



Command Reference

Description of the Nanotec firmware serial commands

For SMCI32 / SMCI47 / PD4-I / PD6-I 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

© 2009

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 stepper motors SMCI32 and SMCI47
- Nanotec Plug & Drive motors PD4-I and PD6-I

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	2008-08-08	Created
V2.0	2009-01-12	Revision (firmware version 04-12-2008)

Contents

Editorial/About this manual	2
Contents	3
1 General information	5
1.1 Command structure.....	5
1.2 Note on old command format.....	6
1.3 Development support	6
2 Command overview	10
3 Read command	11
4 Records	12
5 General commands	13
5.1 Setting the phase current.....	13
5.2 Setting the phase current at a standstill.....	13
5.3 Setting the step mode	14
5.4 Setting the motor address.....	14
5.5 Setting the motor mode.....	15
5.6 Setting the limit switch behaviour.....	15
5.7 Setting the limit switch type.....	17
5.8 Setting the step angle	18
5.9 Setting the error correction mode	18
5.10 Setting the record for auto correction.....	19
5.11 Setting the encoder direction	19
5.12 Setting the settling time.....	20
5.13 Setting the maximum encoder deviation.....	20
5.14 Resetting the position error	21
5.15 Reading out the error memory	21
5.16 Reading out the encoder position	22
5.17 Reading out the position	23
5.18 Resetting the position.....	23
5.19 Reading out the parameter	24
5.20 Reading out the motor address.....	24
5.21 Reading out the status	25
5.22 Reading out the firmware version	26
5.23 Reading out the firmware version (old).....	26
5.24 Masking and demasking the inputs.....	26
5.25 Reversing the polarity of the inputs and outputs.....	29
5.26 Switching the interrupts of the inputs on and off.....	30
5.27 Setting the interrupts of the inputs to a rising flank.....	31
5.28 Setting the interrupts of the inputs to a falling flank	32
5.29 Setting the debounce time for the inputs	33

Contents

5.30	Setting the outputs	34
5.31	Setting automatic sending of the status	35
5.32	Starting the bootloader	35
5.33	Setting the reverse clearance	36
6	Record commands	37
6.1	Starting a record	37
6.2	Stopping a record	37
6.3	Loading a record from the EEPROM	37
6.4	Reading out the current record	39
6.5	Saving a record	40
6.6	Setting the positioning mode	41
6.7	Setting the travel distance	42
6.8	Setting the minimum frequency	42
6.9	Setting the maximum frequency	42
6.10	Setting the maximum frequency 2	42
6.11	Setting the ramp	42
6.12	Setting the direction of rotation	42
6.13	Setting the change of direction	42
6.14	Setting the repetitions	42
6.15	Setting the record pause	42
6.16	Setting the continuation record	42
7	Mode-specific commands	42
7.1	Setting the dead range for the joystick mode	42
7.2	Setting the filter for the analogue and joystick modes	42
7.3	Setting the minimum voltage for the analogue mode	42
7.4	Setting the maximum voltage for the analogue mode	42
7.5	Setting the dead range for the joystick mode	42
7.6	Increasing the speed	42
7.7	Reducing the speed	42
7.8	Actuating the trigger	42
Index	42

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

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 the old format see the Plug and Drive manual, series PDx-I (version 2.1).

