



Command Reference

Description of the Nanotec firmware serial commands

NANOTEC ELECTRONIC GmbH & Co. KG
Gewerbestraße 11
D-85652 Landsham near Munich, Germany

Tel. +49 (0)89-900 686-0
Fax +49 (0)89-900 686-50
info@nanotec.de

Editorial / About this manual

© 2008

Nanotec[®] Electronic GmbH & Co. KG

Gewerbestraße 11

D-85652 Landsham / Pliening, Germany

Tel.: +49 (0)89-900 686-0

Fax: +49 (0)89-900 686-50

Internet: www.nanotec.com

All rights reserved!

MS Windows 98/NT/ME/2000/XP are registered trademarks of the Microsoft Corporation.

Target group

This technical manual is aimed at programmers who wish to program their own driver software for communication with drivers for the following Nanotec motors:

- Nanotec SMC1 stepper motors
- Nanotec Plug & Drive motors

About this manual

This technical manual must be read carefully before the Nanotec firmware command references are used for creating driver programs.

Nanotec[®] reserves the right to make technical alterations and further develop hardware and software in the interests of its customers to improve the function of this product without prior notice.

This manual has been written with due care. It is exclusively intended as a technical description of the Nanotec firmware command references. The warranty is limited to the repair or replacement of defective equipment of the Nanotec stepper motors, according to our general terms and conditions; liability for damage or errors resulting from the incorrect use of the command references for the programming of motor drivers is excluded.

For criticisms, proposals and suggestions for improvement, please contact the above address or send an email to: info@nanotec.de

Version/Change overview

Version	Date	Changes
V1.0	30.06.2008	Created

Contents

Editorial / About this manual	2
Contents	3
1 General information	5
1.1 Command structure.....	5
1.2 Note on old command format.....	5
1.3 Development support.....	6
2 Read command	10
3 Records	11
4 General commands	12
4.1 Start the bootloader.....	12
4.2 Joystick mode dead range	12
4.3 Read out the error memory.....	13
4.4 Read out the encoder position	13
4.5 Set the error correction mode	14
4.6 Set the filter for the analogue and joystick modes	14
4.7 Set the step angle	15
4.8 Read out the firmware version	15
4.9 Mask the inputs	16
4.10 Read out the inputs and outputs.....	17
4.11 Read out the firmware version (old).....	17
4.12 Read out the status.....	18
4.13 Read out the motor address	18
4.14 Read out a parameter	19
4.15 Read out the position	19
4.16 Set the motor address.....	20
4.17 Set the step mode	20
4.18 Set the phase current.....	21
4.19 Set the phase current at standstill.....	21
4.20 Set the motor mode.....	22
4.21 Set the limit switch behaviour	23
4.22 Set the limit switch type.....	24
4.23 Save a record.....	25
4.24 Set the reverse clearance	26
4.25 Reset the position	26
4.26 Set automatic sending of the status.....	27
4.27 Set the record for auto correction	27
4.28 Set the encoder direction	28
4.29 Set the swing out time.....	28
4.30 Set the maximum encoder deviation.....	29

Contents

4.31	Set the minimum voltage for the analogue mode	29
4.32	Set the maximum voltage for the analogue mode	30
4.33	Set the dead range for the joystick mode	30
4.34	Start a record	30
4.35	Stop a record.....	31
4.36	Increase the speed.....	31
4.37	Reduce the speed	31
4.38	Triggering	32
4.39	Reset a position error.....	32
4.40	Load a record from the EEPROM	32
4.41	Read out the current record	33
5	Record commands	34
5.1	Set the positioning mode	34
5.2	Set the travel distance.....	36
5.3	Set the minimum frequency	36
5.4	Set the maximum frequency	37
5.5	Set the maximum frequency 2	37
5.6	Set the ramp.....	38
5.7	Set the direction	38
5.8	Set the change of direction	39
5.9	Set the repetitions	39
5.10	Set the record pause.....	40
5.11	Set the continuation record	40
Index		41

1 General information

1.1 Command structure

Driver command structure

A command begins with start character '#' and ends with carriage return '\r'. All characters between the start and stop characters are ASCII characters (i.e., they are not control characters).

The start character is followed by the address of the motor as an ASCII decimal number.

This value may range from 1 to 254. If '*' is sent instead of the number, all drivers connected to the bus are addressed.

This is followed by the actual command, which generally consists of an ASCII character and an optional ASCII number. This number must be written in decimal notation with a prefix of '+' or '-'.

When the user sends a setting to the firmware, a '+' sign is not mandatory for positive numbers.

Note: Some commands consist of multiple characters while others do not require a number as a parameter.

Driver response

If a driver recognizes a command as relevant to it, it confirms receipt by returning the command as an echo, but without the '#' start character.

If the driver receives an unknown command, it responds by returning the command followed by a question mark '?'.

The response of the driver ends with a carriage return '\r', like the command itself.

Examples

Set the travel distance of driver 1:	"#1s1000\r" -> "1s1000\r"
Start a record:	"#1A\r" -> "1A\r"
Invalid command:	"#1/\r" -> "1/?\r"

1.2 Note on old command format

Downward compatibility

To maintain downward compatibility with older Nanotec drivers, the old command format is still supported in the current version of the firmware (09-11-2007).

In the old command format, when numbers consist of multiple bytes, each byte is converted to a decimal number (0 to 255) and then sent (e.g. 255 => 255 ; 256 => 1000 ; 257 => 1001).

This type of conversion is applied in **NONE** of the commands described here. All commands described here use normal decimal numbers.

Description of the old command format

A description of the old command format is not included in this documentation. For information on the old format, refer to the Plug & Drive Series PDx-I (Version 2.1) Manual.

