SensorReady 3D

Instruction Set

Rev.1.2

1. Table of Contents

2. Introduction

Communication interface:
All SensorReady 3D devices communicate via a USB Interface, established as a virtual COM port. True RS-232 communication is an optional feature.

Axes:
SensorReady 3D devices support up to 3 Encoders also called axes (x,y,z). The axis specifiers used in the SensorReady 3D instruction set are the ASCII characters x, y and z. Axes can be addressed individually by using the axis specifier or combined if no axis is specified in the instruction.

Instruction syntax:
The instructions and parameters are sent as plain text ASCII strings with a terminating carriage return [CR], which is 0x0d hex. Characters may be upper-, lower- or camel-case. The parameters are separated by a space character.
This provides easy access to all functions by using a simple terminal program such as HyperTerminal. A typical instruction syntax is as follows:

[!,?][instruction][SP][optional axis] [parameter1][SP][parameter2] [etc…] [CR]

[!,?] Read/write specifier, required by all instructions:
! (exclamation mark) = to write parameter, execute an instruction etc.
? (question mark) = to read data (returns settings, or status, etc.)

[instruction]	Is the instruction word itself.
[SP]	Space (ASCII 0x20 hex) as separation.
[optional axis]	Axis character x, y or z if only one axis must be addressed.
[parameter]	Usually integer or floating point numbers, floating point uses decimal point, no comma.
[CR]	Termination (ASCII 0x0d hex), causes instruction execution.

A read instruction may return more than one parameter. In many cases the number of returned parameters depends on the amount of activated axes, set by the **'encnumber'** parameter:
[axis X] [if active: axis Y] [if active: axis Z]

For some instructions that return decimal numbers (e.g. ?originoffset, ?vs, ?encperiod, ?voltcontrol and more) the number of returned fractional digits is fixed. For "?pos" and similar position returning functions (e.g. "corrdata"), the number of decimal digits can be set by the '**resolution**' parameter.

Syntax examples:

!dim 1 1 0	set measuring unit for X and Y to [mm] and for Z to [µm]
!save	save parameter settings to device nonvolatile memory
!enctype y 3	set encoder type for Z-axis to analog 1Vpp
?pos	returns position of all axes (e.g. 0.0000 0.0000 0.0000)
?encperiod x	returns encoder signal period of X-axis only (e.g. 0.500000)

Settings:
Most settings can be stored permanently in the SensorReady 3D device, so they are available from power on. When stored once, this reduces initilization overhead of the application software. Refer to the "**save**" instruction for further information. Parameters that are saved can be identified by a 'Y' in the Save column of the **brief instruction set description** later in this document.

Character limits:
To prevent the input buffer from overflow, please do not send more than 255 characters at once.
Such may occur when sending the setup sequence to the SensorReady 3D device. A good practice is to request the **“?err”** state after each setup instruction. This will return the information if the parameters were accepted or not while preventing overflow.

Important: Measuring units!
The measuring unit is set by the **“dim”** instruction, where dim 1 [which is mm] is the default setting.

3. Hint for initialization

The SensorReady 3D device must be configured to meet the hardware requirements. The configuration can be stored permanently with the **“save”** instruction.
It is recommended to save and reboot the device after changing the setup parameters (e.g. !**enctype**, !**originref**, !**originsw**) to ensure all changes will be applied.

- The encoder types: !**enctype**
- The encoder periods: !**encperiod**
- The axis units: !**dim**

Brief Description of the SensorReady 3D Instruction Set

Device Informations					
Instruction		**Example**	**Save**	**Brief description**	**Page**
?	version	?version	-	Read detailed firmware and device version	10
?	vs	?vs	-	Read firmware version number only	10
?!	serialnr	?serialnr	Y	Set customer specific number	10
?	voltcontrol	?voltcontrol	-	Read current values of operating voltages (3.3 V, 5 V, 24 V)	11

Communication Interface Settings (available with RS-232 option)					
Instruction		**Example**	**Save**	**Brief description**	**Page**
? !	baudtt	!baudtt 7	Y	Set RS-232 baud rate to 115200 Bd (default = 57600)	12

System Instructions					
Instruction		**Example**	**Save**	**Brief description**	**Page**
!	save	!save	-	Save parameters to device nonvolatile memory	13
!	reset	!reset	-	Reset device (forces restart, similar to cycle power)	13
!	setdefaults	!setdefaults 1	-	Reset device to factory settings	14

Error Messages					
Instruction		**Example**	**Save**	**Brief description**	**Page**
?	err	?err	-	Read error number	15

