FAULHABER

Motion Control

Serie MCBL 300x RS
Serie MCDC 300x RS
Serie 3564...B CS
Serie 32xx...BX4 CS
Serie 22xx...BX4 CSD

Communication and Function Manual

EN

RS232

WE CREATE MOTION

Imprint

Version:
2nd issue, 08.12.2011

This communication and function manual has been prepared with care.
Dr. Fritz Faulhaber GmbH & Co. KG cannot accept any liability for any errors in this communication and function manual or for the consequences of such errors. Equally, no liability can be accepted for direct or consequential damages resulting from improper use of the equipment.

The relevant regulations regarding safety engineering and interference suppression as well as the requirements specified in this communication and function manual are to be noted and followed when using the software.

Subject to change without notice.

The respective current version of this communication and function manual is available on FAULHABER's internet site:
www.faulhaber.com

Overview

Overview of the Faulhaber Motion Control Drives documents

Document	Contents
Technical Manual	Device installation, safety, specification
Communication and function manual (RS232)	Initial start-up, function overview, protocol description, parameter description and notes on autonomous sequential programs
Motion Manager instruction manual	Operation of the "FAULHABER Motion Manager" PC software for configuration and commissioning
Product data sheets	Technical limit and operating data

Guide to the Document

Table of Contents

Table of Contents

1 Important Information

1.1 Symbols used in this manual

WARNING! ***Warning!***

This pictogram with the wording "Warning!" indicates an imminent danger which can result in physical injuries.

► *This arrow points out the appropriate action to take to prevent the imminent danger.*

CAUTION! ***Caution!***

This pictogram with the wording "Caution!" indicates an imminent danger which can result in slight physical injuries or material damage.

► *This arrow points out the appropriate precautions.*

REGULATION! ***Regulations, guidelines and directives***

This pictogram with the wording "Regulation" indicates a statutory regulation, guideline or directive which must be observed in the respective context of the text.

NOTE ***Note***

This "Note" pictogram provides tips and recommendations for use and handling of the component.

1 Important Information

1.2 Additional information

WARNING! ***Risk of injuries***

Failure to comply with the safety instructions during installation and operation can result in irreparable damage to the device and a risk of injuries to the operating personnel.

- ► *Please read through the whole of your drive's technical manual before installing the drive.*
- ► *Keep this communication and function manual in a safe place for subsequent use.*

NOTE

Always use the current version of the Faulhaber MotionManager.

The respective current version is available to download from www.faulhaber.com/MotionManager.

NOTE

The information given in this instruction manual refers to the standard version of the drives.

Please refer to any additional information sheet provided in the event of differences in information due to a customer-specific motor modification.

NOTE ***RS232 interface***

The drive can also be operated independently of the RS232 interface if the desired function, such as velocity or position controller, has been previously programmed via analog input, stepper motor or electronic gear.

2 Quick Start

To facilitate introduction, this chapter highlights the initial steps for commissioning and operation of FAULHABER Motion Controllers with serial interface. However, the detailed documentation must always be read and adhered to, particularly Chapter 5.1 „Basic settings“!

The units are delivered as standard without a valid node address (NODEADR0) and with a transfer rate of 9 600 baud. The settings can be changed via the interface, e.g. with the FAULHABER Motion Manager (see Chapter 5.2.1 „Connection Parameters“).

The following steps are necessary for commissioning using the default configuration:

1. Connect the drive unit to a 12V – 24V voltage source.
   For details of the connection cable assignment, see **Chapter 3 „Installation“ in the technical manual.**
2. Connect drive unit to a serial interface of the PC (e.g. COM1) and switch on.
   For details of the interface, see **Chapter 3 „Installation“ in the technical manual.**
3. **Configuration and motion commands can now be executed via suitable software, e.g. FAULHABER Motion Manager.**

NOTE

Use of a USB serial adapter is recommended if the PC used does not have a serial port.

Operation via FAULHABER Motion Manager

The FAULHABER Motion Manager offers easy access to the Motion Controller’s command set. The desired node must have been activated beforehand by double clicking in Node Explorer in the case of network operation.

