minimum value at the adjusted cutoff velocity. Below the cutoff velocity the I-value has it´s maximum again. cutoff = 0 disables this function.
[SP3]	no function
[np]	The number of parameters that are send with the command.
[axis]	Axis index

parameter	range	default setting	unit
P		0	-
I		10	1/s
D		0	s
16383		16383 (maximal value)	
SP1		0	
SP2		0	
dpos		0 (0 = disabled)	mm
ivel		2 (0 = disabled)	mm/s
cutoff		2 (0 = disabled)	mm/s
SP3	0	0	
Axis	1, 2, 3		

Related commands:

getclpara, sp

Examples:

0_20_2_1_ setclpara

The parameters of "P" and "I" will configured for Axis-1, all following controller loop parameters remain unchained.

0_15_0_16383_0_0_1_2_2_9_3_setclpara

In summary nine controller loop parameters are configured for Axis-3.

# ***getclpara***	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getclpara*** return the settings of the loop controller.

10 parameters are replied.

Syntax:

[axis] ***getclpara***

	Range
[axis]	-1, 1, 2, 3

Reply:

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

Example:

1 getclpara

Reply:

0.000000 10.000000 0.000000 16383.000000 0.000000 0.000000 0.000000 2.000000 1.000000 0.000000

setsp

setsp	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.50

Description:

Command ***setsp*** defines the settings of the loop controller. Unlike the command setclpara, this command allows to change each parameter separately.

- Adjustment of the proportional gain (P), integral gain (I) and derivative gain (D)
- Safety function
  Disables the axis if the position deviation of the actual position and a desired position exceeds a determined value.
- Limiting the adjustment speed caused from the I-Controller.
- Limiting the influence of the I-Controller (anti windup)

Syntax:

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

SP	Para	Description
1	[P]	Proportional gain (Only used to adjust linear motors)
2	[I]	Integral gain Important by the use of stepper motors
3	[D]	Derivative gain (Only used to adjust linear motors)
4	[16383]	This setting may not be changed.
5	[SP1]	Boost-factor (Only used to adjust linear motors)
6	[SP2]	Disables the axis if the load angle deviation exceeds a determined value. Command ***gme*** returns error code 13. With value = 0 the function is disabled. (Only used to adjust linear motors)
7	[dpos]	Disables the axis if the position deviation of the actual position and a desired position exceeds a determined value. Command ***gme*** returns error code 13. With value dpos = 0, the function is disabled.
8	[ivel]	This function limits the adjustment velocity caused from the I-Controller during moving and stand still. With value ivel = 0, the function is disabled.
9	[cutoff]	The cutoff parameter influences the smoothness of the motors in closed loop mode and optimizes the steady state error and settling time. With cutoff the I-value is dynamically reduced depending on the current velocity. The move is starting with a maximal I-value and is reduced dynamically to a minimum value at the adjusted cutoff velocity. Below the cutoff velocity the I-value has it´s maximum again. cutoff = 0 disables this function.
10	[SP3]	no function

parameter	range	default	unit
SP1 (P)		0	-
SP2 (I)I		10	1/s
SP3 (D)		0	s
SP4 (16383)		16383 (maximum value)	
SP5		0	
SP6		0	
SP7 (dpos)		0 0 = disabled	mm
SP8 (ivel)		2 0 = disabled	mm/s
SP9 (cutoff)		2 0 = disabled	mm/s
SP10	0	0	
axis	1, 2, 3		

Related commands:

getsp, setclpara

Example:

100 2 setsp

The integral gain of Axis-1 is adjusted to 100 1/s.

getsp

	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getsp*** returns the selected servo parameter (sp) of an axis.

Syntax:

[SP Index] [axis] ***getsp***

	range
[axis]	1, 2, 3
[SP index]	1 - 10

Reply:

[**SP**]

Example:

1 getsp

Reply:
100

setscaleinterface

Corvus T, Corvus eco, Corvus PCI

Description:

Command ***setscaleinterface*** configures one of the both Closed Loop interfaces.

Corvus is equipped with an On-Board digital encoder interface for RS-422 input signals.
Additionally a separate hardware (sin/cos Module) is provided to support analog measuring systems with 1Vpp or MR signals.

This setting is active if the commands ***save*** and ***reset*** are executed.

Syntax:

[index] [axis] ***setscaleinterface***

	Range
[index]	0, 1, 2
[axis]	1, 2, 3

[index]	Description
0	Quadrature Interface disabled
1	Quadrature Interface enabled
2	Analog Interface (sin/cos Module) enabled

Example:

2 1 setscaleinterface
Initializes the analog encoder interface for Axis-1

getscaleinterface

Corvus T, Corvus eco, Corvus PCI

Description:

Command ***getscaleinterface*** verifies the type of the enabled encoder interface.

Syntax

[axis] ***getscaleinterface***

	Range
[axis]	1, 2, 3

Reply:

[index]

	Range
[index]	0, 1, 2

Example:

1 getscaleinterface

Reply:
1

setscaletype

setscaletype	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setscaletype*** adapts the encoder interface to the type of measurement system.
A distinction is drawn between linear encoders, rotational encoders with analog output an rotational encoders with digital output.

To adjust the encoder interface to the resolution of digital encoders we recommend to use command ***setclfactor***.
For linear encoders and analog rotational encoders the command ***setclperiod*** should be used.

The settings of ***setscaletype*** and ***setscaleinterface*** are executed from the factory if a controller is delivered with the option "closed loop".
.

Syntax:

[index] [axis] ***setscaletype***

	Range
[index]	0, 1
[axis]	1, 2, 3

[index]	Description
0	linear measurement system (analog/digital) analog rotary encoder
1	digital rotary encoder

Example:

1 1 setscaletype
Adapts the encoder interface for Axis-1 to a digital rotary encoder.

getscaletype

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getscaletype*** verifies the type of measurement system that is configured for the encoder interface.

Syntax

[axis] ***getscaletype***

	Range
[axis]	1, 2, 3

Reply:

[index]

	Range
[index]	0, 1

Example:

1 getscaletype

Reply:
1

setclfactor

setclfactor	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setclfactor*** adapts the digital encoder interface to the resolution of a digital rotational encoder. The value is equivalent to the number of pulses per revolution.

With the algebraic sign -/+ it is possible to adapt the count direction to the motor direction.

To support digital rotational encoders, the controller must be equipped with the Corvus digital encoder interface.
The interface has to be configured as follows.

Command	Value
setscaletype	0
setscaleinterface	1

Syntax:

[Count direction] [pulses/rev.] [axis] ***setclfactor***

Parameter	Range	Unit
+ / -		
pulses	1-50000	pulses/rev.
Axis	1, 2, 3	

Related command:

getclfactor, setscaletype, setscaleinterface

Example:

- 500 3 setclfactor

getclfactor

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getclfactor*** returns the setting of ***setclfactor***

Syntax:

[axis] ***getclfactor***

Parameter	Range
Axis	-1, 1, 2, 3

Reply:

[Count direction] [pulses / rev.]

Example:

1 getclfactor

Reply:

- 500

setclperiod

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setclperiod*** adapts the analog or digital encoder interface to the following encoder types:

- Linear encoders (RS-422, 1Vpp, MR)
- Rotational encoders (1Vpp)

The value for ***setclperiod*** is the resulting move distance within a one signal period of the encoder.

For digital rotational encoders the command *setclfactor* is recommended.

The transmission ratio caused from the spindle or gear has to be considered to calculate the ***setclperiod*** value (see examples).
With the algebraic sign -/+ it is possible to adapt the count direction to the motor direction.

Syntax:

[Counter direction] [Distance] [axis] ***setclperiod***

	Description
[Direction]	counting direction
[Distance]	distance within one signal period

	Range	**Unit**
[Direction]	+ / -	
[Distance]	0.0000001-1.999999	mm
[axis]	1, 2, 3	

Related command:

getclperiod, setscaleinterface, setscaletype

Examples:

- 0.002 3 setclperiod

Example: Linear stage with linear encoder

Specifications:
Linear stage with following specifications:
Encoder with signal period = 20µm, Spindle pitch = 10mm

The value for ***setclperiod*** is the resulting move distance within a one signal period of the encoder.
Therefore ***setclperiod = 0.020mm***

The value of ***setclperiod*** does **not** depend on the pitch. because the resulting move distance is the same as the signal period of the encoder.