General Adjustments					
Instruction		**Example**	**Save**	**Brief description**	**Page**
? !	dim	!dim 0 0 0	Y	Set position units of X, Y and Z to µm	16
? !	resolution	!resolution 6	Y	Set position return string resolution to 1 nm	17
? !	language	!language 2	Y	Set device language to English	17

Switch Instructions					
Instruction		**Example**	**Save**	**Brief description**	**Page**
?	swrefstatus	?swrefstatus	-	Read actuation state of all reference switches	18
?	swrefstatusl	?swrefstatusl	-	Read+clear latched actuation state of all reference switches	18
?	footswitch	?footswitch	-	Read foot switch actuation states	19
? !	footswitchl	?footswitchl	-	Read+clear latched actuation states of foot switch	19

Machine Zero Instructions					
Instruction		**Example**	**Save**	**Brief description**	**Page**
? !	originsw	!originsw x 1	Y	Use reference switch to determine machine zero of X-axis	20
? !	originref	!originref z 1	Y	Use reference mark to determine machine zero of Z-axis	21
? !	originoffset	!originoffset 5 5 5	Y	Set offset from machine zero to 5 mm for all axes	21

Encoder Instructions

Instruction		Example	Save	Brief description	Page
? !	pos	!pos 0 0 1.5 ?pos x	-	Set current X-, Y-encoder positions to 0 and Z position to 1.5 Read current X-encoder position	22
? !	enctype	!enctype z 1	Y	Set encoder type of Z-axis to TTL	23
? !	encperiod	!encperiod x 0.1	Y	Set signal period of X-encoder to 100 µm	24
? !	encdir	!encdir y 1	Y	Reverse counting direction for Y-encoder	24
? !	encvoltage	!encvoltage x 1	Y	Enable supply voltage for X-encoder (activate X-encoder)	25
? !	swapxy	!swapxy 1	Y	Change axis assignment for X and Y (switch X ↔ Y)	25
? !	encnumber	!encnumber 2	Y	Set number of active encoders (2 = X, Y)	26
?	encrefstatus	?encrefstatus x	-	Read X-encoder reference signal state (1 = on reference)	26
?	encrefstatusl	?encrefstatusl y	-	Read latched Y-encoder reference signal state	26
?	encnasstatus	?encnasstatus y	-	Read Y-encoder NAS signal state (1 = NAS error)	27
?	encnasstatusl	?encnasstatusl y	-	Read latched Y-encoder NAS signal state	28
?	encamp	?encamp x	-	Read X-encoder signal amplitude in percent	28
?	hwcount	?hwcount	-	Read all encoder positions (TTL counter, no interpolation)	28
?	encsin	?encsin x	-	Read A/D conversion result of the sine signal for X	29
?	enccos	?enccos y	-	Read A/D conversion result of the cosine signal for Y	29

MR Encoder Instructions

Instruction		Example	Save	Brief description	Page
?	mroffsin	?mroffsin x	-	Read offset correction value of the sine signal for X	30
?	mroffcos	?mroffcos	-	Read offset correction value of the cosine signal for all axes	30
?	mrcosamp	?mrcosamp x	-	Read amplitude correction factor (sine /cosine ratio) of X	31

Position Correction Instructions

Instruction		Example	Save	Brief description	Page
? !	corr	!corr 1 1 1	Y	Enable position correction for all axes	32
?	corrdata	?corrdata x	-	Read all saved position correction data of X-axis	32

4. Instruction Syntax Description

Most instructions work in both directions (reading and writing). (?)! means the instruction accepts write and read access. The device identifies a read instruction by a preceding '?', while '!' indicates writing to a parameter or executing an instruction. More information can be found in the **Introduction** chapter of this document.

Some examples of legal instruction syntax:
!*Instruction parameter1 parameter2 parameter3*
!*Instruction axis parameter*
!*Instruction*
?*Instruction axis*
?*Instruction*

5. Error Numbers and their possible Root Cause

0	No Error
1	No valid Axis Name
2	Unknown Instruction
3	Number is not inside allowed Range
4	Wrong Data Length (number of arguments)
5	Either ! or ? is missing
99	Device is in Bootloader Mode

6. Device Informations

The firmware version may be read by sending the instruction **“version”** to the controller.

6.1. version (Read detailed Version Information)

Syntax: ?version
Parameter: none

Description: Read the SensorReady 3D type and firmware version.

