Command Reference

Description of the Nanotec firmware serial commands

For SMCI33 / SMCI47-S drivers

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.com

Editorial/About this manual

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

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 stepper motors SMCI33 and SMCI47-S

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 the user's own motor drivers is excluded.

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

Version/Change overview

Version	Date	Changes
V1.0	2009-02-10	Created (firmware version 04.12.2008)

Contents

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 carriage return '\r', like the command itself. The address is returned as '001' and not as '1'.

Examples

Set the travel distance of driver 1:	"#1s1000\r" -> "001s1000\r"
Start a record:	"#1A\r" -> "001A\r"
Invalid command:	"#1/\r" -> "001/?\r"

RS485 interface specification

19200 baud

8 bit

1 start bit

1 stop bit

No parity bit

CanOpen interface specification

Information on programming with CanOpen can be found in the corresponding manual for the interface under www.nanotec.com.

1.2 Long command format

Use

With the launch of the SMCI47-S driver, commands were introduced that consist of more than one character. These commands are used to read and change machine parameters. Because these generally only need to be set up during commissioning, the slower transmission rate due to the length of the command has no impact on operation.

Long command structure

A long command begins with the addressing scheme already described ("#<ID>"). This is followed by a colon that marks the beginning of the long command. Next comes the keyword and the command, followed by a carriage return character ("\r") that indicates the end of the command.

A long command can consist of the characters "A" to "Z" or "a" to "z" and the underscore ("_"). The syntax is case-sensitive. Digits are not allowed.

Keywords

The following keywords are defined for long commands:

:CL	For the controller settings and the motor settings (closed loop)
:brake	For the motor driver
:Capt	For the scope mode

Driver response

The firmware response does not begin with a "#" like the user request.

When the values are positive, the keyword is followed by a "+" sign. For negative values, a "-" sign is used.

Both signs ("+" and "-") can be used as separators.

If an unknown keyword is sent (unknown command), the firmware responds with a question mark after the colon

Unknown command	"#ID:CL_does_not_exist\r"
Firmware response	"ID:?\r"

Command for reading a parameter

Read command

To read a parameter, the end of the command name is terminated with a carriage return character.

Read command: "#ID:keyword_command_abc\r"

Firmware response

The firmware responds with an echo of the command and its value.

Response: "ID:keyword_command_abc+value\r"

Command for changing a parameter

Change command

To change a parameter, the command name is followed by a "=" character, followed by the value to be set. For positive values, a "+" sign is not mandatory but is also not disallowed. The command is terminated with a carriage return character.

Change command: "#ID:keyword_command_abc=value\r"

Firmware response

The firmware responds with an echo of the command as a confirmation.

Response: "ID:keyword_command_abc+value\r"

See also the following example.

Example

The structure of the long command is shown in the following example:

"Read out the motor pole pairs"

Reach parameter "#1:CL_motor_pp\r"

Firmware response "1:CL_motor_pp+50\r"

Change parameter "#1:CL_motor_pp=100\r"

Firmware response "1:CL_motor_pp100\r"

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

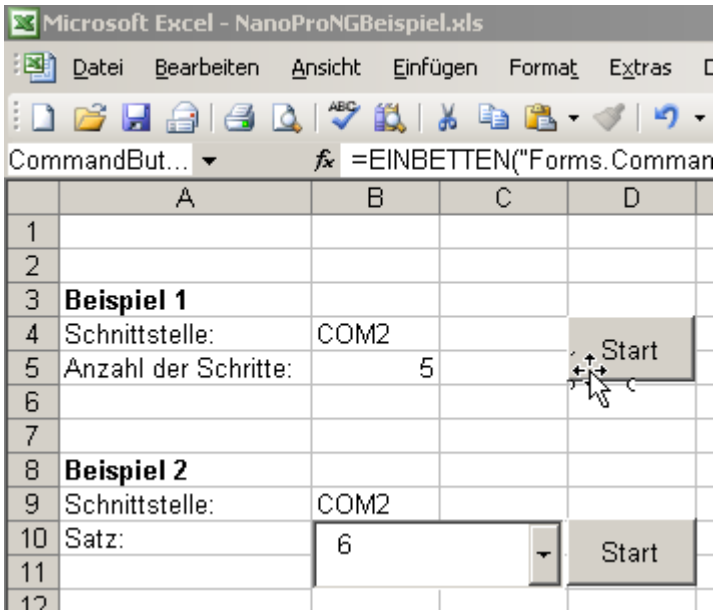
If you work under Windows with a .NET-capable programming language (e.g. Visual Basic of version 8 or higher, C#, Delphi.NET), we offer you the convenience of being able to integrate our DLL library in your application. You can conveniently control the functions of the driver via a function call without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