Example: Linear stage with rotary encoder

Specifications:
Rotary encoder with 1Vpp output signal,
1000 signal periods / rev., mounted on the motor shaft.
Spindle pitch = 10mm

In this combination the movement distance within one signal period depends on the pitch.
The value for setclperiod will be calculated as follows:

setclperiod = 10 mm / 1000 periods = 0.01

getclperiod

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getclperiod*** returns the setting of *setclperiode*.

Syntax:

[axis] ***getclperiod***

Parameter	Range
[axis]	1, 2, 3

Reply:

[Value]

	Range	Unit
[Value]	0.0000001-1.999999	mm

Example:

1 getclperiod

Reply:

0.020

setclwindow

setclwindow	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setclwindow*** enables a +/- target window for the closed loop function. Within the target window the position control loop is not active. If the position is beyond the target window, the position control loop gets active.
Status Bit D5 is active if all axes are within the target window.

If the actual position of an axis is out of the target window, the assigned LED in the diagnostic panel is flashing.

The window function is disabled if the target window is set to 0.

Syntax:

[Window] [axis] ***setclwindow***

	Range	Unit	Function
[Window]		user	+/- window enabled
[Window]	0 *	user	window disabled
[axis]	1, 2, 3	-	

*factory setting

Related command:

getclwindow

Example:

0.001 1 setclwindow
(User unit = mm)
The target window of Axis-1 is adjusted to +/-1µm

getclwindow

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getclwindow*** returns the setting of the Closed Loop target window.

Syntax:

[axis] ***getclwindow***

	Range
[axis]	-1, 1,2,3

Reply:

+/- [value]

	Range	Unit
[value]		user

Example:

1 getclwindow

Reply:
0.001

-1 getclwindow

Reply:
0.001 0.002 0.000

setref

Corvus TT | Corvus eco | Corvus PCI

Description:

Command ***setref*** defines if the encoder reference mark is identified at the rising or falling edge.

Syntax:

[index] [axis] ***setref***

	Range
[index]	0, 1, 2
[axis]	1, 2, 3

[value]	Description
0	rising edge
1	falling edge
2	disabled

Related commands:

getref / getrefst, setrefvel

Example:

0 1 setref

The encoder interface identifies the reference mark at the rising edge of the reference signal.

getref

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getref*** returns the setting of *setref*.

Syntax:

[axis] ***getref***

	Range
[axis]	-1, 1, 2, 3

Reply:

[index]

	Range
[index]	0, 1, 2

Example:

1 getref

refmove

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***refmove*** moves all active axes to the reference mark of the measurement system.

The reference move velocity is adjusted with command *setrefvel*.
If no reference mark is found, the move stops until the specified distance is reached. The status of a reference move is returned with command *getrefst*.
With *Ctrl-C* the reference move will be aborted.

To move only a single axis to the reference mark, the other axes must be adequate configured with command *setref* or disabled with command *setaxis*.

The command is only functional if the following settings are made for all axes.

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

While a refmove is performed, the command interpreter of the controller is blocked.
Received commands are stored in the command FIFO but not executed until the refmove is finished.

Syntax:

[direction] [distance] ***refmove***

	Range	Unit
[direction]	+/-	
[distance]	+/-1000	rev.

Related commands:

setref, setrefvel, getrefst

Examples:

100 refmove

The controller moves all active axes in positive direction.
If the reference mark is found, the controller stops with the defined acceleration (command ***sa***).
If no reference mark is found, the axes are moving exactly 100 revolutions before they stop.

-7 refmove
With this command the controller moves all active axes in negative direction, to find the zero reference mark.

getrefst

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getrefst*** returns the status of the refmove procedure (command *refmove*).

The following operating states are coded in the binary representation of the returned decimal value.

Syntax:

[axis] ***getrefst***

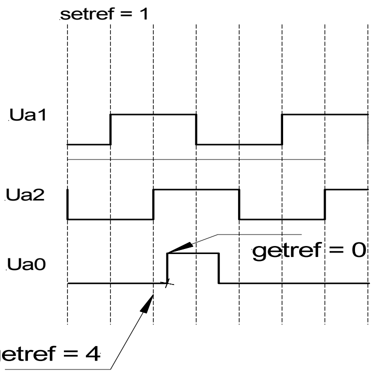
	Range
[axis]	-1, 1, 2, 3

Reply:

[decimal value]

[decimal value]
0, 1, 3, 4, 5

	Description
0	Reference mark found, active signal input
1	Reference mark not found
3	Reference mark not found, limit switch active
4	Reference mark found, zero signal input
5	Reference not found, zero signal input



U
a
1 / Ua2 = Quadrature input signal
Ua0 = Reference input signal
setting: ***setref*** = 1

Example:

2 getrefst
Returns the status of the refmove procedure of Axis-2

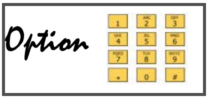
Trigger Output functions

setcloop

setcloop	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setcloop*** enables the Closed Loop mode.
This feature requires an external measurement system, the controller must be equipped with an analog or digital encoder interface



The On-Board digital encoder interface must be enabled with a release code.

The analog encoder interface is a hardware module, that must be installed from the factory or service personal.

For proper Closed Loop configuration see also the following commands:
setscaletype, setscaleinterface, setclperiod, setclpara, setnselpos

Syntax:

[index] [axis] ***setcloop***

	Range
[index]	0, 1
[axis]	1, 2, 3

[index]	Description
0	Closed Loop disabled
1	Closed Loop enabled

Related command:

getcloop

Example:

1 2 setcloop
0 3 setcloop

Axis-1 Closed Loop mode enabled
Axis-3 closed-loop mode disabled

getcloop

getcloop	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getcloop*** returns the Closed Loop status of the controller.

Syntax

[axis] ***getcloop***

	Range
[axis]	-1, 1, 2, 3

Reply:

[index]

	Range
[index]	0, 1

Example:

1 getcloop

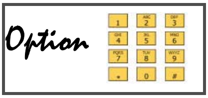
Reply:
1

outtrig (ot)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***ot*** generates a trigger output pulse at a specified I/O interface output. If several ***ot*** commands are performed, they will stored in a FIFO and executed one by one.



To support function "position capture", the controller must be equipped with the feature "Digital Input/Output".
Details, see in the hardware manual

I

Syntax:

[time] [pol] [output] ***ot***

	Description
[time]	Trigger out pulse width
[pol]	Trigger out polarity
[output]	I/O-Interface output

	Range	Unit	Function
[time]	1-1000	ms (integer)	
[pol]	0, 1		0 = active low 1 = active high
[output]	1, 2, 3		

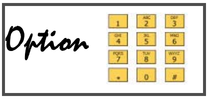
Example:

100 1 1 ot
A 100ms active high trigger pulse is generated at digital output 1.

waitposot (wpot)

Corvus T, Corvus eco, Corvus PCI

Description:



Command ***wpot*** (**w**ait_**pos**_**o**ut_**t**rigger) enables the position synchronized output function (PSO).
The maximum output frequency is 2kHz.
This means the smallest time between two triggers is 500µs.

With command ***setotmode*** the trigger output can be related to the actual or desired position.
Additionally it is possible to use the trigger as a source to store the actual position data in the position capture memory (see command *setpc*).

.

Syntax:

[pos] [dir] [axis] [time] [pol out] [output] ***wpot***

	Description
[pos]	Desired trigger output position
[dir]	Move direction where trigger out is enabled
[axis]	Specified axis
[time]	Trigger output pulse width
[pol_out]	Trigger out polarity, active low, active high
[output]	I/O Interface output

	Range	Unit	Function
[pos]	+/-16383	unit	
[dir]	0, 1		0 = negative direction 1 = positive direction
[axis]	1, 2, 3		Controller axes
[time]	0-1000	ms	
[pol_out]	0, 1		0 = active low 1 = active high
[output]	1, 2, 3		Equivalent I/O interface outputs

Example:

12.54 1 1 10 0 1 wpot

12.54 1 1 10 0 1 *wpot*

Digital output

Output trigger polarity

Output trigger pulse width

Desired axis

Move direction of trigger axis

Trigger out position of desired axis

waitpos (wp)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***wp*** (**w**ait_**p**os) interrupts the execution of all following commands, until the specified axis reaches the desired coordinate.