Response syntax: Character string including device type, firmware version and build date separated by a comma, e.g.
SensorReady 3D, Version 1.20, November 04 2013

Different SensorReady 3D types: SensorReady 3D

Version 1.05 Firmware version number
November 04 2013 Firmware build date

Example:
?version ==> SensorReady 3D, Version 1.05, November 04 2013

Remark: Currently, the SensorReady device replies: “PROFILER ST”

6.2. vs (Read Firmware Version Number)

Syntax: ?vs
Parameter: none

Description: Read the firmware version number only.
The firmware version number has 2 decimal places.

Response: Firmware version number

Example:
?vs ==> 1.20

6.3. serialnr ()

Syntax: ?!serialnr
Parameter: none

Description: Read or set the customer specific number.
The response is a string of maximal 8 ASCII characters.

Response: Serial number of the corrected microscope stage as maximal 8 character ASCII string

Example:
?serialnr ==> 12110616

6.4. voltcontrol (Read Values of Operating Voltages)

Syntax: ?voltcontrol
Parameter: none

Description: Read current values of operating voltages.
The device has 3 different operating voltages, the optimal voltage values are 3.30 V, 5.00 V and 24.00 V.
The operating voltages are returned with two decimal places.

Response: Current values of operating voltages

Example:
?voltcontrol ==> 3.31 5.16 23.90

7. Communication Interface Settings

7.1. baudtt (Baud Rate) (Option)

Syntax: !baudtt or ?baudtt
Parameter: 3, 4, 5, 6 or 7

Description: Set or read the baudrate of the serial COM Port interface. It applies only to the true RS-232 serial connection of the device.

3 => 9600 [Bd]
4 => 19200 [Bd]
5 => 38400 [Bd]
6 => 57600 [Bd]
7 => 115200 [Bd]

Remarks: For the USB interface of the device this instruction has no effect, as communication is managed by the driver at a very high, internally fixed baudrate. In this case it does not matter which baudrate the virtual COM port is opened with, it has no effect on performance.

After sending this instruction the PC has to re-open the COM Port with the new baudrate, else no communication is possible. Then a **"!save"** instruction may be sent to permanantly store the new baudrate in the controller.

Response: Current baud rate of the controller as decimal value

Example:
!baudtt 6 Set the baud rate to 57600 [Bd]

?baudtt ==> 6 Read currently set baud rate

8. System Instructions

8.1. save (Save Parameters)

Syntax: !save
Parameter: none

Description: The save instruction permanently stores the parameter settings (e.g. encoder type, encoder period) to the SensorReady 3D device. These parameters will be applied as default values after each consecutive power-on or reset.

Response: none

Example:
!save Save current parameter settings

8.2. reset (Force a Software Reset)

Syntax: !reset
Parameter: none

Description: The device is forced to perform a software reset. It is a restart similar to power on. Rebooting from reset will take more than 1 second, where the device is not responding. There is no reply to a software reset.

Remarks: After a software reset, no connection can be established via the USB interface.
To establish a new connection over the virtual COM port, the opened connection must be closed and the SensorReady 3D device must to be restarted again. A device restart can be forced by using the on/off switch or by disconnecting the power supply (removing the external power adapter from the mains).

A connection via the true RS-232 serial interface is not affected by a reset. When the device was restarted, it responds to incoming instructions again.

Response: none

Example:
!reset Force a software reset

8.3. setdefaults (Restore Factory Settings)

Syntax: !setdefaults
Parameter: 0 or 1

Description: Reset SensorReady 3D device to factory settings.
Software reset is performed automatically.

0 => Current parameter settings are retained
1 => Restore factory settings

Remarks: All user settings will be lost.
Saved position correction values remain kept when restoring the factory setting.

After the automatically performed software reset, no connection can be established via the USB interface.
To establish a new connection over the virtual COM port, the opened connection must be closed and the SensorReady 3D device must to be restarted again. A device restart can be forced by using the on/off switch or by disconnecting the power supply (removing the external power adapter from the mains).

A connection via the true RS-232 serial interface is not affected by a reset. When the device was restarted, it responds to incoming instructions again.

Response: none

Example:
!setdefaults 1 Reset device to factory settings

9. Error Messages

9.1. err (Read Error Number)

Syntax: ?err or !err
Parameter: none

Description: The instruction ?err return the device error state or 0, if no error occurred. The error state wil be updated or re-set by the next instruction.
If not a permanent error (like e.g. 99) the error state may be cleared to zero by sending !err.

Response: Error number as decimal value
(refer to Chapter 5. **“Error Numbers”**)

Example:
?err ==> 0 May return a 0

!err Clear error state if no permanent error

10. General Adjustments

With the following instructions the parameters of the device are widely scalable to the given mechanic construction and to customer requirements. The device is adaptable to the requested requirements.