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

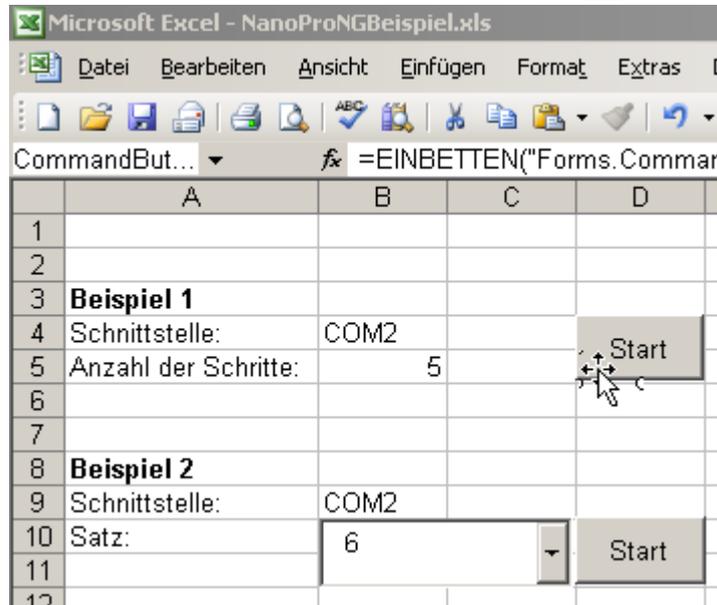
If you work under Windows with a .NET-capable programming language (e.g. Visual Basic of version 8 or higher, C#, Delphi.NET), we offer you the convenience of being able to integrate our DLL library in your application. You can conveniently control the functions of the driver via a function call without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