The FAULHABER commands described below can be entered directly in the command input line or selected from the Commands menu.

In order to drive a motor via the Motion Manager, follow the procedure below (assuming a matching baud rate):

1. Start FAULHABER Motion Manager.
2. Configure drive functions:

Motion control systems with electronics built onto the motor are already preset in the factory. Motion controllers with an externally connected motor must be equipped with current limitation values suitable for the motor and suitable controller parameters before being started up.

The motor wizard is available for selection of the motor and corresponding suitable basic parameters from Version 4.6 and higher of the Motion Manager.

Other settings, e.g. for the function of the fault output, can be made under the “Configuration – Drive functions” menu item, where a convenient dialog is provided (see Chapter 5.2 „Configuration using FAULHABER Motion Manager“). The configuration dialog is also available for direct access in the Wizard bar on the right-hand side of the Motion Manager window (configuration wizard).

CAUTION!

Check basic settings

Incorrect values in the Motion Controller's settings can result in damage to the controller and/or drive (see Chapter 5.1 „Basic settings“).

To operate the drive via the PC, set value presetting must be set to digital (SOR0).

If the settings are to be permanently stored, press the “EEPSAV” button.

2 Quick Start

3. Activate drive:

   „**EN**“ command.

   Enter in command input field and press “Send” button or select in “Commands – Motion control – Enable drive” menu and press “Send” button.

4. Operate drive (examples):

   - Drive motor with 100 rpm velocity control:

     „**V100**“ command.

     Enter in command input field and press „Send“ button or

     select from “Commands – Motion control – Drives with constant velocity“ menu, enter value 100 in dialogue box, press OK and “Send” button.

   - Stop drive:

     „**V0**“ command.

   - Move motor relatively by 10000 increments:

     „**LR10000**“ command to load the relative target position, “M” command to move to loaded target position.

5. Deactivate drive again

   „**DI**“ command.

The controller tuning wizard

Version 4.6 and higher of the Motion Manager provides a Controller Tuning Wizard, with which the controller parameters of the speed and positioning controller can be adjusted to the application.

WARNING!

Warning!

During operation with the Tuning Wizard, the motor is alternately run at different speeds.

► *The motor must be installed so that it can freely move for the parameter search.*

3 Functional Description

Guide

The Motion Controllers can be configured for different operating modes. As standard the drive unit is delivered as a servomotor with set value presetting via the serial interface. The drive can be reconfigured by means of the corresponding FAULHABER commands.

Command	Argument	Function	Description
SOR	0 – 4	Source for Velocity	Source for velocity presetting 0: Serial interface (default) 1: Voltage at analog input 2: PWM signal at analog input 3: Current target value via analog input 4: Current target value via analog input with presetting of the direction of rotation via input polarity
CONTMOD	-	Continuous Mode	Switch back to normal mode from an enhanced mode .
STEPMOD	-	Stepper Motor Mode	Change to stepper motor mode
APCMOD	-	Analog Position Control Mode	Change to position control via analog voltage
ENCMOD	-	Encoder Mode	Change to encoder mode (not for MCDC) An external encoder serves as position detector (the current position value is set to 0)
HALLSPEED	-	Hall sensor as speed sensor	Speed via Hall sensors in encoder mode (not for MCDC)
ENCSPEED	-	Encoder as speed sensor	Speed via encoder signals in encoder mode (not for MCDC)
GEARMOD	-	Gearing Mode	Change to gearing mode
VOLTMOD	-	Set Voltage Mode	Activate Voltage Regulator Mode
IXRMOD	-	Set IxR Mode	Activate IxR control (MCDC only)

If the settings are to be permanently stored, the command SAVE must be executed after the configuration; this saves the current settings in the Flash data memory, from where they are reloaded when the unit is next switched on. Alternatively, the EEPSAV command can also be executed. Both commands are identical, therefore SAVE only is used in the following.

The power stage must be activated (EN) for the drive to operate.