10.1. dim (Unit for Positions)

Syntax: !dim or ?dim
Parameter: x, y, z or none
0 to 5

Description: The dim instruction sets or reads the unit (here “dimension”) of the input and output parameters related to length, e.g. position.

0 => µm
1 => mm (default)
2 => cm
3 => m
4 => inch
5 => mil

Response: Current dim settings

Example:

```
!dim 1 1 0     Set dim unit for X and Y to [mm] and for Z to [µm]
!dim y 2       Set dim unit for Y to [cm]
!dim 0 0 0     Set dim unit for axes X, Y and Z to [µm]

?dim     ==> 0 0 0     Read dim unit of all axes
?dim x   ==> 0         Read dim unit of X-axis only
```

10.2. resolution (Position Number Format)

Syntax: !resolution or ?resolution
Parameter: 0, 1, 2, 3, 4, 5 or 6

Description: This instruction sets or reads the resolution of “?pos” and similar position returning instructions for dim 0 to 5. It affects the amount of returned decimal places, as listed below. One value applies to all axes, default = 3 (1µm resolution).

Value	Resolution dim 1	Resolution dim 3
0	= 1 mm	1 m
1	= 0.1 mm	0.1 m
2	= 0.01 mm	0.01 m
3 (default)	= 0.001 mm	0.001 m
4	= 0.0001 mm	0.0001 m
5	= 0.00001 mm	0.00001 m
6	= 0.000001 mm	0.000001 m

Affected instructions are: ?pos, ?corrdata.

Response: Responded decimal places for the “pos” and other position returning instructions.

Example:
!resolution 5 Set position read resolution to 10 nm (5 decimal places and default unit mm) e.g. “?pos x” returns 0.00000 for all dim settings.

?resolution ==> 5 Read currently set resolution

10.3. language (User Interface Language)

Syntax: !language or ?language
Parameter: 1, 2 or 3

Description: This instruction sets or reads the user interface language of the SensorReady 3D device.

1 => German
2 => English
3 => French

Response: Current language setting as decimal number

Example:
!language 1 Set the user interface language to German

?language ==> 1 Read currently set user interface language

11. Switch Instructions

11.1. swrefstatus (Read Status of Reference Switch)

```
Syntax:           ?swrefstatus
Parameter:        x, y, z or none

Description:      The swrefstatus instruction reads the actuation state of the
                  reference switch.
                  A reference switch is actuated, when an axis is in this
                  switch.

                  0  => Reference switch is currently not actuated (TTL high)
                  1  => Reference switch is currently actuated (TTL low)

Response:         Actuation state of reference switch(es)

Example:
?swrefstatus    ==> 0 0 0   Read all 3 reference switch actuation states
?swrefstatus x  ==> 0       Read reference switch actuation state of X-axis only
```

11.2. swrefstatusl (Latched Status of Reference Switch)

```
Syntax:           ?swrefstatusl
Parameter:        x, y, z or none

Description:      This instruction reads the latched actuation state of the ref-
                  erence switch.
                  If the reference switch was actuated since last reading of
                  swrefstatusl, a 1 is returned. The corresponding latch state
                  is cleared after reading.

                  0  => Reference switch is/was not actuated (TTL high)
                  1  => Reference switch is/was actuated (TTL low)

Response:         Latched actuation state of reference switch(es)

Example:
?swrefstatusl     ==> 0 1 0   Read+clear latched actuation state of all switches
?swrefstatusl y   ==> 1       Read+clear latched actuation state of X-axis only
```

11.3. footswitch (Read Status of Foot Switch)

```
Syntax:           ?footswitch
Parameter:        1, 2 or none
                  0 or 1

Description:      The footswitch instruction reads the actuation state of the
                  foot switch.

                  0  => Foot switch is currently not actuated (TTL high)
                  1  => Foot switch is currently actuated (TTL low)

Remarks:          The SensorReady 3D device has two foot switch inputs:
                  Foot switch 1 (left pedal):   Zero position values of all axes
                  Foot switch 2 (right pedal):  Send position values to the PC

Response:         Actuation state of foot switch(es)

Example:
?footswitch    ==> 0 1  Read both foot switch actuation states
?footswitch 1  ==> 0    Read actuation state of foot switch 1 (left pedal) only
?footswitch 2  ==> 1    Read actuation state of foot switch 2 (right pedal) only
```