With Ctrl-C the command is cleared.

.

Syntax:

[pos] [dir] [axis] ***wp***

	Description
[pos]	desired coordinate
[dir]	move direction from where the position is reached
[axis]	specified axis

	Range	Unit	Function
[pos]	+/-16383	unit	specified unit for the axis
[dir]	0, 1		0 = negative direction 1 = positive direction
[axis]	1, 2, 3		equivalent controller axes

Example:

axis unit = 2 (mm)

12.54 1 1 *wp*

Axis

Move direction

Position

Command sequence:

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

With this sequence the axes are moving to the coordinates 100mm/10mm. If Axis-1 has reached position 12.54 mm, the controller executes command *getin* and returns the status of the digital inputs.

waittime (wt)

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***waittime*** (**w**ait_**t**ime) locks the command interpreter a specified time to disable the command execution.

While the interpreter is locked the controller is further on able to receive commands from the communication interface. These commands are executed if the interpreter gets unlocked.

The ***wt*** command does not interrupt the momentary executed command.

The ***wt*** command disables also the manual move.

Syntax:

[time_wait] [wait_unit] ***wt***

	Description
[time_wait]	time frame that the interpreter is locked
[wait_unit]	time unit

	Range	**Function**
[time_wait]		
[wait_unit]	0	Ticks (1 Tick = 250µs)
	1	seconds

Example:

1000 1 *wt*

Wait time unit

Wait time

Command sequence:

1000 0 wt
ge

The ***wt*** command configures the controller to reply the ***ge*** command after 1000 ticks = 0.25s

1 1 wt
ge

The ***wt*** command configures the controller to reply the ***ge*** command after 1s

waitintrigot (witot)

Corvus T, Corvus eco,
Corvus PCI

Description:

Command ***witot*** (wait_in_trigger out trigger) is a fast combination of the commands ***waitintrig (wit)*** and ***outtrig (ot)***
See the description in the appropriate descriptions.

To support this function, the controller must be equipped with the feature "Digital Input/Output".
Details, see in the hardware manual

Syntax:

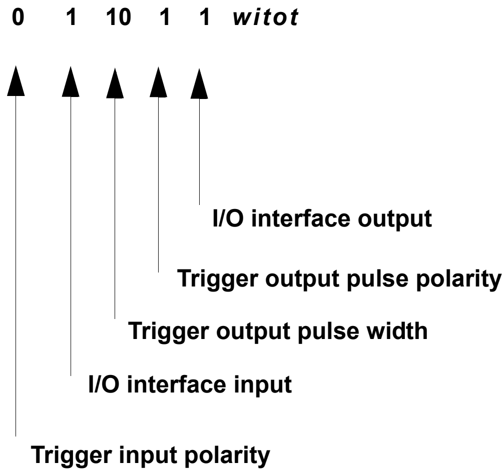
[pol_in] [input] [time] [pol_out] [output] ***witot***

	Description
[pol_in]	Trigger input polarity
[input]	I/O-interface input
[time]	Trigger output pulse width
[pol_out]	Trigger output pulse polarity
[output]	I/O-interface output

	Range	**Unit**	**Function**
[pol_in]	0, 1		
[input]	1, 2, 3		Equivalent I/O interface input
[time]	1-1000	ms	
[pol_out]	0, 1		
[output]	1, 2, 3		Equivalent I/O interface output

Example:

0 1 10 1 1 witot



waittimeot (wtot)

Corvus T, Corvus eco, Corvus PCI

Description:

wtot is fast combination of the commands ***waittime (wt)*** and ***outtrig (ot)***.
With command ***wtot*** (wait_time out_trigger) a delayed trigger output can be performed.
Details see command description ***wt*** and ***ot.***

Option 

To support this function, the controller must be equipped with the feature "Digital Input/Output".
Details, see in the hardware manual

Syntax:

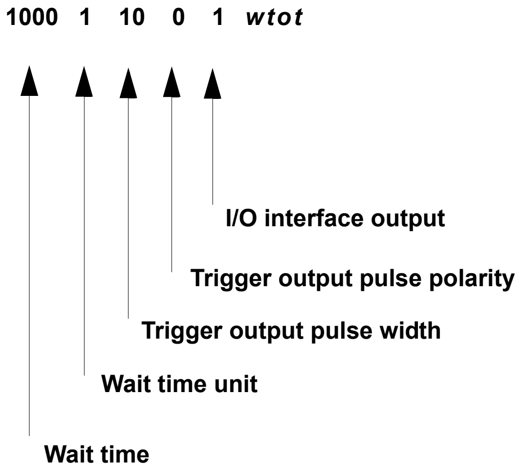
[time_wait] [wait_unit] [time_trigger] [pol_out] [output] ***wtot***

	Description
[time_wait]	waiting time
[wait_unit]	wait time unit
[time_trigger]	trigger pulse width
[pol_out]	trigger output polarity
[output]	I/O-interface output

	Range	**Unit**	**Function**
[time_wait]	integer value	wait unit	
[wait_unit]	0, 1		0 = ticks (250µs each tick) 1 = seconds (s)
[time_trigger]	1-1000 (integer value)	ms	
[pol_out]	0, 1		
[output]	1, 2, 3		

Example:

1000 1 10 0 1 wtot



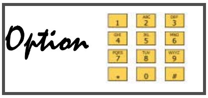
setrptdata

Corvus TT | Corvus eco | Corvus PCI

Valid from firmware version 4.5.0

Description:

Command ***setrtptdata*** initializes the Position-Interval-Triggering.
With this function the controller is able to generate output triggers in a regular frequency.
The trigger positions relate selectively to the desired position or the current (measured) position.



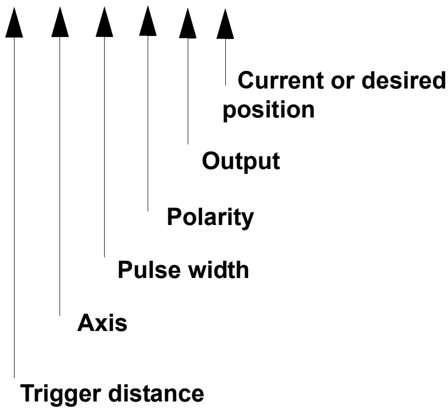
For this function the controller has to be equipped with the Digital Input/Output function.

.

Syntax:

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

1.234 1 100 1 0 1 *setrptdata*



	Description
[rpos]	Trigger distance (unit defined with setunit)
[axis]	The Triggers relate to this axis
[time]	Trigger pulse width
[pol]	Polarity of the trigger signal
[output]	Number of Digital Output
[selpos]	Trigger relates to current or desired position

	Range	Unit	Function
[rpos]	+/-16383mm	user unit	
[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 = desired position 1 = current position

Example:

1.234 2 100 1 2 0 setrptdata

The Position-Interval-Triggering is initialized as follows:
Trigger distance 1.234 units (i.e. mm, µm)
Related axis for triggering is Axis-2
Trigger pulse width = 100ms
Trigger comes at Digital Output-2
The trigger position relates to the desired position

Complete command sequence to generate triggers at equidistant coordinates.

1.234 2 100 1 2 0 setrptdata
0 10 startrpt
20 20 m

getrptdata

Description:

Command ***getrptdata*** returns the configuration of the Position-Interval-Trigger.

Syntax:

getrptdata

Reply:

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

	Description
[rpos]	Trigger distance (unit defined with setunit)
[axis]	The Triggers relate to this axis
[time]	Trigger pulse width
[pol]	Polarity of the trigger signal
[output]	Number of Digital Output
[selpos]	Trigger relates to current or desired position

Example:

getrptdata

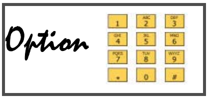
Reply:
1.234 2 100 1 2 0

startrpt	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.5.0

Description:

Command ***startrpt*** enables the Position-Interval-Trigger.
Additionally the absolute coordinate is determined where the Trigger starts or stops.
If the stop coordinate is reached, the Position-Interval-Trigger is disabled.
To execute further triggers, the ***startrpt*** command must be performed again with new coordinates.



Option "Digital Input/Output" must be released to use this function.