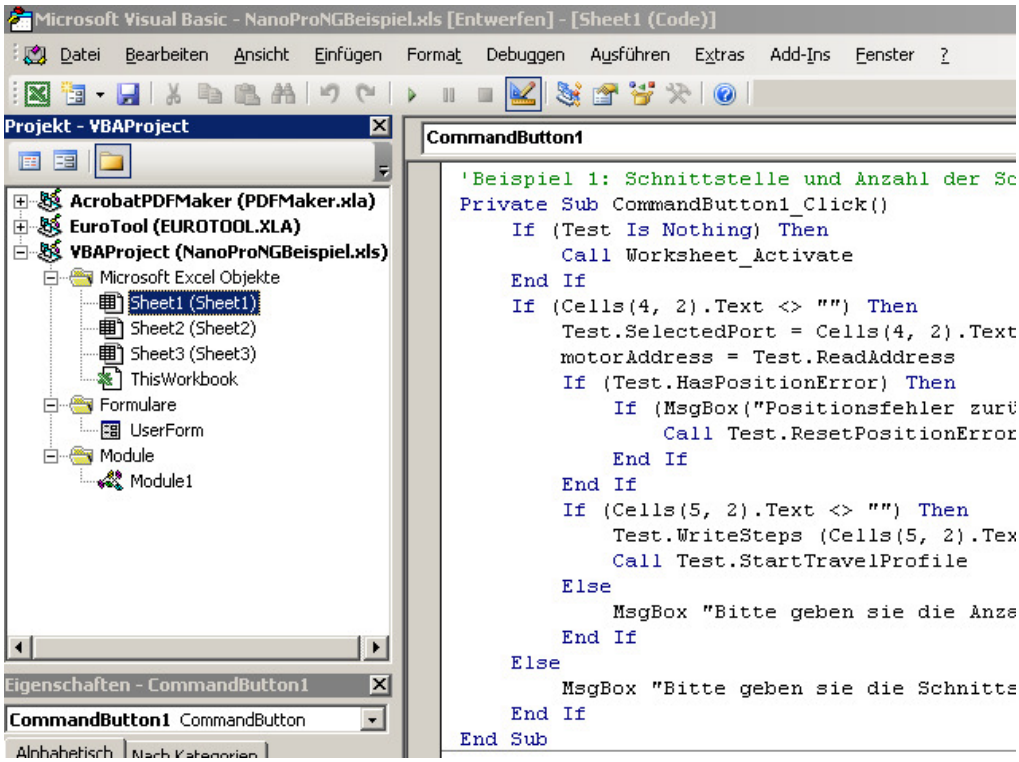
DLL library application

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



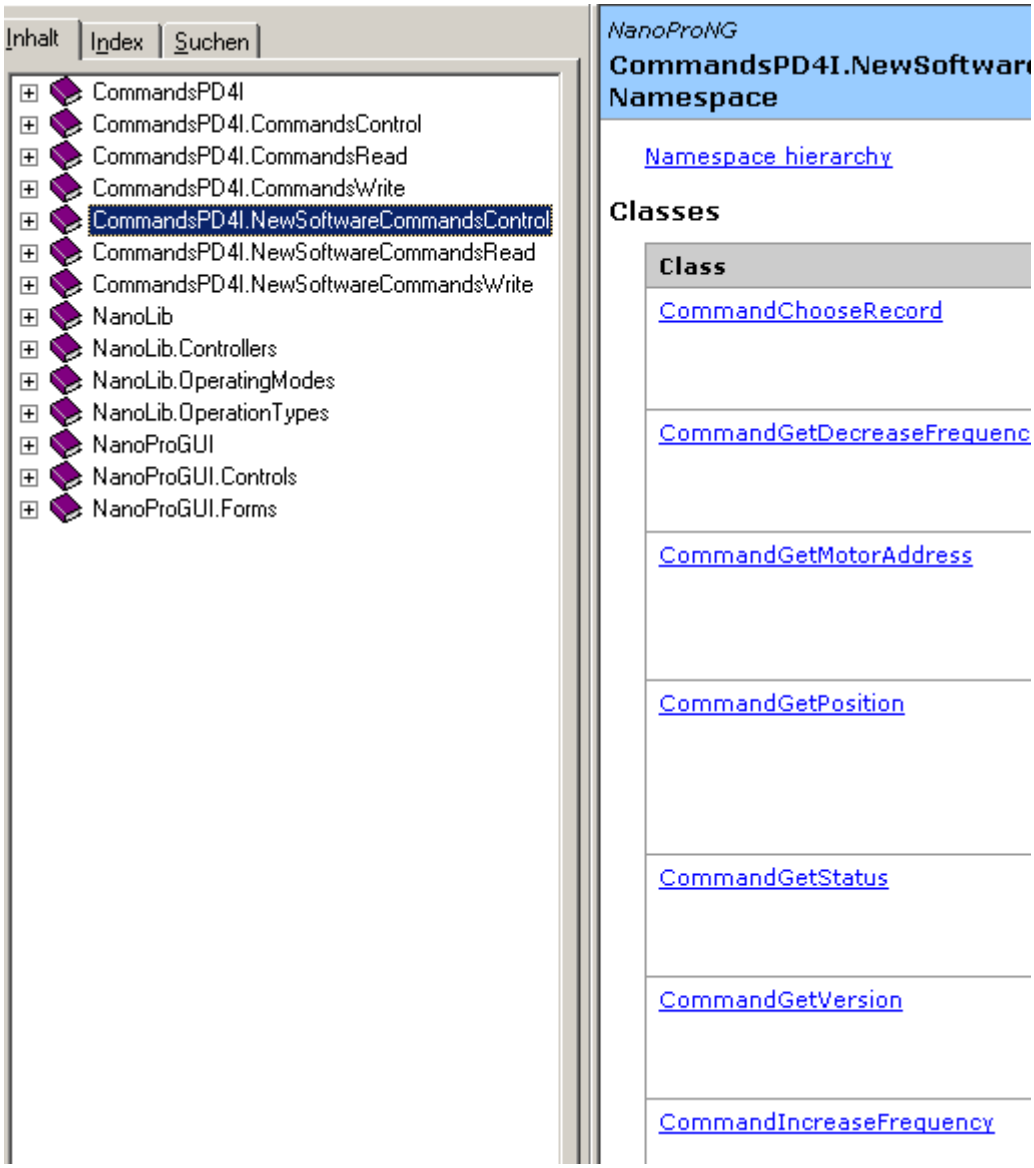
Associated VBA script



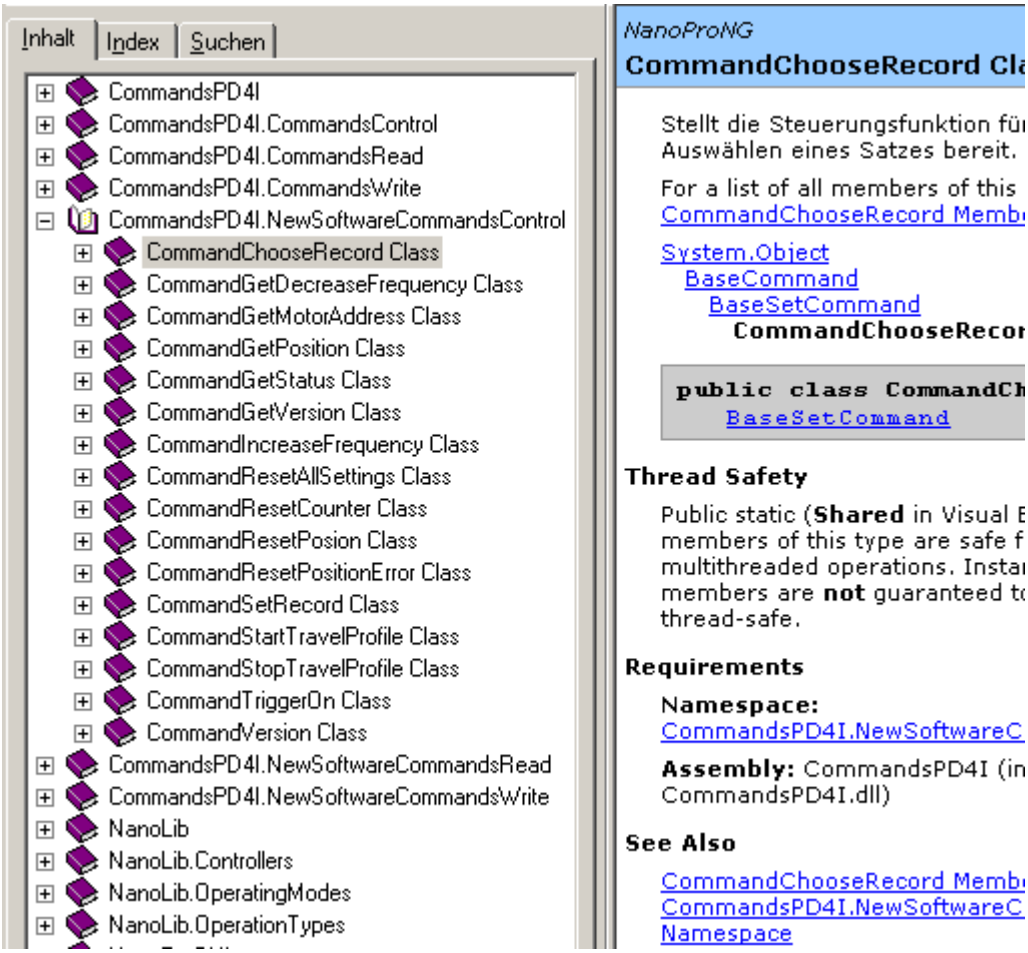
Windows help on the DLL library

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

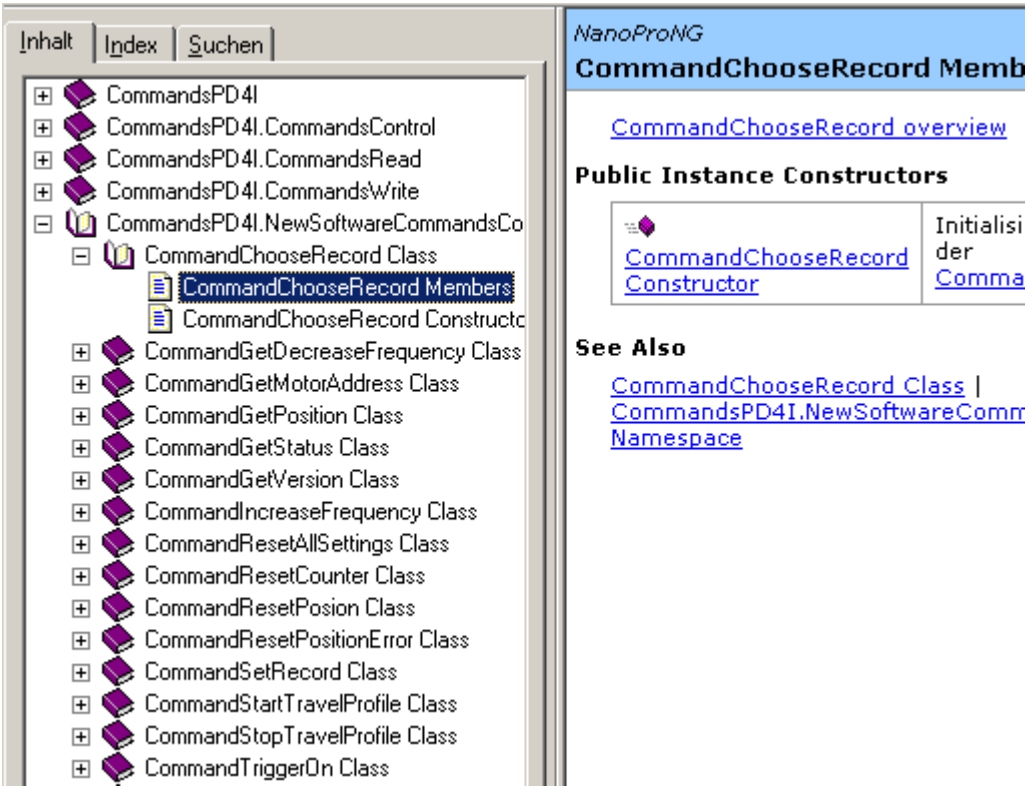
Overview of the CommandsPD4I.NewSoftwareCommandsControl Classes



Overview of the CommandsControl Classes



Overview of the CommandChooseRecord Members



2 Command overview

Overview of the commands

Below you will find an overview of all of the commands (characters and parameters):

3 Read command

Function

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

Command

Character	Parameter
'Z' + parameter '	The read command is composed of the 'Z' character and the command for the corresponding parameter.