1.3 Development support

Overview

The following manual describes the commands for communication with Nanotec drivers via the serial or USB interface. This enables you to address our drivers with any programming language and from any suitable programmable device.

This section briefly discusses the following points:

- DLL library
- Application example of the DLL library
- Windows help on the DLL library

DLL library

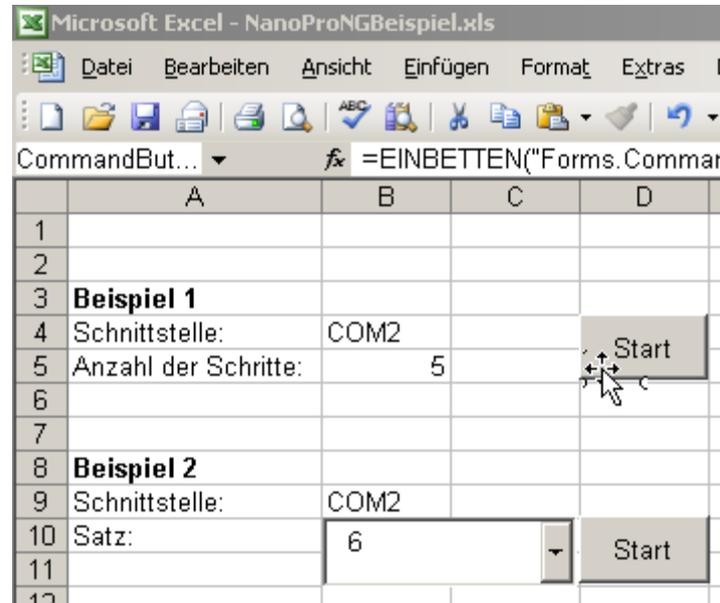
If you work under Windows with a .NET-capable programming language (e.g. Visual Basic of version 8 or higher, C#, Delphi.NET), we offer you the convenience of being able to integrate our DLL library in your application. You can conveniently control the functions of the driver via a function call without having to concern yourself with communication details.

The Dynamic Link Library (DLL), included as a development aid, thus lets you quickly, comfortably and correctly integrate the supplied command record into your individual motor control application.

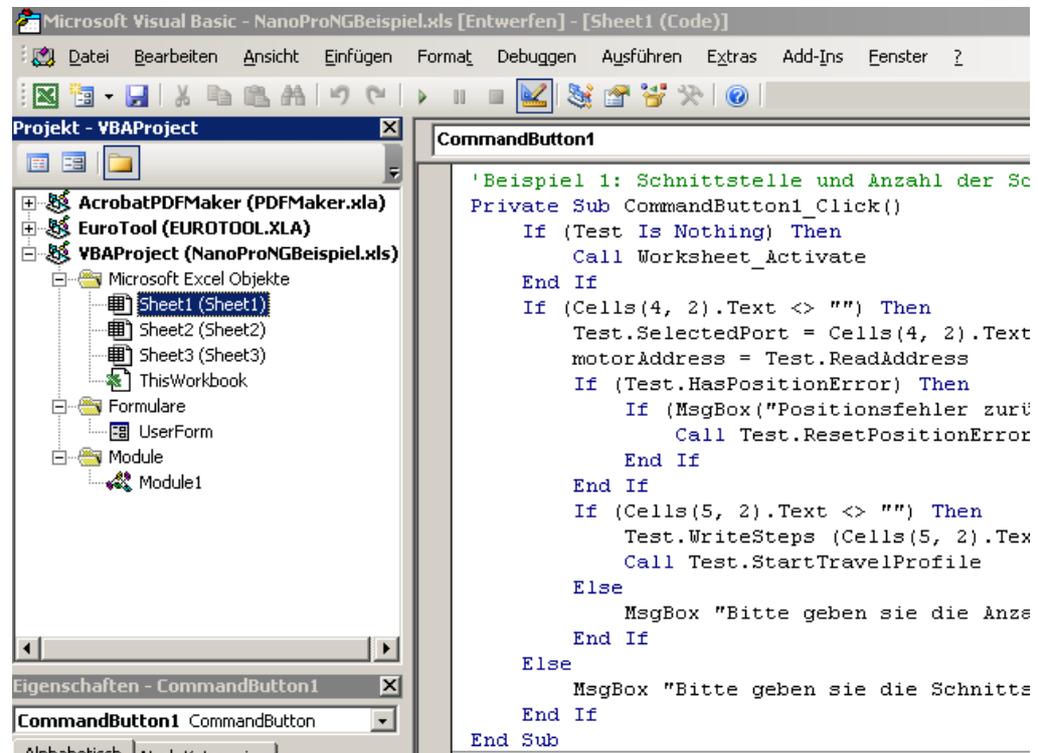
DLL library application

The EXCEL example shown here illustrates how our drivers can be addressed in a rapid and straightforward manner by means of the Visual Basic for Applications (VBA) scripting language and using the DLL.