All commands listed further below are summarised and explained again in Chapter 7 „Parameter Description“.

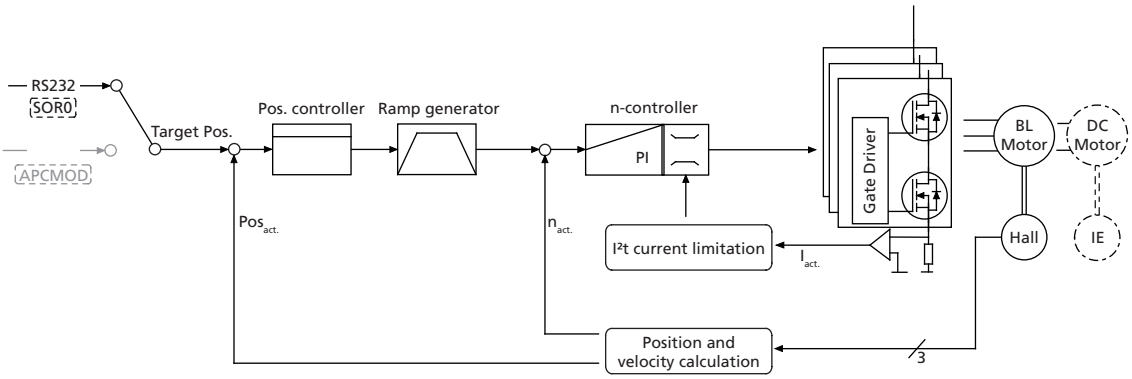
3 Functional Description

3.1 Position control

Guide

3.1.1 Set value presetting via the serial interface

Controller structure for set value presetting via the serial interface or via a sequence program



In this operating mode, target positions can be preset via the serial interface or a sequence program:

Basic settings

CONTMOD and SOR0 operating mode.

The positioning range limits can be set via the command LL and activated via APL.

The proportional amplification PP and a differential term PD can be set for the position controller.

Command	Argument	Function	Description
PP	Value	Load Position Proportional Term	Load position controller amplification. **Value: 1 … 255**
PD	Value	Load Position Differential Term	Load position controller D-term. **Value: 1 … 255**
LL	Value	Load Position Range Limits	Load limit positions (the drive cannot be moved out of these limits). Positive values specify the upper limit and negative values the lower. The range limits are only active if APL1 is. **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
APL	0 -1	Activate/Deactivate Position Limits	Activate range limits (LL) (valid for all operating modes except VOLTMOD). 1: Position limits activated 0: Position limits deactivated

3 Functional Description

3.1 Position control

Additional settings

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Velocity controller / current limitation

The controller parameters POR and I of the velocity controller can be adjusted. In addition, the current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.2 „Velocity control“).

Motion control commands

The positioning is executed via the FAULHABER motion control commands. An overview of all motion control commands is given in Chapter 7.4 „Motion control commands“.

Command	Argument	Function	Description
EN	-	Enable Drive	Activate drive
DI	-	Disable Drive	Deactivate drive
LA	Value	Load Absolute Position	Load new absolute target position **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
LR	Value	Load Relative Position	Load new relative target position, in relation to last started target position. The resulting absolute target position must lie between the values given below. **Value: $-2.14 \cdot 10^9$ … $2.14 \cdot 10^9$**
M	-	Initiate Motion	Activate position control and start positioning
HO	-/value	Define Home Position	Without argument: Set actual position to 0. With argument: Set actual position to specified value. **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
NP	-/value	Notify Position	Without argument: A “p” is returned when the target position is attained. With argument: When the specified position is passed a "p" is returned. **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
NPOFF	-	Notify Position Off	A notify position command not yet triggered is deactivated again.

Example:

- Load target position: `LA40000`
- Start positioning: `M`

Attainment of the target position or any intermediate position is indicated by a “p” on the serial interface if “Notify Position” is set before the start of positioning, provided that ANSW1 or ANSW2 is set:

Position resolution

If the linear Hall sensors of the brushless motors are used as position transducers, 3000 pulses per revolution are supplied.