Syntax:

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

	Description
[Start]	coordinate where the trigger starts
[Stop]	coordinate where the trigger stops

	Range	**Unit**
[Start]	+/-16383mm	user
[Stop]	+/-16383mm	user

related commands:

setrptdata, getrptdata, setotmode

Examples:

10.234 12.56 startrpt

The trigger starts at coordinate 10.234 and stops at coordinate 12.56

Complete command sequence example:
Settings
Output: DOUT-1,
Trigger interval: 10µm
Trigger pulse width: 0.5ms
Related axis: Axis-2
Related position: Nominal

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

Trigger-Input functions

setotmode

	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.5.0

Description:

Command ***setotmode*** has two tasks:

1. It assigns the trigger input of the *wpot* command to the calculated position or the measured position.
2. Determines the output trigger as a trigger source to log position data (see command *setpc*).

Syntax:

[mode] ***setotmode***

[mode]	Trigger position	Data logging
0	calculated	off
1	measured *	off
2	measured *	on
3	calculated	on

* The controller must be equipped with the Closed Loop option

Example:

3 setotmode

getotmode

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getotmode*** returns settings made with command *setotmode.*

Syntax:

getotmode

Reply:

[mode]

	Range
[mode]	0, 1, 2, 3

[mode]	Trigger source	Position capture
0	calculated position	off
1	measured position	off
2	measured position	on
3	calculated position	on

Example:

getotmode

setpcin

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***setpcin*** initializes the trigger input for the "position capture" function.

To support function "position capture", the controller must be equipped with the feature "Digital Input/Output".
Details, see in the hardware manual

Syntax:

[edge] [input] ***setpcin***

[edge]	Description
0	data are latched with the rising edge
1	data are latched with the falling edge

[input]	Description
1	Input DIN 1 (Pin 6)
2	Input DIN 2 (Pin 2)
3	Input DIN 3 (Pin 7)

Example:

1 3 setpcin
To use the function "position capture", input 3 is determined as trigger input. The data will be latched into the "capture memory" with the rising edge of the trigger signal.

getpcin

getpcin	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.2.0

Description:

The command ***getpcin*** returns the settings for the function "position capture".

Syntax:

getpcin

Reply:
[edge] [input]

	Range
[edge]	0,1
[input]	1, 2, 3

Example:

getpcin

Reply: 1 3

setpc

Corvus TT | Corvus eco | Corvus PCI

Valid from firmware version 4.2.0

Description:

The command ***setpc*** enables or disables the function "position capture". This function stores the actual position data, triggered from an user input signal or internal output signal (see command ***setotmode***).

With command *setnselpos* it is determined if the internal calculated position is stored (nominal position) or the actual position, replied from a measurement system.

I

Command ***setpc*** must be executed again, if the setting of *setnselpos* was changed.

Syntax:

[On/Off] ***setpc***

[On/Off]	Description
0	position capture Off
1	position capture On

Example:

1 setpc

getpc

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getpc*** returns the status of the function "position capture", additionally the trigger counter is displayed.

If the selected input is triggered, a data record is stored, containing the position of all active axes, as well as the internal time, called ticks.
The stored records are indicated with the counter value of the trigger counter and can be recalled with this index number.
Overall 65000 records can be indicated.
The "capture memory" has a free memory space to store 1000 data records.
If more than 1000 records are captured, the data memory will be overwritten.

Syntax:

getpc
Reply:
[trigger counter] [status]

	Range
[trigger counter]	0-65000
[status]	0,1 0 = On, 1 =Off

Example:

getpc

Reply: 1204 1
The function "position capture" is active.
The trigger counter displays 1204 counts. That means the first 204 counts from the data memory are overwritten.

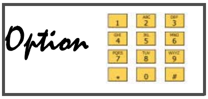
waitintrig (wit)

Corvus T, Corvus eco, Corvus PCI

Description:

Command ***wit*** (**w**a**i**t_**i**n_**t**rigger) configures the controller to interrupt the command interpreter until a specified input signal is active (level triggered).

- The ***wit*** setting is cleared after the command interpreter is is released.
- The command does not interrupt the momentary executed commands.
- The ***wit*** command always awaits the end of a momentary move command, it has no influence to the manual move.
- With ***Ctrl-C*** the command is aborted.



To support this function the controller must be equipped with the feature "Digital Input/Output".
Details, see in the hardware manual

Syntax:

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

	Description
[pol_in]	operate signal input polarity
[input]	I/O-interface input

	Range	Function
[pol_in]	0, 1	0 = active low 1 = active high
[input]	1, 2, 3	I/O interface input

Example:

0 1 *wit*

I/O interface input

operate signal input polarity

Command sequence for example:

0 1 wit st

With active low level at digital input-1 the interpreter is locked.
Venus-1 command ***ge*** is not executed.
If the signal level changes to high level, the command ***st*** will be executed and the ***wit*** setting is removed.

getpcdata (gpd)

Corvus TT, Corvus eco, Corvus PCI

Description:

The command ***getpcdata*** reads the "position capture data" that are recorded in the "capture memory".

Details about the function "position capture"
With each trigger input, a data record is stored. Each record contains the position of all active axes, as well as the controller time, called ticks.
Each record uses one memory cell in the "capture memory". The stored records are indicated with the counter value of the trigger counter and can be recalled with this index number, assumed it is not overwritten. Overall 65000 records can be indicated.
The "capture memory" has space for 1000 data records.

The minimum time resolution (tick) is 250 µs, maximum trigger input frequency is 2 kHz.
The trigger input is polled with the controller cycle time of 250 µs or 4 kHz.

Syntax:

[index record A] [index record B] ***getpcdata***

	range
[index record A]	1-65000
[index record B]	1-65000

Value A<B.

Reply:
[Tick] [Pos. Axis-1] [Pos. Axis-2] [Pos. Axis-3]

	Range of value
[Tick]	1 Tick = 250 µs Maximum time value after power up is reached in 298 hours.
[Pos. Axis-1] until [Pos. Axis-3]	Format depends on the settings of setunit, setdim and setnselpos

Example:

3450 3460 getpcdata

Data records with index 3450 until 3460 are recalled.

time	Pos. Axis-1	Pos. Axis-2	Pos. Axis-3
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

The values based on the following settings:

Velocity = 1mm/s,

Trigger frequency = 2 kHz, position unit = mm

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

9.750868 mm - 9750368 mm = 0.5 µm

With an axis speed of 1mm/s and a trigger frequency of 2 kHz, position data with an interval of 0.5 µm are stored in the "capture memory".

clearpcdata (cpd)

Corvus TT, Corvus eco, Corvus PCI

Description:

The command ***clearpcdata*** (cpd) clears the "position capture memory" and the trigger counter.

Syntax:

clearpcdata

or ***cpd***

Examples:

cpd

setintrigtimeout

Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 4.5.0.

Description:

With command ***sitto*** a time-out period can be defined for command *waitintrig*.
If the trigger input is not valid within these time, the *waitintrig* command is executed.

With waiting time setting = 0 the time-out period is infinite.

Syntax:

[time] ***setintrigtimeout***

	Description
[time] *	waiting time [s]
time = 0	waiting time infinite

	Range
[time]	0.01 to 100s

*factory setting = 0

Example:

10 sitto

Waiting time for trigger input is 10s.

getintrigtimeout

Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 4.5.0.

Description:

Command ***gitto*** returns the time-out setting for the trigger input signal of command *waitintrig.*

Syntax:

gitto

Reply:

[time]

	Range
[time]	0 0.01 to 100s

Example:

gitto

Joystick / Handwheel

setjoysticktype

Corvus TT, Corvus eco, Corvus PCI

Description:

With command ***setjoysticktype*** the controller is adjusted to the manual device, Joystick or Handwheel.

Syntax:

[Index*] ***setjoysticktype***

*factory setting = 3

	Range
[Index]	2, 3, 8

	Description
2	analog Joystick
3	analog Joystick
8	Handwheel

Related command:

getjoysticktype

Example:

8 setjoysticktype

getjoysticktype

Corvus TT, Corvus eco, Corvus PCI

Description:

Command ***getjoysticktype*** returns the settings of *setjoysticktype*.

Syntax:

getjoysticktype

Reply:

[Index]

Index	Description
2	analog Joystick
3	analog Joystick
8	Handwheel

Example:

getjoysticktype

joystick (j)

joystick (j)	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***joystick*** enables or disables the manual mode.
The activity of this mode is indicated in status bit D1 and also displayed at the front panel of the controller.