Call button in an EXCEL worksheet



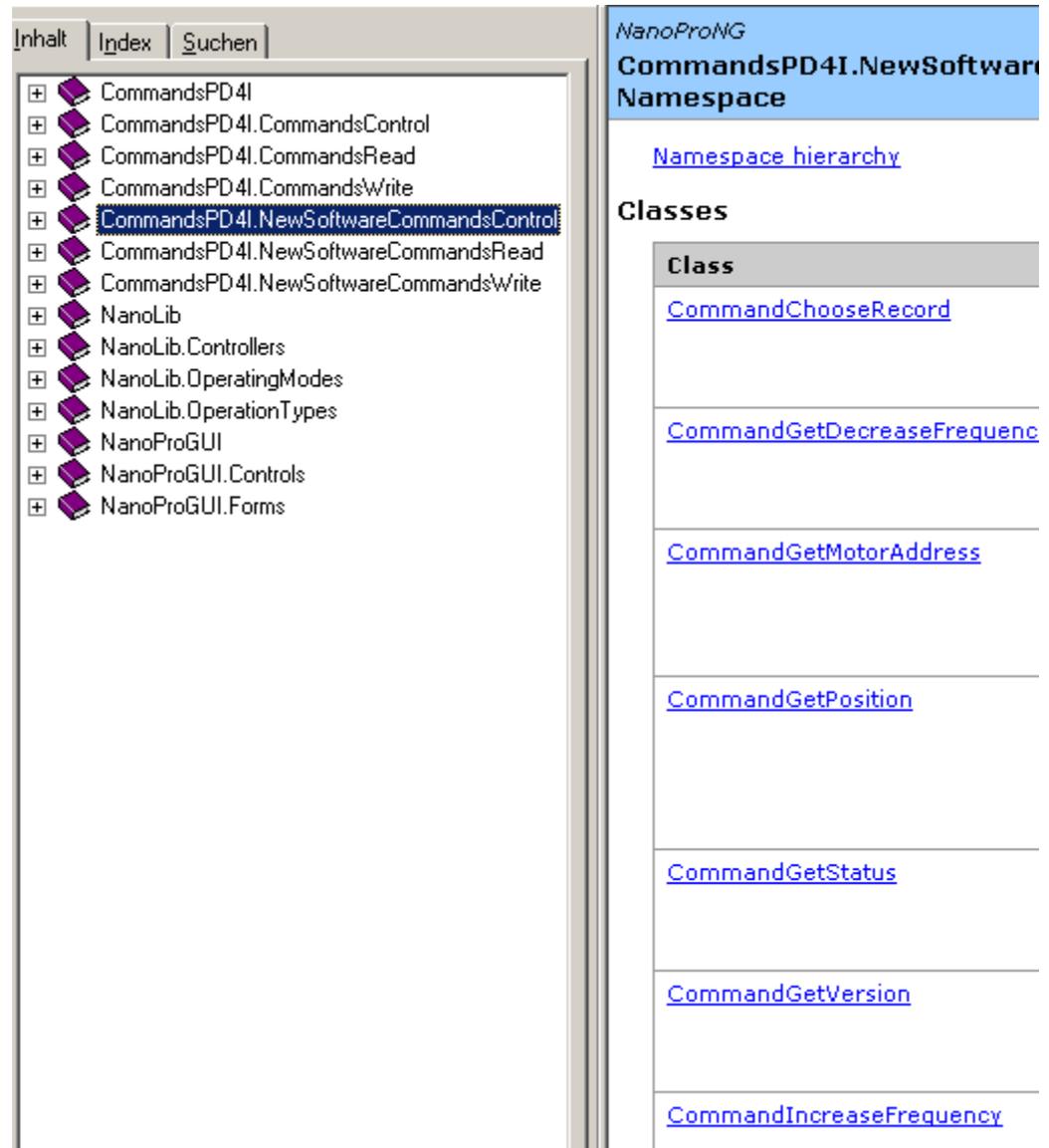
Associated VBA script



Windows help for DLL library

The commands in the DLL library are individually documented in a Windows help file, where they are arranged by class.

Overview of the CommandsPD4I.NewSoftwareCommandsControl Classes



The screenshot shows a Windows help window for the DLL library. The left pane displays a tree view of the namespace hierarchy, with **CommandsPD4I.NewSoftwareCommandsControl** selected. The right pane shows the **CommandsPD4I.NewSoftwareCommandsControl Namespace** and lists the following classes:

- [CommandChooseRecord](#)
- [CommandGetDecreaseFrequency](#)
- [CommandGetMotorAddress](#)
- [CommandGetPosition](#)
- [CommandGetStatus](#)
- [CommandGetVersion](#)
- [CommandIncreaseFrequency](#)

Overview of the CommandChooseRecord Class

Inhalt | Index | Suchen

- [-] CommandsPD4I
- [-] CommandsPD4I.CommandsControl
- [-] CommandsPD4I.CommandsRead
- [-] CommandsPD4I.CommandsWrite
- [-] CommandsPD4I.NewSoftwareCommandsControl
 - [-] **CommandChooseRecord Class**
 - [-] CommandGetDecreaseFrequency Class
 - [-] CommandGetMotorAddress Class
 - [-] CommandGetPosition Class
 - [-] CommandGetStatus Class
 - [-] CommandGetVersion Class
 - [-] CommandIncreaseFrequency Class
 - [-] CommandResetAllSettings Class
 - [-] CommandResetCounter Class
 - [-] CommandResetPosion Class
 - [-] CommandResetPositionError Class
 - [-] CommandSetRecord Class
 - [-] CommandStartTravelProfile Class
 - [-] CommandStopTravelProfile Class
 - [-] CommandTriggerOn Class
 - [-] CommandVersion Class
- [-] CommandsPD4I.NewSoftwareCommandsRead
- [-] CommandsPD4I.NewSoftwareCommandsWrite
- [-] NanoLib
- [-] NanoLib.Controllers
- [-] NanoLib.OperatingModes
- [-] NanoLib.OperationTypes

NanoProNG

CommandChooseRecord Class

Stellt die Steuerungsfunktion für Auswählen eines Satzes bereit.