3 Functional Description

3.1 Position control

Complex motion profiles

More complex motion profiles can be generated through appropriate presetting of new values (maximum speed, acceleration, end position) during positioning. After a value change, simply execute a new motion start command (M). The commands NP and NV can be used to control the sequence.

Further information on compiling motion profiles is given in Chapter 3.6.1 „Ramp generator“.

Positioning beyond the range limits

In the case of APL0, relative positioning can also be executed beyond the range limits.

If the upper (1 800 000 000) or lower limit (-1 800 000 000) is exceeded, counting is continued at 0 without loss of increments.

Digital signal target position

The entry into the target corridor can be displayed via the fault output as a digital output signal in the POSOUT function. The signal is not reset until a further Motion start command (M).

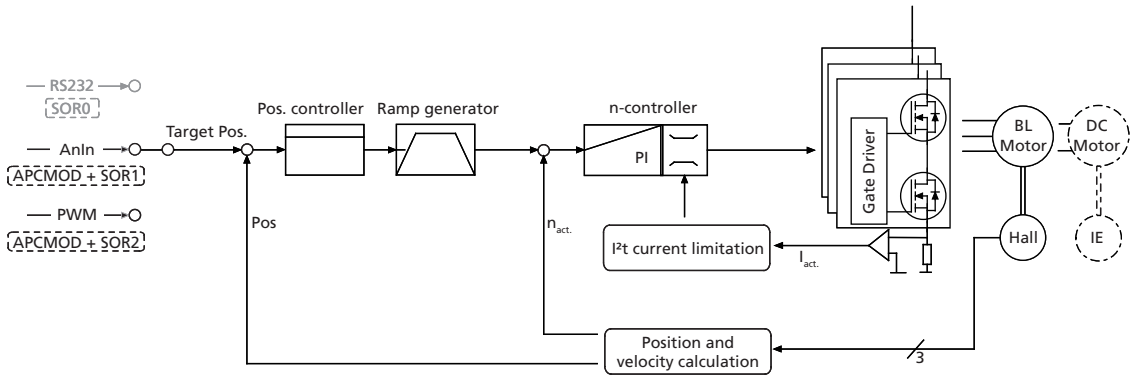
For notes on configuration, see Chapter 3.5 „Special fault output functions“.

3 Functional Description

3.1 Position control

3.1.2 Analog positioning mode (APCMOD)

Controller structure for set-point presetting via an analog voltage



In this operating mode the target position can be preset using an analog voltage at the AnIn input.

Basic settings

APCMOD mode and SOR1 or SOR2.

The positioning range limits can be set via the command LL and activated via APL.

The proportional amplification PP and a differential term PD can be set for the position controller.

The maximum position to be approached with a voltage of 10 V can be preselected with the LL command. At -10 V the drive moves in the opposite direction up to the set negative range limit.

Irrespective of the preset LL value, the maximum position is limited to 3 000 000 in APCMOD.

Comment: The resolution of the analog input is limited to 12 bit (4096 steps).

The direction of rotation can be predefined with the commands ADL and ADR.

Additional settings

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Velocity controller / current limitation

The controller parameters POR and I of the velocity controller can be adjusted. In addition, the current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.2 „Velocity control“).

3 Functional Description

3.1 Position control

Positioning via pulse width signal (PWM) at the analog input (SOR2)

If SOR2 is set in APCMOD, the pulse duty factor of a PWM signal can be used as position set-point.

On delivery:

- Pulse duty factor > 50% ➔ positive target position
- Pulse duty factor = 50% ➔ target position = 0
- Pulse duty factor < 50% ➔ negative target position

Absolute positioning within one revolution (only for BL 2 pole):

In motion control systems with brushless 2-pole motors, the initial position is absolutely initialised within one revolution after the motor is switched on (0 - 3 000 corresponds to 0 - 360° of the rotor position). This means that even if the power supply is disconnected, the position determination supplies the correct position value after restarting (if the rotor has only been turned within one revolution).