Example

Read out the travel distance: "#1Zs\r" -> "001Zs1000\r"

4 Records

Saving 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:

Setting	Para-meter	See section	Page
Position mode	'p'	*6.6 Setting the* positioning mode	48
Travel distance	's'	*6.7 Setting the* travel distance	50
Initial step frequency	'u'	*6.8 Setting the* minimum frequency	50
Maximum step frequency	'o'	*6.9 Setting the* maximum frequency	51
Second maximum step frequency	'n'	*6.10 Setting the maximum* frequency 2	51
Acceleration and braking ramp	'b'	*6.11 Setting the* ramp	52
Direction of rotation	'd'	*6.12 Setting the* direction of rotation	52
Reversal in direction of rotation for repeat records	't'	*6.13 Setting the* change of direction	53
Repetitions	'w'	*6.14 Setting the* repetitions	53
Pause between repetitions and continuation records	'P'	*6.15 Setting the* record pause	54
Record number of continuation record	'N'	*6.16 Setting the* continuation record	54

5 General commands

5.1 Setting the phase current

Parameter

Character	Parameter
'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.

5.2 Setting the phase current at a standstill

Parameter

Character	Parameter
'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.

5.3 Setting the step mode

Parameter

Character	Parameter
'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.

5.4 Setting the motor address

Parameter

Character	Parameter
'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 *5.20 Reading out* the motor address *'M'*.

5.5 Setting the motor mode

Parameter

Character	Parameter
'!'	Integer, allowed values between 1 and 5

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 mode

6: Joystick mode

Reading out

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

5.6 Setting the limit switch behaviour

Parameter

Character	Parameter
'l'	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 revolution.

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 'Zl' is used to read out the current valid value.

5.7 Setting the limit switch type

Parameter

Character	Parameter
'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' means opener

'1' means 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 SMCIxx) 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.

5.8 Setting the step angle

Parameter

Character	Parameter
'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. A value of 9 must be set for 0.9° motors, and 18 must be set for 1.8° motors. Other values are not supported.

Reading out

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

5.9 Setting the error correction mode

Parameter

Character	Parameter
'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 'ZU'+Index is used to read out the error number of the respective error memory.

5.10 Setting the record for auto correction

Parameter

Character	Parameter
'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 *5.9 Setting the* error correction mode *'U'*.

Reading out

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

5.11 Setting the encoder direction

Parameter

Character	Parameter
'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.

5.12 Setting the settling time

Parameter

Character	Parameter
'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 settling 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 *5.9 Setting the* error correction mode *'U'*.

Between repetitions or continuation records, this position is only checked if the pause time (see command *6.15 Setting the* record pause *'P'*) is longer than the settling time.

After a record, the settling 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.

5.13 Setting the maximum encoder deviation

Parameter

Character	Parameter
'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 1/1 step in 10° and 1/1 step in 5° 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.

5.14 Resetting the position error

Parameter

Character	Parameter
'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.

5.15 Reading out the error memory

Parameter

Character	Parameter
'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.

The last 32 errors are stored. When memory location 32 is reached, the next error is again stored at memory position 1. In this case, memory position 2 contains the oldest error code that can be read out.

This command is used to read out the index of the memory space with the last error that occurred and the corresponding error code.

Reading out

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

Error codes

```
//! Error codes for error byte in EEPROM
#define ERROR_LOWVOLTAGE    0x01
#define ERROR_TEMP          0x02
#define ERROR_TMC           0x04
#define ERROR_EE            0x08
#define ERROR_QEI           0x10
#define ERROR_INTERNAL      0x20
```

Meaning

Error	Meaning
LOWVOLTAGE	Undervoltage
TMC	Driver module returned one error.
EE	Useless data in EPROM, e.g. step resolution is 25th of one step.
QEI	Position error
INTERNAL	Internal error (equivalent to the Windows blue screen).

Driver status

The status of the driver can be read out with the 5.21 Reading out the status *'$'* command.

5.16 Reading out the encoder position

Parameter

Character	Parameter
'I'	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. Provided that the motor has not lost any steps, the values of the *5.17 Reading out* the position *'C'* command and the *6.4 Reading out the* current record *'|'* (pipe) command are the same.

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

5.17 Reading out the position

Parameter

Character	Parameter
'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 that of a 32-bit signed integer (value range $\pm 2^{31}$).

5.18 Resetting the position

Parameter

Character	Parameter
'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.

5.19 Reading out the parameter

Parameter

Character	Parameter
'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 "4 Records"

5.20 Reading out the motor address

Parameter

Character	Parameter
'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.

5.21 Reading out the status

Parameter

Character	Parameter
'$'	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

5.22 Reading out the firmware version

Parameter

Character	Parameter
'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: d-mm-yyyy, e.g. 26-09-2007

Example of a complete response

"001v PD4_RS485_26-09-2007\r"

5.23 Reading out the firmware version (old)

Parameter

Character	Parameter
' ' (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.