11.4. footswitchl (Latched Status of Foot Switch)

```
Syntax:           ?footswitchl or !footswitchl
Parameter:        1, 2 or none
                  0 or 1

Description:      This instruction reads the latched actuation state of the foot
                  switch.
                  If the foot switch was actuated since last reading of
                  footswitchl, a 1 is returned. The corresponding latch state is
                  cleared after reading.

Remarks:          The SensorReady 3D device has two foot switch inputs:
                  Foot switch 1 (left pedal):   Zero position values of all axes
                  Foot switch 2 (right pedal):  Send position values to the PC

Response:         Latched actuation state of foot switch(es)

Example:
?footswitchl    ==> 1 0 Read+clear latched actuation state of both foot switches
?footswitchl 1  ==> 1   Read+clear latched actuation state of foot switch 1 only
?footswitchl 2  ==> 0   Read+clear latched actuation state of foot switch 2 only

!footswitchl     Reset latched actuation state of both foot switches to zero
```

12. Machine Zero Instructions

12.1. originsw (Use Reference Switch)

```
Syntax:          !originsw or ?originsw
Parameter:       x, y, z or none
                 0 or 1

Description:     This instruction enables the function for an axis to determine
                 the machine zero with a reference switch.
                 If this function is enabled, the origin for the respective ax-
                 is is set by approaching the reference switch.

                 0  => Reference switch not used
                 1  => Reference switch used for machine zero

Remarks:         The machine zero is set with the reference switch of the re-
                 spective axis, which is approached after enabling the function
                 or after each start of the SensorReady 3D device.
                 The following reference switch actuations are ignored.

Response:        Reference switch used, not used

Example:
!originsw 1 1 0     Use reference switch as origin for X- and Y-axis
!originsw x 1       Use reference switch as origin for X-axis

?originsw     ==> 1 1 0     Read reference switch usage of all axes
?originsw z   ==> 0         Read reference switch usage of Z-axis
```

12.2. originref (Use Encoder Reference Mark)

Syntax: !originref or ?originref
Parameter: x, y, z or none
0 or 1

Description: This instruction enables the function for an axis to determine the machine zero with an encoder reference mark.
If this function is enabled, the origin for the respective axis is set by approaching the first reference mark.

0 => Encoder reference mark not used
1 => Encoder reference mark used for machine zero

Remarks: The machine zero is set with the first encoder reference mark, which is approached after enabling the function or after switching on the SensorReady 3D device.
The following reference marks are ignored.

If the used measuring system has several reference marks, it must be ensured to approach the reference mark, which is used for configuration of the position correction. Otherwise there can be no proper position correction.

Response: Encoder reference mark used, not used

Example:
!originref 0 1 1 Use encoder reference mark as origin for Y- and Z-axis
!originref z 1 Use encoder reference mark as origin for Z-axis

?originref ==> 0 0 1 Read encoder reference mark usage of all axes
?originref z ==> 1 Read encoder reference mark usage of Z-axis

12.3. originoffset (Offset from Machine Zero)

Syntax: !originoffset or ?originoffset
Parameter: x, y, z or none
-1000.0 to 1000.0 [mm]

Description: This instruction sets or reads the offset from machine zero, which indicates the deviation between the machine zero and the zero point of the sample/workpiece.
When a machine zero is fixed with the selected function, the position value of the corresponding axis is set to the specified offset value.
The unit is always [mm] and the read-resolution is 4 decimal places.

Remarks: To exactly approach a determined zero point of a sample/workpiece after setting/approaching the machine zero, indicate the offset from machine zero (deviation between machine zero and zero point of the sample/workpiece).

Response: Current offset value(s) [mm]

Example:
!originoffset 5.5 5.5 0 Set offset value for X and Y to 5.5 mm, Z to 0 mm
!originoffset y 22.4 Set offset value of Y-axis to 22.4 mm

?originoffset ==> 5.5000 22.4000 0.0000 Read offset value of all axes
?originoffset x ==> 5.5000 Read offset value of X-axis only

13. Encoder Instructions

The SensorReady 3D devices support incremental encoders with or without a reference mark and absolute encoders with a SSI interface. The type of encoder (analog 1Vpp, analog 5Vpp, TTL/RS422 or SSI) should be configured by the instruction **“enctype”**.

13.1. pos (Read or Set Encoder Position)

Syntax: ?pos or !pos
Parameter: x, y, z or none
Position value

Description: This instruction either reads or sets the encoder position.
If set, this defines a new absolute position of the axis.
The unit depends on the selected dimension (**dim**).