For a list of all members of this [CommandChooseRecord Member](#) [System.Object](#) [BaseCommand](#) [BaseSetCommand](#)

CommandChooseRecord

```
public class CommandChooseRecord : BaseSetCommand
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [CommandsPD4I.NewSoftwareCommandsControl](#)

Assembly: CommandsPD4I (in CommandsPD4I.dll)

See Also

[CommandChooseRecord Member](#)
[CommandsPD4I.NewSoftwareCommandsControl Namespace](#)

Overview of the CommandChooseRecord Members

Inhalt | Index | Suchen

- [-] CommandsPD4I
- [-] CommandsPD4I.CommandsControl
- [-] CommandsPD4I.CommandsRead
- [-] CommandsPD4I.CommandsWrite
- [-] CommandsPD4I.NewSoftwareCommandsControl
 - [-] CommandChooseRecord Class
 - [-] **CommandChooseRecord Members**
 - [-] CommandChooseRecord Constructors
 - [-] CommandGetDecreaseFrequency Class
 - [-] CommandGetMotorAddress Class
 - [-] CommandGetPosition Class
 - [-] CommandGetStatus Class
 - [-] CommandGetVersion Class
 - [-] CommandIncreaseFrequency Class
 - [-] CommandResetAllSettings Class
 - [-] CommandResetCounter Class
 - [-] CommandResetPosion Class
 - [-] CommandResetPositionError Class
 - [-] CommandSetRecord Class
 - [-] CommandStartTravelProfile Class
 - [-] CommandStopTravelProfile Class
 - [-] CommandTriggerOn Class

NanoProNG

CommandChooseRecord Members

[CommandChooseRecord overview](#)

Public Instance Constructors

CommandChooseRecord Constructor	Initialisiert den CommandChooseRecord
---	---------------------------------------

See Also

[CommandChooseRecord Class](#) | [CommandsPD4I.NewSoftwareCommandsControl Namespace](#)

2 Read command

Function

A series of settings that can be set with a specific command can be read out with a corresponding read command.

This read command is composed of the 'Z' character and the command for the corresponding parameter.

Example

Reading out the travel distance: "#1Zs\r" -> "1Zs1000\r"

3 Records

Save travel distances

The firmware supports saving of travel distances in records. These data are saved in an EEPROM and are thus retained even if the device is switched off.

The EEPROM can accommodate 32 records with record numbers 1 to 32.

Saved settings per record

The following settings are saved in every record:

- 5.1 Set the positioning mode 'p': Positioning mode
- 5.2 Set the travel distance 's': Travel distance
- 5.3 Set the minimum frequency 'u': Initial step frequency
- 5.4 Set the maximum frequency 'o': Maximum step frequency
- 5.5 Set the maximum frequency 2 'n': Second maximum step frequency
- 5.6 Set the ramp 'b': Acceleration and braking ramp
- 5.7 Set the direction 'd': Rotation direction
- 5.8 Set the change of direction 't': Change in rotation direction in repeat records
- 5.9 Set the repetitions 'w': Repetitions
- 5.10 Set the record pause 'P': Pause between repetitions and continuation records
- 5.11 Set the continuation record 'N': Record number of the continuation record

4 General commands

4.1 Start the bootloader

Parameters

Character	Parameters
'@A'	None

Firmware response

No response, bootloader responds with '@OK'

Description

The command instructs the firmware to launch the bootloader. The firmware itself does not respond to the command. The bootloader responds with '@OK'.

The bootloader itself also requires this command to prevent it from automatically terminating itself after one half second. Therefore, this command needs to be sent repeatedly until the bootloader responds with '@OK'. The bootloader uses the same addressing scheme as the firmware itself.

4.2 Joystick mode dead range

Parameters

Character	Parameters
'='	Integer between 0 and 100

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the dead range in joystick mode.

In joystick mode, the motor can be moved forward and backward via a voltage on the analogue input.

The value range halfway between the maximum and minimum voltages in which the motor does not rotate is the dead range. It is specified as a percentage of the range width.

Reading out

Command 'Z=' is used to read out the current setting of the dead range.

4.3 Read out the error memory

Parameters

Character	Parameters
'E'	None

Firmware response

Returns the index of the error memory with the last error that occurred.

Description

The firmware contains 32 error memory locations. This command returns the index of the memory location with the last error that occurred.

Reading out

Command 'Z'+Index'+E' is used to read out the error number of the respective error memory.

4.4 Read out the encoder position

Parameters

Character	Parameters
'T'	None

Firmware response

Returns the current position of the motor according to the encoder.

Description

In motors with an encoder, this command returns the current position of the motor in motor steps as indicated by the encoder. As long as the motor has not lost steps, the values of the command *4.15 Read out the position 'C'* and the command *4.41 Read out the current record 'I' (pipe)* match.

However, it should be noted that the encoder has a resolution that is too low for step modes greater than 1/10 in 1.8° motors and 1/5 in 0.9° motors, and differences will therefore still arise between the two values specified above.

4.5 Set the error correction mode

Parameters

Character	Parameters
'U'	Integer, allowed values are 0, 1 and 2

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the error correction mode.

Parameter 0: Off

Parameter 1: Correction after travel

Parameter 2: Correction during travel (not implemented)

In a motor without an encoder, this value must be explicitly set to 0; otherwise, it will continuously attempt to make a correction because it assumes that there are step losses.

Reading out

Command 'ZE'+Index is used to read out the error number of the respective error memory.

4.6 Set the filter for the analogue and joystick modes

Parameters

Character	Parameters
'f'	Integer, allowed values are from 0 to 16

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

In the analogue and joystick modes, the analogue input is used to set the speed. Command 'f' is used to set the number of samples averaged to determine the final value.

Reading out

Command 'Zf' is used to read out the current setting of the value.

4.7 Set the step angle

Parameters

Character	Parameters
'a'	Integer, allowed values are 9 and 18

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

To convert the encoder position to the rotor position, the firmware requires information on the step angle of the motor. The value must be set to 9 for 0.9° motors and to 18 for 1.8° motors. Other values are not supported.

Reading out

Command 'Za' is used to read out the current setting of the value.

4.8 Read out the firmware version

Parameters

Character	Parameters
'v'	None

Firmware response

Returns the version string of the firmware.

Description

The return sting consists of several blocks:

'v' echo of the command

' ' separator (space)

Hardware: 'PD4','PD4lc','PD2lc','SMCI32','SMCI47' are possible versions

'_' separator

Communication: 'USB' or 'RS485'

'_' separator

Release date: dd-mm-yyyy, e.g. 26-09-2007

Example of a complete response