5.24 Masking and demasking the inputs

Validity

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

Parameter

Character	Parameter
'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

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 are available to the user. See also command *5.30 Setting the* outputs *'Y'*.

The bit assignment is shown below:	Bit on 1:
Bit0: Input 1	1
Bit1: Input 2	2
Bit2: Input 3	4
Bit3: Input 4	8
Bit4: Input 5	16
Bit5: Input 6	32
Bit16: Output 1	65536
Bit17: Output 2	131072
All other bits are '0'	All on 1: 196671

Attention:
If a bit is not addressed when the mask is set, it is automatically set to '0', regardless of the state. All bits must be set at once.

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.

Examples

All bits should be set to '0'.

Send: #1L0\r

Read: 1L0\r

Bit3 and Bit5 should be set to '1':

Send: #1L20\r

Read: 1L20\r

'20' because Bit3 is addressed with the value of 4 and Bit5 with the value of 16, i.e. 4 + 16 = 20.

5.25 Reversing the polarity of the inputs and outputs

Validity

Valid for firmware version 30-01-2008 and higher.

Parameter

Character	Parameter
'h'	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

Sets a bit mask with which the user can reverse the polarity of the inputs and outputs. If the bit of the corresponding I/O is set to '1', there is no polarity reversal. If it is set to '0', the polarity of the I/O is inverted.

The bit assignment is shown below:

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'.

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

Reading out

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

5.26 Switching the interrupts of the inputs on and off

Validity

Valid for firmware version 30-01-2008 and higher.

Parameter

Character	Parameter
'k'	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

Sets a bit mask with which the user can switch the interrupts of the inputs on and off.

If the bit of the corresponding I/O is set to '1', the interrupt is switched on. Unless a different setting is made, an interrupt is initiated with every signal change (see also command *5.27 Setting the interrupts of the inputs to* a rising flank '/' and *5.28 Setting the interrupts of the inputs to* a falling flank '\'). The only exception is input 6, which responds either to a rising or falling flank. Unless a different setting is made, it only responds to the rising flank.

The bit assignment is shown below:

Bit0: Input 1

Bit1: Input 2

Bit2: Input 3

Bit3: Input 4

Bit4: Input 5

Bit5: Input 6

All other bits are '0'.

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

Reading out

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

5.27 Setting the interrupts of the inputs to a rising flank

Validity

Valid for firmware version 30-01-2008 and higher.

Parameter

Character	Parameter
'/'	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

Sets a bit mask with which the user can switch the interrupts of the inputs to rising flanks.

If the bit of the corresponding I/O is set to '1', the interrupt responds to a rising flank. If the interrupt of the corresponding I/O has not yet been switched on (see command *5.26 Switching the interrupts of the inputs* on and off 'k'), it is activated automatically.

The bit assignment is shown below:

Bit0: Input 1

Bit1: Input 2

Bit2: Input 3

Bit3: Input 4

Bit4: Input 5

Bit5: Input 6

All other bits are '0'.

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

Reading out

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

5.28 Setting the interrupts of the inputs to a falling flank

Validity

Valid for firmware version 30-01-2008 and higher.

Parameter

Character	Parameter
' \'	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

Sets a bit mask with which the user can switch the interrupts of the inputs to falling flanks.

If the bit of the corresponding I/O is set to '1', the interrupt responds to a falling flank. If the interrupt of the corresponding I/O has not yet been switched on (see command *5.26 Switching the interrupts of the inputs* on and off 'k'), it is activated automatically.

The bit assignment is shown below:

Bit0: Input 1

Bit1: Input 2

Bit2: Input 3

Bit3: Input 4

Bit4: Input 5

Bit5: Input 6

All other bits are '0'.

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

Reading out

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

5.29 Setting the debounce time for the inputs

Validity

Valid for firmware version 30-01-2008 and higher.

Parameter

Character	Parameter
'K'	Integer, allowed values are from 0 to 10

Firmware response

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

Invalid values are ignored.

Description

Sets the time in ms that needs to elapse after a signal change at an input until the signal has stabilized (so-called "debouncing").

Reading out

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

5.30 Setting the outputs

Validity

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

Parameter

Character	Parameter
'Y'	Bit mask as integer

Firmware response

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

Invalid values are ignored.

Description

This bit mask has 32 bits.

Sets the outputs of the firmware, provided that these have been masked for free use using command *5.24 Masking and demasking 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'.

5.31 Setting automatic sending of the status

Parameter

Character	Parameter
'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 *5.21 Reading out* the status *'$'*, with the difference that a lower case 'j' is sent instead of the '$'.

Reading out

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

5.32 Starting the bootloader

Parameter

Character	Parameter
'@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.

5.33 Setting the reverse clearance

Parameter

Character	Parameter
'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.

5.34 Setting the ramp in the positioning mode

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_ramp_mode'	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 ramp in the positioning mode ("!1"):

- "0" = The trapezoid ramp is selected
- "1" = The sinusoidal ramp is selected

Currently, this parameter only has an influence on the positioning mode.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

5.35 Setting the wait time for switching off the brake voltage

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':brake_ta'	Unsigned 16, value range 0 to 65535

Unit

ms

Firmware response

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

Invalid values are ignored.

Description

Sets the wait time in milliseconds between switching on of the motor current and switching off of the brake voltage.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

5.36 Setting the wait time for the motor movement

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':brake_tb'	Unsigned 16, value range 0 to 65535

Unit

ms

Firmware response

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

Invalid values are ignored.

Description

Sets the wait time in milliseconds between switching off of the brake voltage and enabling of a motor movement.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

5.37 Setting the wait time for switching off the motor current

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':brake_tc'	Unsigned 16, value range 0 to 65535

Unit

ms

Firmware response

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

Invalid values are ignored.