Response: Encoder position(s) (depends on **dim** and **resolution**)

Example:

```
?pos     ==> 0.000 0.000 2.500     Read all encoder positions
?pos z   ==> 2.500                 Read Z-encoder position only

!pos 100 200 5     Set the current X-, Y- and Z-encoder positions
!pos -0.1          Set the current X position to -0.1 (unit depends on dim)
!pos y 2000        Set the current Y position to be 2000 (unit depends on dim)
```

13.2. enctype (Encoder Type)

Syntax: !enctype or ?enctype
Parameter: x, y, z or none
1, 2, 3 or 4

Description: This instruction sets or reads the type of encoder signal.
In order to ensure a correct evaluation, the encoder type setting of each axis must be adapted to the respective connected encoder.
If digital encoders (A/B-TTL, RS422) are used with an analog encoder interface (encoder type configured for 1Vpp or 5Vpp), the corresponding encoder type has to be set to TTL (RS-422). Else the TTL signal will be found as invalid (due to signal level) and not be used.

1 => TTL (RS-422)
2 => analog MR (5Vpp)
3 => analog 1Vpp
4 => SSI (absolute encoder)(Option)

Remarks: Not every type of SensorReady 3D device supports all available encoder types.
In the following table are listed, which type of SensorReady 3D support which encoder types.

SensorReady 3D Type	Encoder	Encoder Type			
		TTL (RS-422)	MR (5Vpp)	1Vpp	SSI
SensorReady 3D	Encoder 1, 2, 3 (X, Y, Z)	X	X	X	

Response: Encoder type(s)

Example:
!enctype 2 2 1 Set encoder type for X and Y to MR (5Vpp) and for Z to TTL
!enctype z 3 Set encoder type for Z to 1Vpp

?enctype ==> 2 2 3 Read encoder type of all axes
?enctype z ==> 3 Read encoder type of Z-axis only

13.3. encperiod (Encoder Signal Period)

```
Syntax:           !encperiod or ?encperiod
Parameter:        x, y, z or none
                  0.000002 to 4.0 [mm]

Description:      This instruction reads or sets the encoder signal period.
                  The unit is always [mm] and the read-resolution is 6 decimal
                  places.

Response:         Encoder signal period(s)

Example:
!encperiod 0.5 0.5 0.001     Set encoder period for X and Y to 500 µm, Z to 1 µm
!encperiod z 0.02            Set encoder period of Z-axis to 20 µm
!encperiod x 0.00001960784   Set encoder period of X-axis

?encperiod     ==> 0.000020 0.500000 0.020000    Read encoder period of all axes
?encperiod z   ==> 0.020000                      Read encoder period of Z-axis
```

13.4. encdir (Encoder Counting Direction)

```
Syntax:           !encdir or ?encdir
Parameter:        x, y, z or none
                  0 or 1

Description:      The encdir instruction sets or reads the encoder counting di-
                  rection.

                  0  => Encoder counting direction default
                  1  => Encoder counting direction reversed

Response:         Encoder counting direction

Example:
!encdir 1 1 1     Reverse encoder counting direction for all axes
!encdir x 1       Reverse encoder counting direction for X-axis only

?encdir     ==> 1 1 1     Read encoder counting direction of all axes
?encdir y   ==> 1         Read encoder counting direction of Y-axis only
```

13.5. encvoltage (Encoder Supply Voltage)

Syntax: !encvoltage or ?encvoltage
Parameter: x, y, z or none
0 or 1

Description: This instruction activates or disables the supply voltage for an encoder.
When no encoder is connected, the supply voltage for the respective encoder input can be disabled.

0 => Disable supply voltage
1 => Activate supply voltage

Remarks: For disabled supply voltage, the respective encoder is not functional and can not be used. The connected encoder returns no signals.

Response: Encoder voltage setting(s)

Example:
!encvoltage 1 1 0 Activate supply voltage for X- and Y-encoder
!encvoltage z 0 Disable supply voltage for Z-encoder

?encvoltage ==> 1 1 0 Read voltage setting of all encoders
?encvoltage x ==> 1 Read voltage setting of X-encoder only

13.6. swapxy (Swap X- and Y-encoder inputs)

Syntax: !swapxy or ?swapxy
Parameter: 0 or 1

Description: The swapxy instruction changes the axis assignment of the X- and Y-encoder inputs (switch X ↔ Y).

0 => X- and Y-encoder inputs are not exchanged (default)
1 => X- and Y-encoder inputs are swapped (X ↔ Y)

Remarks: The position correction values are also exchanged.
When exchanging the X- and Y-encoder inputs, the saved position correction values are allocated to the respective axis.