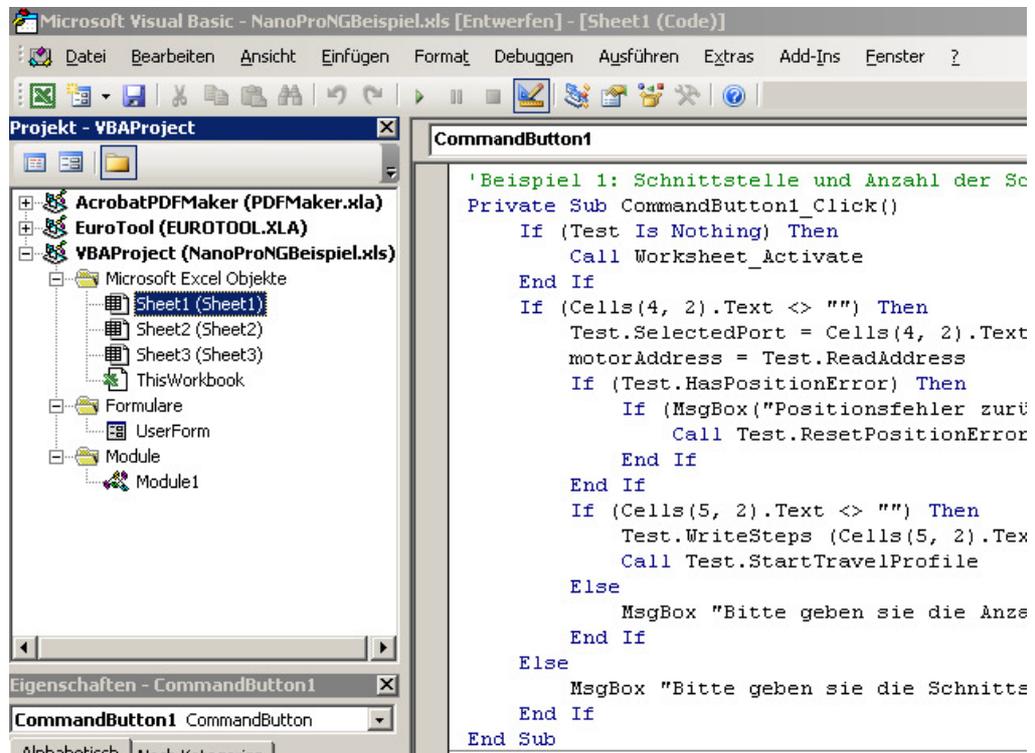
DLL library application

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



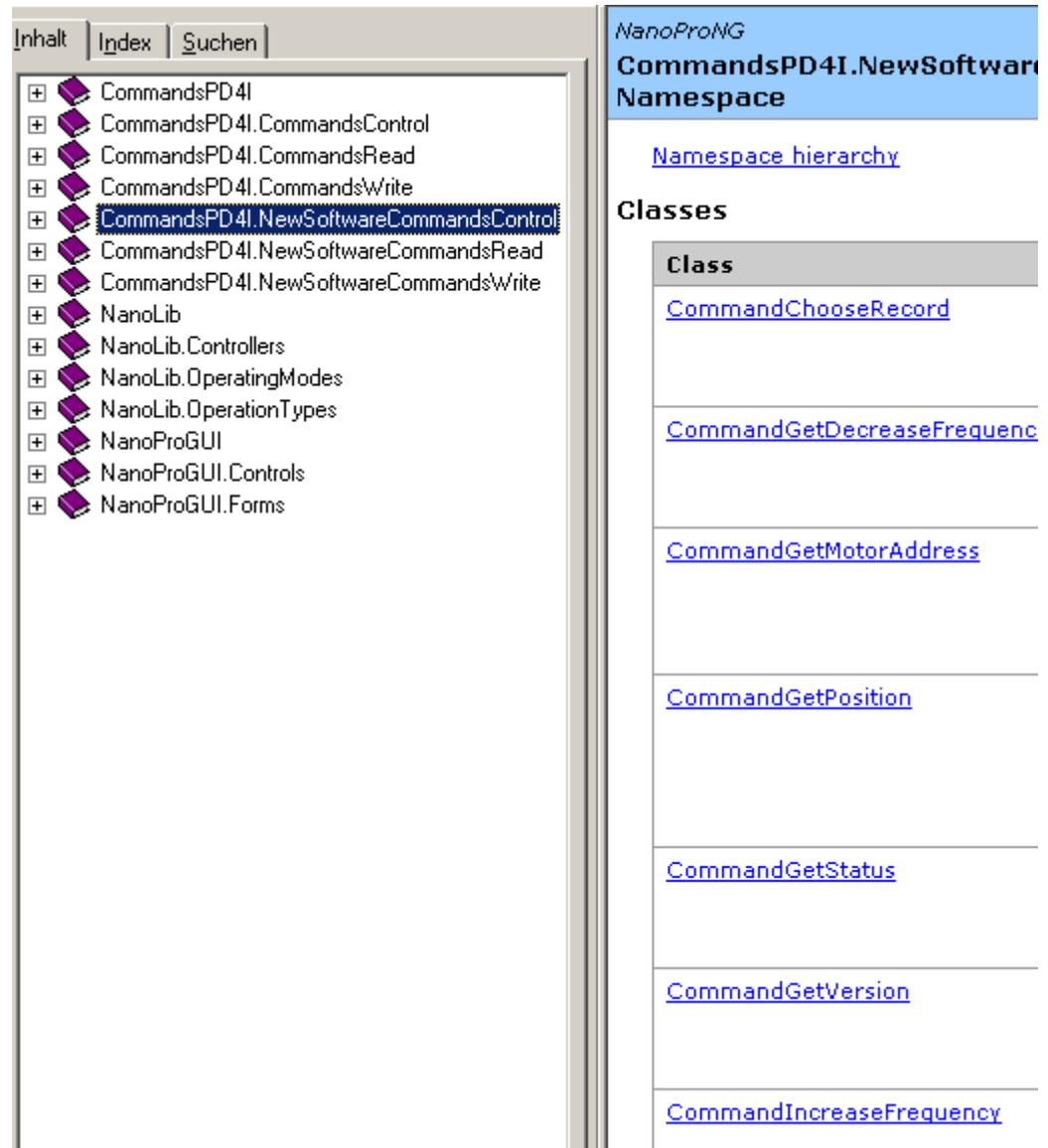
Associated VBA script



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



The screenshot shows a Windows help window with a tree view on the left and a class list on the right. The tree view shows a hierarchy of folders, with 'CommandsPD4I.NewSoftwareCommandsControl' selected. The right pane shows the 'Classes' section for the selected namespace, listing several classes with blue underlined links.