Description

Sets the wait time in milliseconds between switching on of the brake voltage and switching off of the motor current.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

6 Record commands

6.1 Starting a record

Parameter

Character	Parameter
'A'	None

Firmware response

Confirms the command through an echo.

Description

Starts the run with the current parameter settings.

6.2 Stopping a record

Parameter

Character	Parameter
'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.

In the three modes named above, the speed should therefore be reduced prior to the stop command.

6.3 Loading a record from the EEPROM

Parameter

Character	Parameter
'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 *6.5 Saving a* record *'>'*.

6.4 Reading out the current record

Validity

Implemented beginning with the next firmware version.

Parameter

Character	Parameter
'|' (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

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'

6.5 Saving a record

Parameter

Character	Parameter
'>'	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.

A record contains the following settings and commands:

Setting	Para-meter	See section	Page
Position mode	'p'	*6.6 Setting the* positioning mode	48
Travel distance	's'	*6.7 Setting the* travel distance	50
Initial step frequency	'u'	*6.8 Setting the* minimum frequency	50
Maximum step frequency	'o'	*6.9 Setting the* maximum frequency	51
Second maximum step frequency	'n'	*6.10 Setting the maximum* frequency 2	51
Acceleration and braking ramp	'b'	*6.11 Setting the* ramp	52
Direction of rotation	'd'	*6.12 Setting the* direction of rotation	52
Reversal in direction of rotation for repeat records	't'	*6.13 Setting the* change of direction	53
Repetitions	'w'	*6.14 Setting the* repetitions	53
Pause between repetitions and continuation records	'P'	*6.15 Setting the* record pause	54
Record number of continuation record	'N'	*6.16 Setting the* continuation record	54

6.6 Setting the positioning mode

Parameter

Character	Parameter
'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; Command *6.7 Setting the* travel distance *'s'* defines the travel distance relative to the current position. Command *6.12 Setting the* direction of rotation *'d'* defines the direction. Parameter *6.7 Setting the* travel distance 's' must be positive.
p=2	Absolute positioning; Command *6.7 Setting the* travel distance *'s'* defines the target position relative to the reference position. Command *6.12 Setting the* direction of rotation *'d'* is ignored.
p=3	Internal reference run; The motor runs at the lowest speed in the direction set in command *6.12 Setting the* direction of rotation *'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 free travel, see command *5.6 Setting 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 highest speed in the direction set in command *6.12 Setting the* direction of rotation *'d'* until it reaches the limit switch. Then a free run is performed, depending on the setting. See command *5.6 Setting 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 performed immediately with the set ramp without having to stop the motor first.
p=2	Not assigned
p=3	Internal reference run; see position mode
p=4	External reference run; see position mode

Flag positioning mode (!=3)	
p=1	Flag positioning mode; after starting, the motor runs up to the maximum speed. After the trigger event occurs (command *7.8 Actuating the* trigger '*T*' or trigger input), the motor moves the set travel distance (command *6.7 Setting the* travel distance '*s*') and, for this purpose, changes its speed to the maximum speed 2 (command *6.10 Setting the maximum* frequency 2 '*n*').
p=2	Not assigned
p=3	Internal reference run; see position mode
p=4	External reference run; see position 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 position mode
p=4	External reference run; see position mode
Analogue mode (!=5)	
	Not applicable
Joystick mode (!=6)	
	Not applicable

Reading out

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

6.7 Setting the travel distance

Parameter

Character	Parameter
'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 *6.12 Setting the* direction of rotation *'d'*.

For absolute positioning, this command specifies the target position. Negative values are allowed in this case. The direction of rotation from command *6.12 Setting the* direction of rotation '*d*' is ignored since it can be derived from the current position and the target position.

The value range is that of a 32-bit signed integer (value range ± 2^{31}).

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

Reading out

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

6.8 Setting the minimum frequency

Parameter

Character	Parameter
'u'	Integer, allowed values between 60 and 25000

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 accelerates with the set ramp (command *6.11 Setting the* ramp *'b'*) to the maximum speed (command *6.9 Setting the* maximum frequency *'o'*).

Reading out

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

6.9 Setting the maximum frequency

Parameter

Character	Parameter
'o'	Integer, allowed values between 60 and 25000

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.

6.10 Setting the maximum frequency 2

Parameter

Character	Parameter
'n'	Integer, allowed values between 60 and 25000

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 *6.6 Setting the positioning mode 'p'*.

Reading out

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

6.11 Setting the ramp

Parameter

Character	Parameter
'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.

6.12 Setting the direction of rotation

Parameter

Character	Parameter
'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 direction of rotation:

0: Left

1: Right

Reading out

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

6.13 Setting the change of direction

Parameter

Character	Parameter
'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 *6.14 Setting the* repetitions *'W'*.

Reading out

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

6.14 Setting the repetitions

Parameter

Character	Parameter
'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.

6.15 Setting the record pause

Parameter

Character	Parameter
'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.

6.16 Setting the continuation record

Parameter

Character	Parameter
'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.

7 Mode-specific commands

7.1 Setting the dead range for the joystick mode

Parameter

Character	Parameter
'='	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.

7.2 Setting the filter for the analogue and joystick modes

Parameter

Character	Parameter
'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.

7.3 Setting the minimum voltage for the analogue mode

Parameter

Character	Parameter
'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.

7.4 Setting the maximum voltage for the analogue mode

Parameter

Character	Parameter
'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.

7.5 Setting the dead range for the joystick mode

Parameter

Character	Parameter
'%'	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.

7.6 Increasing the speed

Parameter

Character	Parameter
'+'	None

Firmware response

Confirms the command through an echo.