```
"001v PD4_RS485_26-09-2007\r"
```

4.9 Mask the inputs

Parameters

Character	Parameters
'L'	Bit mask as integer

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored (i.e. the entire mask is discarded).

Description

Valid for firmware version 09-11-2007 and higher.

This bit mask has 32 bits.

Sets a bit mask that permits the user to use the inputs and outputs. If the bit of the corresponding I/Os is set to '1', the firmware uses these I/Os. If it is set to '0', the I/Os can be used by the user. See also command *4.10 Read out the inputs and outputs 'Y'*.

The bit assignment is as follows:

Bit0: Input 1

Bit1: Input 2

Bit2: Input 3

Bit3: Input 4

Bit4: Input 5

Bit5: Input 6

Bit16: Output 1

Bit17: Output 2

All other bits are '0'

In the PDx-N variants, only input 1 and input 6 can be used.

If invalid bit masks are used, these are discarded, even if the firmware confirms them correctly.

Reading out

Command 'ZL' is used to read out the current setting of the mask.

4.10 Read out the inputs and outputs

Parameters

Character	Parameters
'Y'	Bit mask as integer

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Valid for firmware version 09-11-2007 and higher.

This bit mask has 32 bits.

Sets the firmware outputs if they are masked for free use by means of the command *4.9 Mask the inputs 'L'*.

Output 1 corresponds to bit 16 and output 2 to bit 17.

Reading out

Command 'ZY' is used to read out the current setting of the value.

The status of the inputs is displayed as well.

Bit0: Input 1

Bit1: Input 2

Bit2: Input 3

Bit3: Input 4

Bit4: Input 5

Bit5: Input 6

Bit6: '0' when the encoder is at the index line, otherwise '1'

Bit 16: Output 1 (as set by the user, even if the firmware is currently using it)

Bit 17: Output 2 (as set by the user, even if the firmware is currently using it)

All other bits are '0'.

Low cost variants only have inputs 1 (Bit0) and 2 (Bit5).

4.11 Read out the firmware version (old)

Parameters

Character	Parameters
' ' (space)	None

Firmware response

String containing firmware version (const, since new command 'v' has assumed this function).

Description

Required for bootloader; otherwise, this command serves no purpose.

4.12 Read out the status

Parameters

Character	Parameters
'\$'	None

Firmware response

Returns the status of the firmware as a bit mask.

Description

The bit mask has 8 bits.

Bit 0: 1: Driver ready

Bit 1: 1: Zero position reached

Bit 2: 1: Position error

Bit 3: 1: Input 1 is set while the driver is ready again. This occurs when the driver is started via input 1 and the driver is ready before the input has been reset.

Bits 4 through 6 specify the current mode as an integer:

0: Unused

1: Driver in positioning mode

2: Driver in speed mode

3: Driver in flag positioning mode

4: Driver in clock direction mode

5: Analogue mode

6: Joystick mode

7: Unused

Bit 7 is unassigned

4.13 Read out the motor address

Parameters

Character	Parameters
'M'	None

Firmware response

Returns the motor address.

Description

Returns the serial address. In particular, this is useful in connection with the '*' addressing type if the motor address is not known.

4.14 Read out a parameter

Parameters

Character	Parameters
'Z'	Readable command and optional associated record number

Firmware response

Returns the required parameter.

Description

This is used to read out the current settings of the values of certain commands. For example, the travel distance is read out with 'Zs', to which the firmware responds with 'Zs1000'.

If the parameter of a specific record is to be read out, the number of the record must be placed in front of the respective command.

Example: 'Z5s' -> 'Z5s2000'

A list of record commands can be found under "2 Read command"

4.15 Read out the position

Parameters

Character	Parameters
'C'	None

Firmware response

Returns the current position.

Description

Returns the current position of the motor in steps of the set step mode. This position is relative to the position of the last reference run.

If the motor is equipped with an angle transmitter, this value should be very close to the value of command "I" with a very low tolerance.

The tolerance depends on the step mode and the motor type (0.9° or 1.8°) since the angle transmitter has a lower resolution than the motor in the microstep mode.

The value range is from -100000000 through +100000000.

4.16 Set the motor address

Parameters

Character	Parameters
'm'	Integer, allowed values between 1 and 254

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the motor address. Ensure that the newly set address is not already occupied by another motor as this would make communication impossible.

Addresses 0 and 255 are reserved for faults of the EEPROM.

Reading out

Command 'Zm' is used to read out the current address. See also command [4.13 Read out the motor address 'M'](#).

4.17 Set the step mode

Parameters

Character	Parameters
'g'	Integer, allowed values: 1, 2, 4, 5, 8, 10, 16, 32, 64 and 255

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the step mode. The passed number equals the number of microsteps per full step, with the exception of the value 255, which selects the adaptive step mode.

Reading out

Command 'Zg' is used to read out the current valid value.

4.18 Set the phase current

Parameters

Character	Parameters
'i'	Integer, allowed values between 0 and 150

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the phase current in percent. Values above 100 should be avoided.

Reading out

Command 'Zi' is used to read out the current valid value.

4.19 Set the phase current at standstill

Parameters

Character	Parameters
'r'	Integer, allowed values between 0 and 150

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the current of the current reduction in percent. Like the phase current, this current is relative to the end value and not relative to the phase current. Values above 100 should be avoided.

Reading out

Command 'Zr' is used to read out the current valid value.

4.20 Set the motor mode

Parameters

Character	Parameters
'!	Integer, allowed values between 1 and 6

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the motor mode. Six different modes are available:

- 1: Positioning mode
- 2: Speed mode
- 3: Flag positioning mode
- 4: Clock directional mode
- 5: Analogue modus
- 6: Joystick modus

Reading out

Command 'Z!' is used to read out the current valid value.

4.21 Set the limit switch behaviour

Parameters

Character	Parameters
'I'	Integer, bit mask; the values are provided in the description

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the limit switch behaviour. The integer parameter is interpreted as a bit mask. The bit mask has 16 bits.

Free travel means that when the switch is reached, the driver travels away from the switch at the set lower speed.

Stop means that when the switch is reached, the driver stops immediately. The switch remains pressed.

Behaviour of the internal limit switch during a reference run:

Bit0: Free travel forwards

Bit1: Free travel backwards

Exactly one of the two bits must be set.

Behaviour of the internal limit switch during a normal run:

Bit2: Free travel forwards

Bit3: Free travel backwards

Bit4: Stop

Bit5: Disable

Exactly one of the four bits must be set.

This setting is useful when the motor is not allowed to turn more than one rotation.

Behaviour of the external limit switch during a reference run:

Bit9: Free travel forwards

Bit10: Free travel backwards

Exactly one of the two bits must be set.

Behaviour of the external limit switch during a normal run:

Bit11: Free travel forwards

Bit12: Free travel backwards

Bit13: Stop

Bit14: Disable

Exactly one of the four bits must be set.

With this setting, the travel distance of the motor can be precisely limited by a limit switch.

Reading out

Command 'ZI' is used to read out the current valid value.

4.22 Set the limit switch type

Parameters

Character	Parameters
'e'	Integer, allowed values are 0 and 1

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the type of limit switch.

'0' indicates an opener, '1' indicates a closer.