Response: Current state of axis assignment

Example:
!swapxy 1 Change axis assignment of the X- and Y-encoder inputs (X ↔ Y)

?swapxy ==> 1 Read current state of axis assignment

13.7. encnumber (Number of active Encoders)

```
Syntax:           !encnumber or ?encnumber
Parameter:        1, 2 or 3

Description:      This instruction sets or reads the number of active encoders.

                  1  => X-encoder is active
                  2  => X- and Y-encoder are active
                  3  => X-, Y- and Z-encoder are active

Remarks:          Inactive encoders are not used. The measurement function for
                  these encoders is disabled and the respective position display
                  is not displayed.

Response:         Number of active encoders

Example:
!encnumber 3     Activate X-, Y- and Z-encoder
!encnumber 2     Activate X- and Y-encoder and disable Z-encoder

?encnumber   ==> 2     Read current number of active encoders
```

13.8. encrefstatus (Encoder REF Signal State)

```
Syntax:           ?encrefstatus
Parameter:        x, y, z or none

Description:      Returns the REF signal input state.

                  0  => REF signal is inactive
                  1  => REF signal is active (encoder is on a reference mark)

Response:         Encoder reference signal state

Example:
?encrefstatus     ==> 1 0 0     Read REF signal state of all axes
?encrefstatus x   ==> 1         Read REF signal state of X-axis only
```

13.9. encrefstatusl (Latched Encoder REF Signal State)

```
Syntax:           ?encrefstatusl
Parameter:        x, y, z or none

Description:      Returns the latched REF signal input state.
                  If the REF signal was active since last reading of
                  encrefstatusl, a 1 is returned. The corresponding latch
                  state(s) are cleared after reading.

                  0  => REF signal is/was inactive
                  1  => REF signal is/was active (encoder is/was on a reference
                                                  mark)

Response:         Latched encoder reference signal state

Example:
?encrefstatusl     ==> 0 0 0  Read+clear latched REF signal state of all axes
```

```
?encrefstatusl x   ==> 0      Read+clear latched REF signal state of X-axis only
```

13.10. encnasstatus (Encoder NAS Signal State)

```
Syntax:           ?encnasstatus
Parameter:        x, y, z or none

Description:      Returns the NAS error signal input state.

                  0  => NAS signal is inactive (encoder signals ‘no error’)
                  1  => NAS signal is active   (error flag is set by encoder)

Response:         Encoder NAS error signal state

Example:
?encnasstatus     ==> 0 0 0     Read NAS signal (error) state of all axes
?encnasstatus x   ==> 0         Read NAS signal (error) state of X-axis only
```

13.11. encnasstatusl (Latched Encoder NAS Signal State)

```
Syntax:           ?encnasstatusl or !encnasstatusl
Parameter:        x, y, z or none

Description:      Returns the latched NAS error signal input state.
                  If the NAS error signal was active since last reading of
                  encnasstatusl, a 1 is returned. The corresponding latch
                  state(s) are cleared after reading.

                  0  => NAS signal is/was inactive (encoder signals 'no error')
                  1  => NAS signal is/was active   (error flag is/was set by
                                                    encoder)

Response:         Latched encoder NAS error signal state

Example:
?encnasstatusl     ==> 0 0 0  Read+clear latched NAS signal state of all axes
?encnasstatusl x   ==> 0      Read+clear latched NAS signal state of X-axis only

!encnasstatusl     Reset latched NAS signal state of all axes to zero
```

13.12. encamp (Encoder Signal Amplitude)

```
Syntax:           ?encamp
Parameter:        x, y, z or none

Description:      Read the encoder signal amplitude.
                  100 (percent) represents the maximum undistorted signal ampli-
                  tude.

Remarks:          In case of single ended TTL encoders the amplitude might be
                  returned as 0.

Response:         Encoder signal amplitude in percent as integer

Example:
?encamp     ==> 57 74 0     Read all encoder signal amplitudes
?encamp x   ==> 57          Read X-encoder signal amplitude
```

13.13. hwcount (Hardware Counter)

```
Syntax:           ?hwcount
Parameter:        x, y, z or none

Description:      Hwcount returns the position(s) of the independent TTL encoder
                  counter. It is a digital counter that counts the signal slopes
                  (4 per period) and does not provide signal interpolation. So
                  one signal period corresponds to a counter reading of 4.

Response:         Encoder hardware counter value(s)

Example:
?hwcount     ==> 24 16 0     Returns the position counter of all axes
?hwcount x   ==> 24          Returns the position counter of X-axis only
```

13.14. encsin (Sine Signal A/D Conversion Result)

```
Syntax:           ?encsin
Parameter:        x, y, z or none

Description:      The encsin instruction reads the A/D conversion result of the
                  sine signal as 12-bit signed digits (-2048 to +2048).
                  The returned value is not offset corrected, it is the un-
                  changed A/D conversion value.

Response:         A/D conversion result of the sine signal(s)

Example:
?encsin    ==> 550 -75 1250  Returns the A/D conversion results of all encoders
?encsin y  ==> -75           Returns the A/D conversion result of Y-encoder only
```