Description

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

7.7 Reducing the speed

Parameter

Character	Parameter
'-'	None

Firmware response

Confirms the command through an echo.

Description

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

7.8 Actuating the trigger

Parameter

Character	Parameter
'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.

8 Closed loop settings

8.1 Activating the closed loop

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_enable'	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 value is set to "1", the firmware is instructed to activate the closed loop. This is only possible if a special reference run was performed since the unit was last switched on (mode 8 "!8").

Important conditions

The following conditions must be met when activating the closed loop:

- The "CL_Motor_pp", "CL_rotenc_inc" and "CL_rotenc_rev" settings must agree with the technical data of the connected stepper motor.
  See commands *8.6 Setting the motor pole* pairs, *8.7 Setting the number of* increments and *8.8 Setting the number of* revolutions.
- Every time a new motor is connected (even if it is the same type), a calibration run must be performed (mode 101 "!101").

ATTENTION:
If one of these conditions is not met, the motor may accelerate to a level that exceeds its maximum mechanical load capacity.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.2 Setting the tolerance window for the limit position

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_position_window'	Unsigned 32, value range 0 to 2^{32}-1

Unit

Increments

Firmware response

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

Invalid values are ignored.

Description

If the closed loop is active, this is a criterion for when the firmware considers the limit position to have been reached. The parameter defines a tolerance window in increments of the encoder.

If the position actually measured is within the desired limit position + – the tolerance that is set in this parameter, and if this condition is met over a certain period, the limit position is considered to have been reached.

The time for this time window is set in the "CL_position_window_time" parameter. See command *8.3 Setting the time for the tolerance* window of the limit position.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.3 Setting the time for the tolerance window of the limit position

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_position_window_time'	Unsigned 16, value range 0 to 65535

Unit

ms

Firmware response

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

Invalid values are ignored.

Description

Specifies the time in milliseconds for the "CL_position_window" parameter. See command *8.2 Setting the tolerance window* for the limit position.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.4 Setting the maximum allowed following error

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_following_error_window'	Unsigned 32, value range 0 to 2^{32}-1

Unit

Increments

Firmware response

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

Invalid values are ignored.

Description

If the closed loop is active, this parameter defines the maximum allowed following error in increments of the encoder.

If, at a certain point in time, the actual position differs from the setpoint position by more than this parameter, a position error is output and the closed loop is switched off.

In addition, the "CL_following_error_timeout" parameter can be used to specify for how long the following error may be larger than the tolerance without triggering a positioning error. See command *8.5 Setting the time for the* maximum following error.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.5 Setting the time for the maximum following error

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_following_error_timeout'	Unsigned 16, value range 0 to 65535

Unit

ms

Firmware response

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

Invalid values are ignored.

Description

This parameter can be used to specify in milliseconds for how long the following error may be larger than the tolerance without triggering a positioning error. See command *8.4 Setting the maximum allowed* following error.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.6 Setting the motor pole pairs

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_motor_pp'	Unsigned 16, current allowed values are 50 and 100

Unit

Pole pair number

Firmware response

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

Invalid values are ignored.

Description

The parameter sets the number of pole pairs of the connected motor.

Note:
After this parameter is changed, the firmware **must** be restarted (disconnect power).

The number of pole pairs equals 1/4 of the number of full steps per revolution. The adjustable values currently equal 50 and 100. If other values are set, this will result in the closed loop not functioning properly. However, even in this case, a conversion for the error correction without the closed loop will still function.

This parameter corresponds to command *5.8 Setting the* step angle 'a'.
If the "CL_motor_pp" or 'a' parameter is changed, the associated parameter is also changed.

The values are converted according to the following formula:

$$\frac{\text{CL\_motor\_pp} = 900}{\text{COMM\_CMD\_SETSTEPANGLE}}$$

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.7 Setting the number of increments

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_rotenc_inc'	Unsigned 32, current allowed values are 1600 and 2000

Unit

Increments

Firmware response

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

Invalid values are ignored.

Description

This parameter defines the number of increments of the encoder for a certain number of revolutions. The number of revolutions can be set using the "CL_rotenc_rev" parameter. See command *8.8 Setting the number of* revolutions.

Currently, the values 1600 and 2000 are supported for the closed loop. If other values are set, this will result in the closed loop not functioning properly. However, even in this case, a conversion for the error correction without the closed loop will still function.

Note:
After this parameter is changed, the firmware **must** be restarted (disconnect power).

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.8 Setting the number of revolutions

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_rotenc_rev'	Unsigned 32, allowed value = 1

Unit

Revolutions

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the number of revolutions for the "CL_rotenc_inc" parameter. See command *8.7 Setting the number of* increments.

This setting is available for compatibility reasons. It should always be set to "1". If other values are set, this will result in the closed loop not functioning properly. However, even in this case, a conversion for the error correction without the closed loop will still function.

Note:
After this parameter is changed, the firmware **must** be restarted (disconnect power).

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.9 Setting the numerator of the P component of the speed controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KP_v_Z'	Unsigned 16, value range 0 to 65535

Unit

Numerator

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the numerator of the proportional component of the speed controller.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.10 Setting the denominator of the P component of the speed controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KP_v_N'	Unsigned 8, value range 0 to 15

Unit

Denominator as powers of 2

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the denominator of the proportional component of the speed controller as a power of 2.

0 = 1

1 = 2

2 = 4

3 = 8

etc.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.11 Setting the numerator of the I component of the speed controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KI_v_Z'	Unsigned 16, value range 0 to 65535

Unit

Numerator

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the numerator of the integral component of the speed controller.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.12 Setting the denominator of the I component of the speed controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KI_v_N'	Unsigned 8, value range 0 to 15

Unit

Denominator as powers of 2

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the denominator of the integral component of the speed controller as a power of 2.