Special function in Joystick mode

After power up the zero setting of the joystick is checked.
A tolerance of +/- 10% is acceptable.
With greater deviations the appropriate axis can not be enabled for the joystick mode.

Syntax:

[index] ***joystick***

	Range
[index]	0, 1

	Function
0	Manual mode disabled
1	Manual mode enabled

Related command:

getjoystick (valid from Firmware Version 4.50)

Example:

1 j
Enables manual mode.

getjoystick (gj)

Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 4.50

Description:

Command ***getjoystick*** returns status of the manual mode.

The state of the manual mode is also reflected in the status reply, see command *status*.

Syntax:

getjoystick

Reply:

[value]

value	Function
0	manual mode disabled
1	manual mode enabled

Example:

getjoystick

setjoyspeed (js)

Corvus TT, Corvus eco, Corvus PCI

Description:

The command ***setjoyspeed*** defines the maximum velocity for the manual mode for all axes.

The rotational motor speed of each axis depends on the relation between the settings of the manual speed and spindle pitch (command ***setpitch***).
If an axis is configured with a low spindle pitch, the resulting rotational speed of the motor, with the given global manual speed setting, can exceed and bring the motor out of it´s operating range. This can cause the motor to stall.
In this cases it is recommended to configure the manual speed for each axis separately with the command *setnjoyspeed.*

Syntax:

[velocity] ***setjoyspeed***

	Unit
[velocity]	units (depends on the settings of 0-Axis, see command setunit)

	Range
min. velocity	15.62 nm/s
max. velocity	180 mm/s

Related command:

getjoyspeed

Example:

20 setjoyspeed
unit = mm
The joystick velocity of each axis is 20 mm/s

getjoyspeed (js)

Corvus TT, Corvus eco, Corvus PCI

Description:

The command ***getjoyspeed*** returns the adjusted maximum global velocity for the manual move.

Syntax:

getjoyspeed

Reply:

[velocity]

	Unit
[velocity]	unit 0-Axis

Example:

getjoyspeed

Reply:

20.000000

setnjoyspeed (njs)

Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 3.7.3

Description:

Command ***setnjoyspeed*** allows to define a individual maximum joystick speed for each axis.
The speed unit is depends on the unit of the 0-Axis, see command ***setunit***

The settings of ***setnjoyspeed*** are overwritten if command ***setjoyspeed*** is executed afterwards.

If an axis is configured with a low spindle pitch, the resulting rotational speed of the motor, with the given global manual speed setting, can exceed and bring the motor out of it´s operating range. This can cause the motor to stall.
In this cases it is recommended to configure the manual speed for each axis separately with the command *setnjoyspeed.*

Syntax:

[velocity] [axis] ***setnjoyspeed***

	Unit
[velocity]	units (see 0-Axis setting)

	Range
[axis]	1, 2, 3

Related command:

getnjoyspeed

Example:

20 1 setnjoyspeed
The maximum joystick velocity of Axis-1 is 20 mm/s

getnjoyspeed (njs)

Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 3.7.3

Description:

The command ***getnjoyspeed*** reads the settings of the adjusted axis specific manual speed.

Syntax:

[axis] ***getnjoyspeed***

	Range
[axis]	1, 2, 3

Reply:

[velocity]

	Unit
[velocity]	unit of the 0-Axis

Example:

1 getnjoyspeed

Reply:

20.000000

setjoybspeed

Corvus TT, Corvus eco, Corvus PCI

Description:

With command ***setnjoybspeed*** a second velocity for the manual device can be defined. This velocity gets active by pressing the switch at the Joystick or Handwheel.

Syntax:

[velocity] ***setjoybspeed***

	Unit
[velocity]	unit of 0-Axis

	Range
min. velocity	15.62 nm/s
max. velocity	60 rev/s x pitch

Related command:

getjoybspeed

Example:

0.01 setjoybspeed

unit = mm

As long as the joystick button is pressed, the maximum joystick velocity is 0.01 mm/s.

getjoybspeed

Corvus TT, Corvus eco, Corvus PCI

Description:

The command ***getnjoybspeed*** reads the secondary velocity of the manual device.

Syntax:

getjoybspeed

Reply:

[velocity]

	Unit
[velocity]	unit 0-Axis

Example:

getjoybspeed

Reply:

0.010000

setjoyassign

Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 4.40

Description:

With command ***setjoyassign*** the moving direction, generated from the Joystick and the assignment of the Joystick axes can be changed.

Syntax:

[assignment] [motor axis] ***setjoyassign***

	Range
[assignment]	-3 to +3
[motor axis]	1, 2, 3

[assignment]	Direction	Joystick axis
0	Joystick disabled	-
1	positive	X-Axis
2	positive	Y-Axis
3	positive	Z-Axis

[assignment]	Direction	Joystick axis
0	Joystick disabled	-
-1	negative	X-Axis
-2	negative	Y-Axis
-3	negative	Z-Axis

Default settings:

1 1 setjoyassign
2 2 setjoyassign
3 3 setjoyassign

Examples:

2 1 setjoyassign
1 2 setjoyassign
-3 3 setjoyassign

Motor axis-1 is moved with Joystick axis Y and motor axis-2 is moved with joystick axis X.
Motor axis-3 is moved with Joystick axis Z in a reversed direction

3 1 setjoyassign
3 2 setjoyassign
0 3 setjoyassign

Motor axis-1 and motor axis-2 are moved simultaneously with Joystick axis Z.
Motor axis-3 is disabled. The Joystick axis X and Y are without effect.

getjoyassign

Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 4.40

Description:

The command ***getjoyassign*** returns the assignment of the axis and moving direction of the Joystick.

Syntax

[motor axis] ***getjoyassign***

	Range
[motor axis]	1, 2, 3

Reply:

[joystick axis]

	Range
[joystick axis]	-3, -2, -1, 0, 1, 2, 3

Example:

1 getjoyassign

Reply:
3

setjoydiag

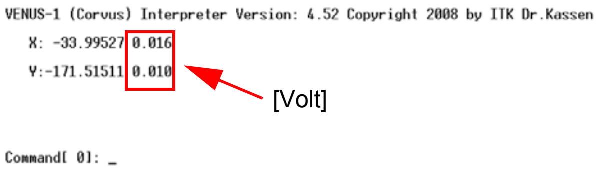
Corvus TT, Corvus eco, Corvus PCI

Valid from firmware version 4.41

Description:

Command ***setjoydiag*** activates the Joystick diagnostic feature.
If the function is enabled, the output voltage of each Joystick axis is returned and displayed in the Terminal window.

The function is available only in the Terminal Mode (1 mode).



Syntax:

[switch] ***setjoydiag***

	Range
[switch]	0, 1

	Function
0	Joystick diagnostic off
1	Joystick diagnostic on

Example:

1 mode
1 setjoydiag

The Terminal mode and the Joystick diagnosis feature will be enabled.

getjoydiag

Corvus TT, Corvus eco, Corvus PCI

Description:

With command ***getjoydiag*** the Joystick diagnosis setting is returned.

Syntax

getjoydiag

Reply:

[switch]

	range
[switch]	0, 1

Example:

getjoydiag

Reply:
0

setwheel

setwheel	Corvus TT	Corvus eco	Corvus PCI

Description:

Command getwheel initialises the Handwheel mode.
To activate the mode it is necessary to perform command *reset* after the *save* command
The Handwheel is enabled with command *setjoystick*
It is not possible to use Handwheel and Joystick mode simultaneously.

Option

For Handwheel operation the Corvus hardware must be prepared in the factory.

The Handwheel occupies the digital encoder interface, therefore the Closed Loop mode is only functional with the analog encoder interface (sin/cos Module).

Syntax:

[Index] ***setwheel***

	Range
[Index]	0, 1

[Index]	Description
0	Encoderbetrieb (default)
1	Handradbetrieb

Example:

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

Handwheel mode is initialized

getwheel

Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***getwheel*** the Handwheel initializing is checked.

Syntax

getwheel

Reply:

[Status]

	Range
[Index]	0, 1

[Status]	Description
0	Default setting (Encoder mode)
1	Handwheel is initialized

Example:

getwheel

setwheelres

Corvus TT | Corvus eco | Corvus PCI

Description:

With command ***getwheelres*** the controller is adapted to the number of electrical and mechanical pulses, the handwheel generates with one revolution (360°).