This parameter is used to indicate to the firmware in what state it sees the external limit switch as activated. The limit switch is connected between the supply voltage (to +5V in SMC1xx) and input 6.

Therefore, 'opener' means that under normal conditions, a high level is applied at the input since the switch is normally closed. When the switch is activated, it opens this contact ("opener") and there is no voltage at the input.

Reading out

Command 'Ze' is used to read out the current valid value.

4.23 Save a record

Parameters

Character	Parameters
'>'	Integer, allowed values between 1 and 32

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

This command is used to save the currently set commands (in RAM) in a record in the EEPROM. The parameter is the record number in which the data are saved.

This command should not be called up during a run because the current values change during subsequent runs.

The following commands belong to a record:

- 5.1 Set the positioning mode 'p'
- 5.2 Set the travel distance 's'
- 5.3 Set the minimum frequency 'u'
- 5.4 Set the maximum frequency 'o'
- 5.5 Set the maximum frequency 2 'n'
- 5.6 Set the ramp 'b'
- 5.7 Set the direction 'd'
- 5.8 Set the change of direction 't'
- 5.9 Set the repetitions 'w'
- 5.10 Set the record pause 'P'
- 5.11 Set the continuation record 'N'

4.24 Set the reverse clearance

Parameters

Character	Parameters
'z'	Integer, allowed values between 0 and 9999

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the reverse clearance in steps.

This setting is used to compensate for the clearance of downstream gears when there is a change in direction.

When there is a change in direction, the motor takes the number of steps set in the parameter before it begins incrementing the position.

Reading out

Command 'Zz' is used to read out the current valid value.

4.25 Reset the position

Parameters

Character	Parameters
'c'	None

Firmware response

Confirms the command through an echo.

Description

Resets the position of the motor to 0.

The current position of the motor is then used as the reference position.

4.26 Set automatic sending of the status

Parameters

Character	Parameters
'J'	Integer, allowed values are 0 and 1

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

If this parameter is set to '1', the firmware independently sends the status after the end of a run. See command *4.12 Read out the status '\$'*, which differs only in that a small 'j' is sent instead of a '\$'.

Reading out

Command 'ZJ' is used to read out the current valid value.

4.27 Set the record for auto correction

Parameters

Character	Parameters
'F'	Integer, allowed values between 1 and 32

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the record used for the correction run.

See command *4.5 Set the error correction mode 'U'*.

Reading out

Command 'ZF' is used to read out the current valid value.

4.28 Set the encoder direction

Parameters

Character	Parameters
'q'	Integer, allowed values are 0 and 1

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

If the parameter is set to '1', the direction of the rotary encoder is reversed.

Reading out

Command 'Zq' is used to read out the current valid value.

4.29 Set the swing out time

Parameters

Character	Parameters
'O'	Integer, allowed values between 0 and 255

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Defines the swing out time in 10ms steps between the end of the run and when the position is checked by the encoder.

This parameter is only valid for the positional check after a run.
 See command *4.5 Set the error correction mode 'U'*.

The position is only checked between repetitions and continuation records if the pause time (see command *5.10 Set the record pause 'P'*) is longer than the swing out time.

After a record, the swing out time is awaited before the motor indicates that it is ready again.

Reading out

Command 'ZO' is used to read out the current valid value.

4.30 Set the maximum encoder deviation

Parameters

Character	Parameters
'X'	Integer, allowed values between 0 and 100

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the maximum deviation in steps between the setpoint position and the encoder position.

In step modes greater than 1/10 step in 1.8° and 1/5 step in 0.9° motors, this value must be greater than 0 since, even then, the encoder has a lower resolution than the microsteps of the motor.

Reading out

Command 'ZX' is used to read out the current valid value.

4.31 Set the minimum voltage for the analogue mode

Parameters

Character	Parameters
'Q'	Integer, allowed values between -100 and 100

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the beginning of the range of the analogue input in 0.1V steps.

Reading out

Command 'ZQ' is used to read out the current valid value.

4.32 Set the maximum voltage for the analogue mode

Parameters

Character	Parameters
'R'	Integer, allowed values between -100 and 100

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the end of the range of the analogue input in 0.1V steps.

Reading out

Command 'ZR' is used to read out the current valid value.

4.33 Set the dead range for the joystick mode

Parameters

Character	Parameters
'%'	Integer, allowed values between 0 and 100

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the dead range of the analogue input as a percentage of the range set for the joystick mode.

Reading out

Command 'Z%' is used to read out the current valid value.

4.34 Start a record

Parameters

Character	Parameters
'A'	None

Firmware response

Confirms the command through an echo.

Description

Starts the run with the current parameter settings.

4.35 Stop a record

Parameters

Character	Parameters
'S'	None

Firmware response

Confirms the command through an echo.

Description

Stops the current run.

In the speed, analogue and joystick modes, this is the only method of returning the motor to the ready state.

The motor is brought to an immediate halt without ramps. This may result in step loss at high speeds.

For this reason, in the three modes named above, the speed should be lowered before issuing the stop command.

4.36 Increase the speed

Parameters

Character	Parameters
'+'	None

Firmware response

Confirms the command through an echo.

Description

Increases the speed in the speed mode by 100 steps/s.

4.37 Reduce the speed

Parameters

Character	Parameters
'-'	None

Firmware response

Confirms the command through an echo.

Description

Decreases the speed in the speed mode by 100 steps/s.

4.38 Triggering

Parameters

Character	Parameters
'T'	None

Firmware response

Confirms the command through an echo.

Description

Trigger for the flag positioning mode.

Before triggering, the motor travels at a constant speed.

After triggering, the motor finishes travelling the set distance from the position where triggering occurred, and then stops.

4.39 Reset a position error

Parameters

Character	Parameters
'D'	None

Firmware response

Confirms the command through an echo.

Description

Resets an error in the speed monitoring and sets the current position to the position indicated by the encoder.

4.40 Load a record from the EEPROM

Parameters

Character	Parameters
'y'	Integer from 1 to 32

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Loads the record data of the record passed in the parameter from the EEPROM.

See also command [4.23 Save a record '>'](#).

4.41 Read out the current record

Parameters

Character	Parameters
' ' (pipe)	Integer, allowed values are 0 and 1

Firmware response

Confirms the command through an echo when the parameter is set to '1'. This is the only response.

Description

Implemented in firmware version 09-11-2007 and higher.

If the parameter is set to '0', the firmware does not respond at all to commands, although it continues to execute them as before. This can be used to quickly send settings to the firmware without awaiting a response.

Reading out

With command 'Z|', the firmware sends all settings of the loaded record together.

With 'Z5|', the data of set 5 in the EEPROM are sent.

The format corresponds to that of the respective commands.

It should be noted that the '|' character is not sent with the response. See the following examples.

Examples