The following commands enable the drive to be accurately positioned in the voltage range 0 V … 10 V within one revolution and to return to the correct position even after the supply has been switched off, without homing.

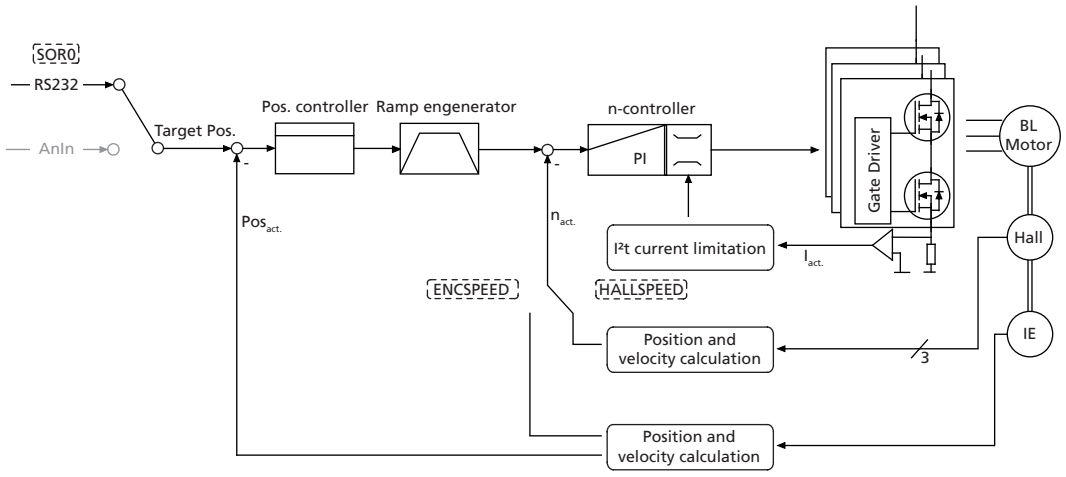
- Switch over to analog positioning: `APCMOD`
- Hide negative range: `LL-1`
- Fix maximum position to 1 revolution: `LL3000`

3 Functional Description

3.1 Position control

3.1.3 External encoder as actual position value (ENCMOD) - not for MCDC

Controller structure for using and external encoder as the actual value encoder



For high-precision applications, the actual values of BL motors can be derived from an external encoder.

- Depending on the application, the velocity can be derived from the encoder or from the Hall sensors.
- The external encoder can be mounted directly on the motor shaft, but an encoder that is mounted to the application output (e.g. glass scale) is particularly advantageous. This allows the high precision to be set directly at the output.
- Commutation still occurs via the analog Hall sensors.

Basic settings

ENCMOD and SOR0 operating mode.

The positioning range limits can be set via the command LL and activated via APL. The proportional amplification PP and a differential term PD can be set for the position controller.

Command	Argument	Function	Description
PP	Value	Load Position Proportional Term	Load position controller amplification. **Value: 1 … 255**
PD	Value	Load Position Differential Term	Load position controller D-term. **Value: 1 … 255**
LL	Value	Load Position Range Limits	Load limit positions (the drive cannot be moved out of these limits). Positive values specify the upper limit and negative values the lower. The range limits are only active if APL1 is. **Value: $-1.8 \cdot 10^9$ … $+1.8 \cdot 10^9$**
APL	0 - 1	Activate/Deactivate Position Limits	Activate range limits (LL) (valid for all operating modes except VOLTMOD). 1: Position limits activated 0: Position limits deactivated

3 Functional Description

3.1 Position control

Settings for external encoder

Command	Argument	Function	Description
ENCMOD	-	Encoder Mode	Change to encoder mode (not for MCDC) An external encoder serves as position transducer (the current position value is set to 0).
ENCSPEED	-	Encoder as speed sensor	Speed via encoder signals in encoder mode
HALLSPEED	-	Hall sensor as speed sensor	Speed via hall sensors in encoder mode
ENCRES	Value	Load Encoder Resolution	Load resolution of external encoder (4 times pulse/rev). **Value: 8 ... 65 535**

Additional settings

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Velocity controller / current limitation

The controller parameters POR and I of the velocity controller can be adjusted. In addition, the current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.2 „Velocity control“ and Chapter 3.6.3 „Current controller and I²t current limitation“).