Factory settings: 100 pulses/rev.

Syntax:

[pulses] [Axis] ***setwheelres***

	Range
[pulses]	1-65535
[Axis]	1, 2, 3

Example:

200 1 setwheelres

# *getwheelres*	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getwheelres*** returns the expected pulses, generated from one handwheel revolution.

Syntax

[Axis] ***getwheelres***

Reply:

[Pulses]

	Range
[Pulses]	1.... 65535

Example:

1 getwheelres

setwheelratio

Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setwheelratio*** the ratio between one handwheel resolution and total stroke is defined.
Additionally the moving direction can be determined.

Example:
setwheelres = 100 pulses
setwheelratio = 1mm (total Stroke with one revolution)

Result:
Each handwheel pulse generates a stroke of
1mm / 100 pulses = 0.01 mm

With the handwheel speed button it is possible to change the total stroke for one handwheel revolution to a predefined value.
This value is parameterizised with command *setwheel**b**ratio.*

Syntax:

[Dir] [Stroke] [Axis] ***setwheelratio***

	Range	Unit
[Dir]	Motion direction	+/-
[Stroke]	-32767...+32767mm	unit
[Axis]	1, 2, 3	

Example:

-10 1 setwheelratio

getwheelratio

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getwheelratio*** returns the stroke that is generated with one handwheel revolution.

Syntax

[Axis] ***getwheelratio***

Reply:

[Dir] [Stroke]

	Range	Unit
[Stroke]	-32767mm...32767mm	unit

Example:

1 getwheelratio

setwheelbratio

setwheelbratio	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setwheelbratio*** a second ratio between one handwheel resolution and total stroke is defined similar to command *setwheelratio.*

This setting is activated with the speed button at the handwheel.

Example:
setwheelres = 100 pulses
setwheelratio = 1mm (fist total stroke)
setwheelbratio = 0.1mm (second total stroke)

Result:
With the settings above, each handwheel revolution generates a stroke of 1mm or 1/100= 0.01 mm with each pulse.
With the handwheel speed button, the resolution can be changed to the predefined value 0.1mm/100 = 0.001mm

Syntax:

[Dir] [Stroke] [Axis] setwheelbratio

	Range	Unit
[Dir]	direction	+/-
[Stroke]	-32767...+32767mm	unit
[Axis]	1, 2, 3	

Example:

0.1 1 setwheelbratio

getwheelbratio | Corvus TT | Corvus eco | Corvus PCI

Description:

Command ***getwheelbratio*** returns the settings of setwheelbratio.

Syntax

[Axis] ***getwheelbratio***

Reply:

[Dir] [Total Stroke generated with one handwheel revolution]

	Range	Unit
[Dir]	direction	+/-
[Stroke]	-32767...+32767mm	unit

Example:

1 getwheelbratio

System commands

save

Description:

The command ***save*** stores all active parameters in a non volatile memory. Always the last saved settings are restored after power on.

Parameters which can be saved are declared with following symbol.

Programmable moves are aborted if the ***save*** command is executed. Manual moves are only interrupted during time of saving.
The end of saving is indicated in Terminal Mode with the character OK
In Host Mode an automatic reply can be defined with the command sequence ***save*** and *status*.
Than the status information is replied after the **save** is finished.

Syntax:

save

Reply:

In Terminal Mode the characters ***OK*** are replied if the save is finished.

Example:

save

restore

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***restore*** reactivates the last saved parameters.
With the command sequence ***restore save*** the controller replies a status information after the restore is finished.

Syntax:

restore

Reply:

A reply can be enabled with the following command sequence
restore
status

Related command:

getfpara

Example:

restore

getfpara

getfpara	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getfpara*** activates the factory configuration.

Attention:
All current parameters are overwritten but can be restored with command *restore*.

Syntax:

getfpara

Reply:

The command execution reply can be controlled with the command
sequence:

getfpara
status

Example:

getfpara

clear

	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***clear*** deletes the content of the parameter stack.
The related command *gsp,* returns the current number of parameters on the stack.

In a accurate operation the number of parameters on the stack will always go to zero if all commands are processed.

It indicates an inaccurate use of a Venus-2 command, (i.e. to many parameters, wrong syntax) if elements remain on the stack. In a worst case the stack will overflow if more than 99 elements are put on the stack and cause a malfunction of the controller

Syntax:

clear

Related command:

gsp

Example:

clear

reset

	Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***reset*** preforms a device reset which is equal to disconnect the device from the power.
The proper state of the controller after a reset is indicated with beep (1s).

Syntax:

reset

Example:

reset

# beep	Corvus TT	-	-

Description:

Command beep triggers the internal beeper that produces a 1 Khz sound.
The length of the beep sound can be determined.

Maximum beep length = 10s

Syntax:

[beep length] beep

	Range	Unit
[beep length]	1-10,000	ms

Example:

1000 beep

version

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***version*** returns the version of the controller firmware.

Syntax:

version

Reply:

[Version number]

Related command:

identify

Example:

version

Reply:

4.5.5.

getmacadr

getmacadr	Corvus TT	-	-

Description:

Command ***getmacadr*** returns the Ethernet MAC Address.

Syntax:

getmacadr

Reply

[Mac-Address]

Example:

getmacadr

Reply in Terminal Mode:

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

Reply in Host Mode:

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

identify	Corvus TT	-	-

Description:

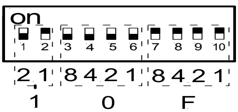
Command ***indentify*** returns the controller hardware and software revision revision.
Additionally the rear Dip-Switch setting is returned.
.

Syntax:

identify

Reply:

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

	Description
[Model]	Model type
[HW-Rev]	Hardware revision
[SW-Rev]	Software revision
[Board-Sw]	Internal use
[Dip-Sw]	Dip-Switch settings The returned value is hex coded. 

Related command:

version

Example:

identify
Reply: `Corvus 1 312 1 10F`

getoptions

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***getoptions*** returns a decimal number that indicates the released options.

Each option is assigned to a binary digit from D0 to D9.
If more options are released, the decimal values of the digits are added.
To get each released option separately, it is necessary to convert the decimal value into a binary value and mask the bit pattern.

Syntax:

getoptions

Reply:

[value]

	range
[value]	0 - 975

Bit	**decimal**	**released option**
D0	1	Axis-3 enabled
D1	2	Ethernet TCP/IP Interface
D2	4	Closed Loop / all axis
D3	8	digital Inputs/ Outputs, 3/3
D4	16	not used
D5	32	not used
D6	64	Closed Loop Axis-1
D7	128	Closed Loop Axis-2
D8	256	Closed Loop Axis-3
D9	512	speed grade 60 U/s

Example:

getoptions

Reply: 9

Axis-3 is released
Digital Input/Output is released.

Bit	decimal	released option
D0	**1**	Axis-3 enabled
D1	2	Ethernet TCP/IP Interface
D2	4	Closed Loop / all axis
D3	**8**	digital Inputs/ Outputs, 3/3
D4	16	not used
D5	32	not used
D6	64	Closed Loop Axis-1
D7	128	Closed Loop Axis-2
D8	256	Closed Loop Axis-3
D9	512	speed grade 60 U/s
summary	**9**	

getserialno

Corvus TT	Corvus eco	Corvus PCI

Description:

The command ***getserialno*** returns the serial number of the controller.

Syntax:

getserialno

Reply:

YY HW SERI

	Description	Digits
YY	Year	2
HW	Hardware Revision	2
SERIAL	Consecutive number	4

Related command:

identify, version

Example:

getserialno

Reply:
01020105
Description:
Year: 2001
Hardwarerevision 02
Serialnumber: 0105

Position error correction

setpcor

setpcor	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***setpcor*** the "Positioning Error Correction" function is switched on or off.

Syntax:

[function] [axis] ***setpcor***

	Range
[function]	0, 1
[axis]	1, 2, 3

[function]	Description
0	Error correction off
1	Error correction on

Related command:

getpcor

Example:

0 1 setpcor

Axis-1 Positioning Error Correction is switched off.

getpcor

Description:

Command ***getpcor*** returns the status of the Positioning Error Correction function.

.

Syntax:

[axis] ***getpcor***

	Range
[axis]	1, 2, 3

Reply:

[0,1]

	Function
[0]	Positioning Error Correction off
[1]	Positioning Error Correction on

Example:

1 getpcor
Returns the status of the Positioning Error Correction of Axis-1

setpdat

setpdat	Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***setpdat*** is used to enter the correction curve for the Positioning Error Correction function.