Class
CommandChooseRecord
CommandGetDecreaseFrequency
CommandGetMotorAddress
CommandGetPosition
CommandGetStatus
CommandGetVersion
CommandIncreaseFrequency

Overview of the CommandsControl Classes

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 Member

[CommandChooseRecord overview](#)

Public Instance Constructors

CommandChooseRecord Constructor	Initialisiert den CommandChooseRecord
---	---

See Also

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

2 Command overview

Overview of the commands

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

- ... Reducing the speed	47	k ... Switching the interrupts of the inputs on and off	28
!... Setting the motor mode	15	l (Pipe) ... Reading out the current record.....	36
\$... Reading out the status.....	24	L ... Masking and demasking inputs	25
% ... Setting the dead range for the joystick mode.....	47	l ... Setting the motor mode.....	15
-(Space) ... Reading out the firmware version (old).....	25	M ... Reading out the motor address	23
\ ... \ Setting the interrupts of the inputs to a falling flank.....	30	m ... Setting the motor address.....	14
/ ... Setting the interrupts of the inputs to a rising flank.....	29	N ... Setting the continuation record	44
@A ... Starting the bootloader	33	n ... Setting the maximum frequency 2	41
+ ... Increasing the speed	47	o ... Setting the maximum frequency	41
= ... Setting the dead range for the joystick mode.....	45	O ... Setting the settling time.....	19
> ... Saving a record	37	p ... Setting the positioning mode	38
a ... Setting the step angle.....	17	P ... Setting the record pause	44
A ... Starting a record.....	35	q ... Setting the encoder direction	18
b ... Setting the ramp	42	Q ... Setting the minimum voltage for the analogue mode	46
C ... Reading out the position	22	r ... Set the phase current at standstill	13
c ... Resetting the position	22	R ... Setting the maximum voltage for the analogue mode	46
D ... Resetting the position error.....	20	s ... Setting the travel distance.....	40
d ... Setting the direction of rotation.....	42	S ... Stopping a record	35
E ... Reading out the error memory	20	T ... Actuating the trigger	48
e ... Set the limit switch type	16	t ... Setting the change of direction	43
f ... Setting the filter for the analogue and joystick modes	45	U ... Setting the error correction mode.....	17
F ... Setting the record for auto correction.....	18	u ... Setting the minimum frequency	40
g ... Setting the step mode.....	14	v ... Reading out the firmware version	25
h ... Reversing the polarity of the inputs and outputs	27	W ... Setting the repetitions.....	43
l ... Reading out the error memory	21	X ... Setting the maximum encoder deviation	19
i ... Setting the phase current	13	y ... Loading a record from the EEPROM	35
J ... Setting automatic sending of the status..	33	Y ... Setting the outputs.....	32
K ... Setting the debounce time for the inputs	31	Z ... Reading out the parameter.....	23
		z ... Set the reverse clearance	34
		Z + parameter ... Read command.....	11

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	Parameter	See section	Page
Position mode	'p'	6.6 <i>Setting the positioning mode</i>	41
Travel distance	's'	6.7 <i>Setting the travel distance</i>	42
Initial step frequency	'u'	6.8 <i>Setting the minimum frequency</i>	42
Maximum step frequency	'o'	6.9 <i>Setting the maximum frequency</i>	42
Second maximum step frequency	'n'	6.10 <i>Setting the maximum frequency 2</i>	42
Acceleration and braking ramp	'b'	6.11 <i>Setting the ramp</i>	42
Direction of rotation	'd'	6.12 <i>Setting the direction of rotation</i>	42
Reversal in direction of rotation for repeat records	't'	6.13 <i>Setting the change of direction</i>	42
Repetitions	'w'	6.14 <i>Setting the repetitions</i>	42
Pause between repetitions and continuation records	'P'	6.15 <i>Setting the record pause</i>	42
Record number of continuation record	'N'	6.16 <i>Setting the continuation record</i>	42

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
'I'	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
'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 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 'ZI' 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 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.

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 'l'* 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.