```
#1Z|\r'
```

```
-> 'Zp+1s+1u+400o+860n+1000b+55800d+1t+0W+1P+0N+0\r'
```

```
#1Z5|\r'
```

```
-> 'Z5p+1s+400u+400o+1000n+1000b+2364d+0t+0W+1P+0N+0\r'
```

5 Record commands

5.1 Set the positioning mode

Parameters

Character	Parameters
'p'	Integer, allowed values between 1 and 4 (depending on the motor mode)

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

In each of the 6 different motor modes, this command has a different meaning:

Positioning mode (!=1)

p=1

Relative positioning;

The command 5.2 *Set the travel distance 's'* specifies the travel distance relative to the current position.

The command 5.7 *Set the direction 'd'* specifies the direction.

The parameter 5.2 *Set the travel distance 's'* must be positive.

p=2

Absolute positioning;

The command 5.2 *Set the travel distance 's'* specifies the target position relative to the reference position.

The command 5.7 *Set the direction 'd'* is ignored.

p=3

Internal reference run;

The motor runs at the lower speed in the direction set in command 5.7 *Set the direction 'd'* until it reaches the index line of the encoder. Then the motor runs a fixed number of steps to leave the index line again. For the direction of the free run, see command 4.21 *Set the limit switch behaviour 'l'*. This mode is only useful for motors with integrated and connected encoders.

p=4

External reference run;

The motor runs at the upper speed in the direction set in command 5.7 *Set the direction 'd'* until it reaches the limit switch. Then a free run is performed, depending on the setting.

See command 4.21 *Set the limit switch behaviour 'l'*.

Speed mode (!=2)

p=1

Speed mode;

When the motor is started, the motor increases in speed to the maximum speed with the set ramp. Changes in the speed or direction of rotation are immediately performed with the set ramp without having to stop the motor first.

p=2

Not assigned

p=3

Internal reference run;

see Positioning mode

p=4
External reference run;
see Positioning mode

Flag positioning mode (!=3)

p=1
Flag positioning mode;
After starting, the motor runs up to the maximum speed. After the trigger event arrives (command 4.38 *Triggering 'T'* or trigger input), the motor finishes running the set travel distance (command 5.2 *Set the travel distance 's'*) and alters its speed to maximum speed 2 (command 5.5 *Set the maximum frequency 2 'n'*).

p=2
Not assigned

p=3
Internal reference run;
see Positioning mode

p=4
External reference run;
see Positioning mode

Clock direction mode (!=4)

p=1
Auto mode;
The motor takes 10 single steps and then increases its speed to the value set for the maximum speed until the enable is disabled again.

p=2
Not assigned

p=3
Internal reference run;
see Positioning mode

p=4
External reference run;
see Positioning mode

Analogue mode (!=5)

Not applicable

Joystick mode (!=6)

Not applicable

Reading out

Command 'Z!' is used to read out the current valid value.

5.2 Set the travel distance

Parameters

Character	Parameters
's'	Integer

Firmware response

Confirms the command through an echo.

Description

This command specifies the travel distance in (micro-)steps. Only positive values are allowed for the relative positioning. The direction is set with command 5.7 *Set the direction 'd'*.

For absolute positioning, this command specifies the target position. Negative values are allowed in this case. The direction of rotation from command 5.7 *Set the direction 'd'* is ignored since it is derived from the current position and the target position.

The value range is from -100000000 through +100000000.

In the adaptive mode, this parameter refers to half steps.

Reading out

Command 'Zs' is used to read out the current valid value.

5.3 Set the minimum frequency

Parameters

Character	Parameters
'u'	Integer, allowed values between 60 and 25000 (a minimum frequency of 1 Hz is under development)

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the minimum speed in Hertz (steps per second).

When a record starts, the motor begins rotating with the minimum speed. It then runs up to the maximum speed (command 5.4 *Set the maximum frequency "o"*) with the set ramp (command 5.6 *Set the ramp "b"*).

Reading out

Command 'Zu' is used to read out the current valid value.

5.4 Set the maximum frequency

Parameters

Character	Parameters
'o'	Integer, allowed values between 60 and 25000 (a minimum frequency of 1 Hz is under development)

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the maximum speed in Hertz (steps per second).

The maximum speed is reached after first passing through the acceleration ramp.

Reading out

Command 'Zo' is used to read out the current valid value.

5.5 Set the maximum frequency 2

Parameters

Character	Parameters