A word about positioning accuracy

Corvus was specially developed to control stepper motors with a very high resolution.
This archives smooth and precise positioning tasks.

However, it should be noted that a precise controlling of the stepper motor is not the solution to get also a accurate positioning system.
Positioning errors occur regardless of the controller, due to mechanical stiction, spindle pitch errors or load dependent positioning inaccuracies.

These disadvantages can be avoided with an additionally mounted measurement system. The controller therefore must work in Closed Loop mode. The gain of accuracy depends on the type of the measurement system.

Due to the costs or critical mechanical circumstances, it is not always possible to use a measurement system.

Alternative a cheap and effective method is to compensate mechanical errors from the controller directly in a open loop mode without the use of a measurement system.

The main principle of this method is to determine the error characteristics curve of a positioning system and store it into the controller. The controller than will calculate the positioning data accordant to this curve.

Description of the Corvus Positioning Error Correction

The Corvus Error Correction function operates for each axis separately.
The error characteristics curve must be determined from the start point in equidistant nodes and transmitted in ascending order to the controller with command ***setpdat.***

The distance between the nodes is fixed to 1mm.
During operation the position error at the nodes are 100% corrected. The error correction between the nodes are calculated from the controller with a linear interpolation.

With command ***save*** the correction values are stored in controller flash memory.

The Positioning Error Correction operates in manual and programmed mode with a travel range up to 499mm or 500 nodes.

Syntax:

$e_{0.}$....e_{499}] [start] [nodes] [axis] ***setpdat***

	Description	Range	Unit
[$e_{0.}$..e_{499}	nodes position errors (in ascending order)	+/- 100	µm
[start]	start point	0-499 (integer) [start] + [node] ≤ 499	mm
[nodes]	number of nodes	1-500	
[axis]	axis	1, 2, 3	

Related command:

getpdat

Examples:

0.5 0.1 0.5 1.2 -0.5 1.2 0 6 1 setpdat
0mm 1mm 2mm 3mm 4mm 5mm (nodes)

Axis-1 is corrected between 0 and 5mm.
Six error values are transmitted to the controller.

0.3 0.5 0.9 1.5 -0.5 6 5 2 setpdat
6mm 7mm 8mm 9mm 10mm

Axis-2 is corrected between 6 and 10 mm.
Five error values are transmitted to the controller.

getpdat

Description:

Command ***getpdat*** returns the positioning error data at the nodes in a sequence of 10 subsequent values.
With the command a start node is defined.

.

Syntax:

[start node] [axis] ***getpdat***

	Range	Unit
[start node]	0-499	mm
[axis]	1, 2, 3	

Reply:

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

	Range	Unit
E_{Start} ... $E_{Start+9}$	+/-100	µm

Example:

12 1 getpdat

Reply:

```
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

The command returns 10 error values from node 12 until node 21

setblc

	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 3.66

Description:

Command ***setblc*** enables or disables the backlash compensation.
With this function backlash errors caused by the reversal of travel direction are compensated.

Syntax:

[Function] [Axis] ***setblc***

[Function]	Description
0	compensation disabled
1	compensation enabled

	Range
[Function]	0, 1
[Axis]	1, 2, 3

Related command:

getblc, setblcd

Example:

1 1 setblc

# *getblc*	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 3.66

Description:

Command ***getplc*** returns the status of the function "backlash-compensation"

.

Syntax:

[Axis] ***getblc***

	Range
[Axis]	1, 2, 3

Reply:

[0,1]

	Function
[0]	compensation disabled
[1]	compensation enabled

Example:

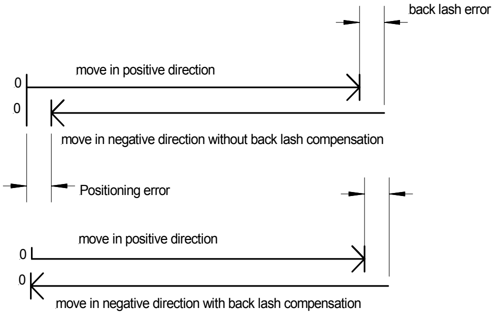
1 getblc

# setblcd	Corvus TT	Corvus eco	Corvus PCI

Valid from firmware version 3.66

Description:

Command ***setbldc*** allows to define the distance value that is compensated with the backlash function.
The compensation is always executed with an abrupt step at the start of a negative move.



Syntax:

[Distance] [Axis] ***setblcd***

	Description
[Distance]	distance that has to be compensated

	Range
[Distance]	0 - 0.1mm
[Axis]	1, 2, 3

smallest value: 0.000001mm

Example:

0.001 1 setblcd

getblcd

Valid from firmware version 3.66

Description:

Command ***getblcd*** replies the backlash distance value.

.

Syntax:

[Axis] ***getblcd***

	Range
[Axis]	1, 2, 3

Reply:

[0.000000 - 0.1mm]

Example:

1 getblcd

Corvus Macros

Corvus Makro FAQ

In the following description we use the German term makro because it is identical with the Venus-1 syntax.
The equivalent name in the English language would be macro.

What is a Corvus Makro:

A Makro in principal is a list of Venus-1 commands, that can be temporary stored and executed in the Corvus controller.

Example of a 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
```

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

Example:

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

Why Makros:

Makros are used to minimize the communication overhead to reduce CPU load and avoid communication time lag problems.
It is great benefit of the Corvus makro function that it relieves the application software to support high speed scanning applications. It also enhances the performance of the various Corvus trigger features.

How to create and execute Corvus Makros

The Corvus Makro is a simple text file that is transmitted via the RS-232 or Ethernet interface into the Makro-Exe Buffer.
After this, the Makro can be executed infinite times with a single start command.
Due to the control words *beginmakro* and *endmakro*.
the Makro list is transferred automatically into the Makro-Exe Buffer.
While a Makro is executed, it is only possible to send a abort command via the communication interface, other commands are not possible.

How many Venus-1 commands can be included in a Makro:

The size of a Makro is counted in symbols and not in Venus-1 command lines.
A Venus-1 command line can consist of one or several symbols.

A maximum of 4000 symbols can be transferred to the Makro-Exe Buffer.

For example the following Venus-1 command line requires three symbols: 100 100 move

Examples:

$\mathit{100}_1\ \mathit{100}_2\ \mathit{move}_3$
This command line consist of three symbols.

$\mathit{100}_1\ \mathit{100}_2\ \mathit{10}_3\ \mathit{move}_4$
This command line consist of four symbols.

st
This command line consist of one symbol.

The following Makro consist of 13 symbols.

beginmakro

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

endmakro

Is it possible to execute Makros in a Loop

The command ***startmakro*** can be called within a Makro, therefore it is possible to execute Makros in a endless loop.

Example:

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

Is it possible to execute a Makro automatically

Not available yet.

To execute commands automatically after power up, use Venus-1 command ***setpowerup***.

Is it possible to store various Makros

Not available yet.

Makro commands overview

For Makro controlling and makro administration the following commands are provided.

beginmakro
Labels the start of a Makro.

endmakro
Labels the end of a Makro.

startmakro
Executes Makro in the Makro-Exe Buffer.

listmakro
Returns the number of symbols in the Makro-Exe Buffer.

Ctrl-D
Interrupts a Makro execution or Makro download.

Macro functions

beginmakro / endmakro

Corvus TT, Corvus eco, Corvus PCI

Description:

The command ***beginmakro*** and ***endmakro*** indicates the begin and the end of a Makro.

Syntax:

beginmakro / endmakro

Example:

```
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
90 1 1 200 0 2 wpot gt
ge
0 0 0 m
endmakro
```

startmakro

	Corvus TT	Corvus eco	Corvus PCI

Description:

With command ***startmakro*** the Makro in the Makro-Exe Buffer is executed.

To execute a Makro in a endless loop, call ***startmakro*** within the Makro.

Syntax:

startmakro

Example:

startmakro

listmakro

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***listmakro*** returns the number of used Symbols in the Makro-Exe buffer.
A maximum of 4000 Symbols can be transferred into the buffer.

Error code 1201 indicates if more symbols are transferred.