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	Parameter	See section	Page
Position mode	'p'	<i>6.6 Setting the positioning mode</i>	41
Travel distance	's'	<i>6.7 Setting the travel distance</i>	42
Initial step frequency	'u'	<i>6.8 Setting the minimum frequency</i>	42
Maximum step frequency	'o'	<i>6.9 Setting the maximum frequency</i>	42
Second maximum step frequency	'n'	<i>6.10 Setting the maximum frequency 2</i>	42
Acceleration and braking ramp	'b'	<i>6.11 Setting the ramp</i>	42
Direction of rotation	'd'	<i>6.12 Setting the direction of rotation</i>	42
Reversal in direction of rotation for repeat records	't'	<i>6.13 Setting the change of direction</i>	42
Repetitions	'w'	<i>6.14 Setting the repetitions</i>	42
Pause between repetitions and continuation records	'P'	<i>6.15 Setting the record pause</i>	42
Record number of continuation record	'N'	<i>6.16 Setting the continuation record</i>	42

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 <i>Setting the travel distance 's'</i> defines the travel distance relative to the current position. Command 6.12 <i>Setting the direction of rotation 'd'</i> defines the direction. Parameter 6.7 <i>Setting the travel distance 's'</i> must be positive.
p=2	Absolute positioning; Command 6.7 <i>Setting the travel distance 's'</i> defines the target position relative to the reference position. Command 6.12 <i>Setting the direction of rotation 'd'</i> is ignored.
p=3	Internal reference run; The motor runs at the lowest speed in the direction set in command 6.12 <i>Setting the direction of rotation 'd'</i> 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 <i>Setting the limit switch behaviour 'l'</i> . 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 <i>Setting the direction of rotation 'd'</i> until it reaches the limit switch. Then a free run is performed, depending on the setting. See command 5.6 <i>Setting the limit switch behaviour 'l'</i> .
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 <i>Actuating the trigger 'T'</i> or trigger input), the motor moves the set travel distance (command 6.7 <i>Setting the travel distance 's'</i>) and, for this purpose, changes its speed to the maximum speed 2 (command 6.10 <i>Setting the maximum frequency 2 'n'</i>).
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 $\pm 2^{31}$).

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.

Index

A

Actuating the trigger48

D

Debouncing31

Debouncing inputs.....31

Demasking inputs25

DLL library6

Driver command structure5

Driver response5

Driver status21

E

Error codes20

I

Increasing the speed47

L

Loading a record from the EEPROM.....35

M

Masking inputs.....25

O

Old command format6

R

Read command11

Reading out the current record.....36

Reading out the encoder position.....21

Reading out the error memory20

Reading out the firmware version.....25

Reading out the firmware version (old)25

Reading out the motor address23

Reading out the parameter.....23

Reading out the position.....22

Reading out the status.....24

Records12

Reducing the speed.....47

Resetting the position22

Resetting the position error.....20

Reversing the polarity of the inputs and
outputs27

RS485 interface specification5

S

Save travel distances.....12

Saving a record.....37

Set the limit switch type16

Set the phase current.....13

Set the phase current at standstill13

Set the reverse clearance34

Setting analogue mode.....39

Setting automatic sending of the status.....33

Setting clock direction mode.....39

Setting flag positioning mode.....39

Setting joystick mode.....39

Setting speed mode.....38

Setting the change of direction43

Setting the continuation record.....44

Setting the dead range for the joystick mode
.....45, 47

Setting the debounce time for the inputs31

Setting the direction of rotation42

Setting the encoder direction18

Setting the error correction mode17

Setting the filter for analogue mode.....45

Setting the interrupts of the inputs to a falling
flank.....30

Setting the interrupts of the inputs to a rising
flank.....29

Setting the limit switch behaviour15

Setting the maximum encoder deviation.....19

Setting the maximum frequency41

Setting the maximum frequency 241

Setting the maximum voltage for the analogue
mode46

Setting the minimum frequency40

Setting the minimum voltage for the analogue mode.....	46
Setting the motor address	14
Setting the motor mode	15
Setting the outputs.....	32
Setting the position mode	38
Setting the positioning mode	38
Setting the ramp	42
Setting the record for auto correction	18
Setting the record pause	44
Setting the repetitions	43
Setting the settling time.....	19
Setting the step angle	17
Setting the step mode	14
Setting the travel distance	40
Starting a record	35
Starting the bootloader.....	33
Stopping a record.....	35
Switching the interrupts of the inputs on and off	28
T	
Travel distance, save	12