0 = 1

1 = 2

2 = 4

3 = 8

etc.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.13 Setting the numerator of the D component of the speed controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KD_v_Z'	Unsigned 16, value range 0 to 65535

Unit

Numerator

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the numerator of the differential component of the speed controller.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.14 Setting the denominator of the D component of the speed controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KD_v_Z'	Unsigned 8, value range 0 to 15

Unit

Denominator as powers of 2

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the denominator of the differential component of the speed controller as a power of 2.

0 = 1

1 = 2

2 = 4

3 = 8

etc.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.15 Setting the numerator of the P component of the position controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KP_s_Z'	Unsigned 16, value range 0 to 65535

Unit

Numerator

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the numerator of the proportional component of the position controller.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.16 Setting the denominator of the P component of the position controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KP_s_N'	Unsigned 8, value range 0 to 15

Unit

Denominator as powers of 2

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the denominator of the proportional component of the position controller as a power of 2.

0 = 1

1 = 2

2 = 4

3 = 8

etc.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.17 Setting the numerator of the I component of the position controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KI_s_Z'	Unsigned 16, value range 0 to 65535

Unit

Numerator

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the numerator of the integral component of the position controller.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.18 Setting the denominator of the I component of the position controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KI_s_N'	Unsigned 8, value range 0 to 15

Unit

Denominator as powers of 2

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the denominator of the integral component of the position controller as a power of 2.

0 = 1

1 = 2

2 = 4

3 = 8

etc.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.19 Setting the numerator of the D component of the position controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KD_s_Z'	Unsigned 16, value range 0 to 65535

Unit

Numerator

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the numerator of the differential component of the position controller.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

8.20 Setting the denominator of the D component of the position controller

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':CL_KD_s_N'	Unsigned 8, value range 0 to 15

Unit

Denominator as powers of 2

Firmware response

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

Invalid values are ignored.

Description

This parameter specifies the denominator of the differential component of the position controller as a power of 2.

0 = 1

1 = 2

2 = 4

3 = 8

etc.

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

9 Scope mode

9.1 Integration of a scope

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Description

In the scope mode, the values to be measured are selected and transmitted to the motor. The motor then carries out a measurement and returns the result in real-time to the NANOPRO driver software.

- The transmitted data are binary.
- The data are transmitted in the order of priority.
- The last data byte of each data packet contains a CRC8 checksum.

Examples

Each data source can be selected separately:

:Capt_Time=10	Send the selected data every 10 ms.
:Capt_Time=0	Ends the scope mode
:Capt_sPos=1	The setpoint position is selected
:Capt_sPos=0	The setpoint position is deselected

No data source is selected by default.

Data word when :Capt_sCurr=1 and :Capt_iIn=1

:Capt_sCurr_BYTE

:Capt_iIn_BYTE_HI

:Capt_iIn_BYTE_LO CRC

9.2 Setting the sample rate

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_Time'	Unsigned 16, value range 0 to 65535

Priority

–

Unit

ms (milliseconds)

Description

The parameter defines the time interval in ms in which the selected data are sent. The value range is "Unsigned 16".

"0" deactivates the scope function.

Example

:Capt_Time=10	Send the selected data every 10 ms.
:Capt_Time=0	Ends the scope mode

Reading out

If the keyword is sent without a "=", the current setting of the value can be read out.

9.3 Reading out the setpoint position of the ramp generator

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_sPos'	Signed 32

Priority

1

Unit

Steps

Description

Delivers the setpoint position generated by the ramp generator.

Example

:Capt_sPos=1	The setpoint position is selected
:Capt_sPos=0	The setpoint position is deselected

9.4 Reading out the actual position of the encoder

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_iPos'	Signed 32

Priority

2

Unit

Steps

Description

Returns the current encoder position.

9.5 Reading out the setpoint current of the motor driver

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_sCurr'	Signed 16

Priority

3

Unit

None

32767 corresponds to 150% of the maximum current (the value can also be negative)

Description

Delivers the setpoint current used for driving the motor.

9.6 Reading out the actual voltage of the driver

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_iVolt'	Unsigned 16, value range 0 to 65535

Priority

4

Unit

Value range 0 – 1023 (10Bit)

1023 is equivalent to 66.33 V

0 is equivalent to 0 V

Description

Delivers the voltage applied at the driver.

9.7 Reading out the digital inputs

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_iIn'	Unsigned 16, value range 0 to 65535

Priority

5

Unit

None

Description

Delivers the bit mask of the inputs.

9.8 Reading out the voltage at the analogue input

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_iAnalog'	Unsigned 16

Priority

6

Unit

0 is equivalent to 0 V

1023 is equivalent to +10 V

Description

Delivers the voltage of the analogue input.

9.9 Reading out the CAN bus load

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_iBus'	Unsigned 8

Priority

7

Unit

%

Invalid values are ignored.

Description

Delivers the approximate degree of utilisation of the CAN bus in %.

9.10 Reading out the driver temperature

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_ITemp'	Unsigned 16

Priority

8

Unit

Value range 0 – 1023

295 = 75 °C

261 = 80 °C

Description

Delivers the temperature measured in the driver.

9.11 Reading out the following error

Validity

Valid for firmware version 04-12-2008 and higher (for hardware SMCI33 and SMCI47-S).

Parameter

Keyword	Parameter
':Capt_IFollow'	Signed 32

Priority

9

Unit

Steps

Description

Delivers the difference between the setpoint and actual position.

Index

S

T