'n'	Integer, allowed values between 60 and 25000 (a minimum frequency of 1 Hz is under development)

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the maximum speed 2 in Hertz (steps per second).

The maximum speed 2 is reached after first passing through the acceleration ramp.

This value is only applied in the flag positioning mode. See command 5.1 *Set the positioning mode 'p'*.

Reading out

Command 'Zn' is used to read out the current valid value.

5.6 Set the ramp

Parameters

Character	Parameters
'b'	Integer, allowed values between 1 and 65535

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the acceleration ramp (and at this time also the brake ramp).

To convert the parameter to acceleration in Hz/ms, the following formula is used:

Acceleration in Hz/ms = (3000.0 / sqrt((float)<parameter>)) - 11.7).

Reading out

Command 'Zb' is used to read out the current valid value.

5.7 Set the direction

Parameters

Character	Parameters
'd'	Integer, allowed values are 0 and 1

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Sets the rotation direction.

Reading out

Command 'Zd' is used to read out the current valid value.

5.8 Set the change of direction

Parameters

Character	Parameters
't'	Integer, allowed values are 0 and 1

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

With repetition records, the rotation direction of the motor is reversed with every repetition if this parameter is set to '1'. See command 5.9 *Set the repetitions 'w'*.

Reading out

Command 'Zt' is used to read out the current valid value.

5.9 Set the repetitions

Parameters

Character	Parameters
'W'	Integer, allowed values between 0 and 254

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the number of repetitions of the current record.

A value of 0 indicates an endless number of repetitions.

Normally, the value is set to 1 for one repetition.

Reading out

Command 'ZW' is used to read out the current valid value.

5.10 Set the record pause

Parameters

Character	Parameters
'P'	Integer, allowed values between 0 and 65535

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the pause between record repetitions or between a record and a continuation record in ms (milliseconds).

If a record does not have a continuation record or a repetition, the pause is not executed and the motor is ready again immediately after the end of the run.

Reading out

Command 'ZP' is used to read out the current valid value.

5.11 Set the continuation record

Parameters

Character	Parameters
'N'	Integer, allowed values between 0 and 32

Firmware response

Confirms the command through an echo (including invalid values).

Invalid values are ignored.

Description

Specifies the number of the continuation record. If the parameter is set to '0', a continuation record is not performed.

Reading out

Command 'ZN' is used to read out the current valid value.

Index

A

- Analogue mode, set.....35
- Automatic sending of the status, set27

B

- Bootloader, start12

C

- Change of direction, set.....39
- Clock direction mode, set35
- Command format, old5
- Continuation record, set40
- Current record, read out33

D

- Dead range for the joystick mode, set.....30
- Dead range, joystick mode12
- Direction, set.....38
- Driver command structure5
- Driver response5

E

- Encoder direction, set.....28
- Encoder position, read out.....13
- Error correction mode, set14
- Error memory, read out13

F

- Filter for analogue mode, set.....14
- Filter for joystick mode, set.....14
- Firmware version, reading out (old).....17
- Flag positioning mode, set.....35

I

- Inputs, mask16
- Inputs, read out.....17

J

- Joystick mode, set35

L

- Limit switch behaviour, set..... 23
- Limit switch type, set..... 24

M

- Maximum encoder deviation, set 29
- Maximum frequency 2, set..... 37
- Maximum frequency, set..... 37
- Maximum voltage for the analogue mode, set 30
- Minimum frequency, set..... 36
- Minimum voltage for the analogue mode, set29
- Motor address, read out..... 18
- Motor address, set 20
- Motor mode, set 22

O

- Outputs, read out 17

P

- Parameter, read out..... 19
- Phase current at standstill, set..... 21
- Phase current, set..... 21
- Position error, reset..... 32
- Position, read out 19
- Position, reset 26
- Positioning mode, set..... 34

R

- Ramp, set..... 38
- Read command..... 10
- Record for auto correction, set 27
- Record in EEPROM, load 32
- Record pause, set..... 40
- Record, save 25
- Record, start 30
- Record, stop..... 31
- Records..... 11
- Repetitions, set 39

Reverse clearance, set.....26

S

Save travel distances11

Speed mode, set34

Speed, increase.....31

Speed, reduce31

Status, read out18

Step mode, set..... 20

Swing out time, set..... 28

T

Travel distance, save..... 11

Travel distance, set..... 36

Triggering..... 32