Syntax:

listmakro

Example:

listmakro

Return:
1204

Ctrl-D

Corvus TT	Corvus eco	Corvus PCI

Description:

Command ***Ctrl-D*** interrupts a Makro execution or a Makro download.

Syntax:

Ctrl-D

ASCII sign	decimal value	hex value
Ctrl-D	4	0x4

Example:

Ctrl-D

Venus-1 command overview

Basic settings

Example:***1 getpolepairs -1 getpolepairs***

Command ***setaxis*** enables or disables the specified axis for positioning tasks.
Example:***1 3 setaxis***

The command ***getaxis*** returns the setting of ***setaxis.***
Example:***2 getaxis -1 getaxis***

With command ***setpowerup*** it is possible to execute
fixed commands automatically after power up.
Example:***15 setpowerup***

Command ***getpowerup*** returns the Power up command settings of the controller.
Example:***getpowerup***

With command ***setphaseares*** it is possible to reduce the resolution of the motor drivers in incremental steps.
Example:***2 1 setphaseares***

Command ***getphaseares*** returns the motor resolution value of the selected axis.
Example:***2 getphaseares***

With command ***setmotiondir*** the factory assigned
relationship between the direction of motor rotation and the motion direction can be reversed.
Example:1 1 ***setmotiondir***

Command ***getmotiondir*** indicates, if the relationship
between motor direction and motion direction differs from the factory settings.
Example:***1 getmotiondir***

Communication

Command ***mode*** enables Terminal or Host Mode.

Velocity and acceleration

Example:***getmanaccel***

Command ***setcalvel*** defines two velocities for the cal limit-switch move. The setting is significant for all axes.
Example:***2 1 setcalvel***

The command ***getcalvel*** returns the adjusted velocities for cal limit-switch move.
Example:***getcalvel***

Command ***setncalvel*** defines the two velocities for the ncal limit-switch move.
Example:***2 1 2 setncalvel***

The command ***getncalvel*** returns the ncal limit-switch move velocities.
Example:***2 getncalvel***

The command ***setrmvel*** defines the two velocities for the rm limit-switch move. The setting is significant for all axes.
Example:***2 1 setrmvel***

Command ***getrmvel*** returns the two adjusted ***rm*** move velocities.
Example:***getrmvel***

Command ***setnrmvel*** configures the two velocities for the ***nrm limit switch*** move.
Example:***2 1 1 setnrmvel***

Command ***getnrmvel*** returns the two adjusted ***nrm*** movement velocities.
Example:***2 getnrmvel***

Command ***setrefvel*** defines the velocity with which the move to a reference mark is executed.
Example:***0.5 1 setrefvel***

Command ***getrefvel*** returns the setting of ***setrefvel***.

Example:***getrefvel***

Positioning commands

Command ***move*** executes point to point positioning tasks to absolute coordinates based on the point of origin. The move profile is calculated in respect to the velocity/acceleration setup and the given hard or software limits.
Example:***12.5 20.0 0.0001 m***

Command ***rmove*** executes point to point positioning tasks relative to the current position.
Example:***0.5 20 0.0001 r***

Command ***speed*** starts a constant velocity move.
Example:***10 1 speed***

Command ***stopspeed*** interrupts the constant velocity move of all axes with the adjusted acceleration.
See command ***sa***.
Example: ***stopspeed***

Command ***test*** preforms a positioning test procedure.
Example:***10 1 test***

Command ***randmove*** moves all active axes to randomized coordinates with a randomized velocity/acceleration setup.
Example:***randmove***

Limit Switch functions

The command ***cal*** executes the limit-switch move to the cal limit-switches. All active axes are simultaneously moved in negative direction, until the cal-switches are in ON state.
Example:***cal***

The command ***rm*** executes the limit-switch move to the rm limit-switches. All active axes are simultaneously moved in positive direction, until the rm-

The command ***nrm*** executes a single axis limit-switch movement to the rm limit-switch. This procedure determines the upper limit of the selected axis.
Example:***1 nrm***

The command ***getnlimit*** returns the current limits of a specified axes.
Example:***1 getnlimit***

Command ***org*** a moves the specified axis a relative stroke until the org-switch is in ON state.
Example:***-10 1 org***

The command ***setorg*** enables or disables the org-switch input.
Example:***1 1 setorg***

Command ***getorg*** returns the org-input settings of the specified axis.
Example:***1 getorg***

The command ***setorgsw*** adapts the specified org-switch input to the connected switch type.
Example:***0 1 setorgsw***

The command ***getorgsw*** returns the setting of the org-switch inputs.
Example:3 getsw

The command ***getorgswst*** returns the current activity of the org limit-switch input.
Example:***1 getorgswst***

Safety functions

Command ***Ctrl-C*** interrupts the current executed command. Moves will be stopped immediately with the acceleration setup, defined with command sa.
Example:***Ctrl-C***

position / origin / coordinate system

Status requests

Input / Output functions

Closed Loop commands

Trigger Output functions

Trigger-Input functions

command setotmode.
Example:***getotmode***

Joystick / Handwheel

setjoysticktype **255**
With command setjoysticktype the controller is adjusted
to the manual device, Joystick or Handwheel.
Example:***8 setjoysticktype***

getjoysticktype **256**
Command ***getjoysticktype*** returns the settings of
setjoysticktype.
Example:***getjoysticktype***

joystick (j) **257**
The command ***joystick*** enables or disables the manual mode.
The activity of this mode is indicated in status bit D1 and
Example:***1 j***

getjoystick (gj) **258**
Command ***getjoystick*** returns status of the manual mode.
Example:***getjoystick***

setjoyspeed (js) **259**
The command ***setjoyspeed*** defines the maximum velocity for the manual
mode for all axes.
Example:***20 setjoyspeed***

getjoyspeed (js) **260**
The command ***getjoyspeed*** returns the adjusted maximum global
velocity for the manual move.
Example:***getjoyspeed***

setnjoyspeed (njs) **261**
Command ***setnjoyspeed*** allows to define a individual
maximum joystick speed for each axis.
The speed unit is depends on the unit of the 0-Axis, see command ***setunit***
Example:***20 1 setnjoyspeed***

getnjoyspeed (njs) **262**
The command ***getnjoyspeed*** reads the settings of the adjusted axis
specific manual speed.
Example:1 ***getnjoyspeed***

setjoybspeed **263**
With command ***setnjoybspeed*** a second velocity for the manual device
can be defined. This velocity gets active by pressing the switch at the
Joystick or Handwheel.

and total stroke is defined.

Example:***-10 1 setwheelratio***

getwheelratio 276
Command getwheelratio returns the stroke that is generated with one handwheel revolution.
Example:***1 getwheelratio***

setwheelbratio 277
With command setwheelbratio a second ratio between one handwheel resolution and total stroke is defined similar to command setwheelratio.

Example:***0.1 1 setwheelbratio***

getwheelbratio 278
Command getwheelbratio returns the settings of setwheelbratio.
Example:***1 getwheelbratio***

System commands

save 281
The command ***save*** stores all active parameters in a non volatile memory. Always the last saved settings are restored after power on.
Example:***save***

restore 283
The command ***restore*** reactivates the last saved parameters.
Example:***restore***

getfpara 285
The command ***getfpara*** activates the factory configuration.
Example:***getfpara***

clear 287
The command ***clear*** deletes the content of the parameter stack.
Example:***clear***

reset 289
The command reset preforms a device reset which is equal to disconnect the device from the power.
Example:***reset***

beep 291
Command beep triggers the internal beeper that produces a
Example:***1000 beep***

Position error correction

Command ***getplc*** returns the status of the function "backlash-compensation"
Example:***1 getblc***

setblcd 315
Command ***setbldc*** allows to define the distance value that is compensated with the backlash function.
Example:0.001 1 ***setblcd***

getblcd 316
Command ***getblcd*** replies the backlash distance value.
Example:***1 getblcd***

Macro functions

beginmakro / endmakro 327
The command ***beginmakro*** and ***endmakro*** indicates the begin and the end of a Makro.

startmakro 329
With command ***startmakro*** the Makro in the Makro-Exe Buffer is executed.
Example:***startmakro***

listmakro 331
Command ***listmakro*** returns the number of used Symbols in the Makro-Exe buffer.
Example:***listmakro***

Ctrl-D 333
Command ***Ctrl-D*** interrupts a Makro execution or a Makro download.
Example:***Ctrl-D***