13.15. enccos (Cosine Signal A/D Conversion Result)

```
Syntax:           ?enccos
Parameter:        x, y, z or none

Description:      The enccos instruction reads the A/D conversion result of the
                  cosine signal as 12-bit signed digits (-2048 to +2048).
                  The returned value is not offset corrected, it is the un-
                  changed A/D conversion value.

Response:         A/D conversion result of the cosine signal(s)

Example:
?enccos    ==> 700 650 -800  Returns the A/D conversion results of all encoders
?enccos x  ==> 700           Returns the A/D conversion result of X-encoder only
```

14. MR Encoder Instructions (Option)

14.1. mroffsin (MR Sine Offset Correction Value)

```
Syntax:           ?mroffsin
Parameter:        x, y, z or none

Description:      This instruction reads the sine offset compensation value as
                  12-bit signed digits (-307 to +307). The offset correction
                  value can be maximal 15 percent of the sine signal, what is
                  equivalent to the range of ±307 digits.

                  This value is calculated automatically and permanently by mov-
                  ing the measuring stage and can not be changed. This calculat-
                  ed correction value is kept after switching off the
                  SensorReady 3D device and is always fed into the operation.

Response:         Currently used sine offset correction value(s)

Example:
?mroffsin     ==> 84 42 24     Read MR signal offset value sine of all axes
?mroffsin y   ==> 42           Read MR signal offset value sine of Y-axis only
```

14.2. mroffcos (MR Cosine Offset Correction Value)

```
Syntax:           ?mroffcos
Parameter:        x, y, z or none

Description:      This instruction reads the cosine offset compensation value as
                  12-bit signed digits (-307 to +307). The offset correction
                  value can be maximal 15 percent of the cosine signal, what is
                  equivalent to the range of ±307 digits.

                  This value is calculated automatically and permanently by mov-
                  ing the measuring stage and can not be changed. This calculat-
                  ed correction value is kept after switching off the
                  SensorReady 3D device and is always fed into the operation.


Response:         Currently used cosine offset correction value(s)

Example:
?mroffcos     ==> -12 -3 -13   Read MR signal offset value cosine of all axes
?mroffcos x   ==> -12          Read MR signal offset value cosine of X-axis only
```

14.3. mrcosamp (MR Amplitude Correction Factor)

Syntax: ?mrcosamp
Parameter: x, y, z or none

Description: This instruction reads the cosine amplification correction factor of the analogue encoder signal (here: sin/cos amplitude ratio).
This factor is calculated automatically and permanently by moving the measuring stage and can not be changed. This calculated correction factor is kept after switching off the SensorReady 3D device and is always fed into the operation.

Response: Currently used cosine amplification correction factor(s)

Example:

```
?mrcosamp    ==> 1.004 1.008 1.002  Read MR signal correction factor of all axes
?mrcosamp x  ==> 1.004              Read MR signal corr. factor of X-axis only
```

15. Position Correction Instructions

15.1. corr (Position Correction Enable)

```
Syntax:           !corr or ?corr
Parameter:        x, y, z or none
                  0 or 1

Description:      This instruction activates the position correction.

                  0  => Position correction disabled
                  1  => Position correction enabled

Response:         Position correction state(s)

Example:
!corr 1 1 1     Position correction enabled for all axes
!corr x 1       Position correction enabled for X-axis only

?corr    ==> 1 1 1     Read position correction states of all axes
?corr x  ==> 1         Read position correction state of X-axis only
```

15.2. corrdata (Position Correction Values)

```
Syntax:           ?corrdata
Parameter:        x, y or z

Description:      This instruction reads the saved position correction values of
                  an axis.
                  The returned position correction values depend on the selected
                  dimension (dim) and resolution (resolution).

Response:         Position correction values (depends on dim and resolution)

Example:
?corrdata x  ==>   0.000  0.000  Read saved position correction values of X-axis
                  20.000 19.995
                  40.000 39.996
                  60.000 60.001
                  80.000 80.000
                 100.000 99.993
?corrdata z  ==>   0.000  0.000  Read saved position correction values of Z-axis
                                 (no values are saved for this axis)
```

16. Document Revision History

No.	Revision	Date	Changes	Remarks
00	1.0	19.12.2013	-	Initial version
	1.1	Feb. 2014	minor changes	
	1.2	13.03.14	minor changes	page 23