3 Functional Description

3.1 Position control

Motion control commands

Positioning in the ENCMOD is executed in precisely the same way as in CONTMOD, using the FAULHABER motion control commands. An overview of all motion control commands is given in Chapter 7.4 „Motion control commands“.

Command	Argument	Function	Description
EN	-	Enable Drive	Activate drive
DI	-	Disable Drive	Deactivate drive
LA	Value	Load Absolute Position	Load new absolute target position **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
LR	Value	Load Relative Position	Load new relative target position, in relation to last started target position. The resulting absolute target position must lie between the values given below. **Value: $-2.14 \cdot 10^9$ … $2.14 \cdot 10^9$**
M	-	Initiate Motion	Activate position control and start positioning
HO	-/value	Define Home Position	Without argument: Set actual position to 0. With argument: Set actual position to specified value. **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
NP	-/value	Notify Position	Without argument: A “p” is returned when the target position is attained. With argument: A “p” is returned if the specified position is over-travelled. **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
NPOFF	-	Notify Position Off	Notify Position command that has not yet been triggered is deactivated again.

Example:

- Load target position: `LA40000`
- Start positioning: `M`

Attainment of the target position or any intermediate position is indicated by a “p” on the serial interface if “Notify Position” is set before the start of positioning, provided that ANSW1 or ANSW2 is set.

Actual value resolution

In ENCMOD the resolution of the position values depends on the resolution of the encoder.

Complex motion profiles

More complex motion profiles can be generated through appropriate presetting of new values (maximum speed, acceleration, end position) during positioning. After a value change, simply execute a new motion start command (M). The commands NP and NV can be used to control the sequence.

Further information on compiling motion profiles is given in Chapter 3.6.1 „Ramp generator“.

Positioning beyond the range limits

In the case of APL0, relative positioning can also be executed beyond the range limits. If the upper (1 800 000 000) or lower limit (-1 800 000 000) is exceeded, counting is continued at 0 without loss of increments.

Digital signal target position

The entry into the target corridor can be displayed via the fault output as a digital output signal in the POSOUT function. The signal is not reset until a further Motion start command (M).

For notes on configuration, see Chapter 3.5 „Special fault output functions“.

3 Functional Description

3.2 Velocity control

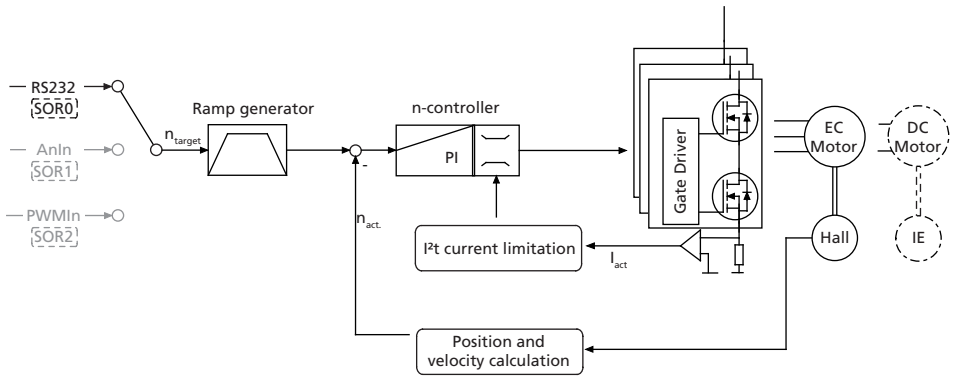
In velocity control mode the velocity of the drive is controlled by a PI controller. Provided the drive is not overloaded, the drive follows the presetting without deviation.

The current velocity of BL motors can be detected both from the Hall signals and via an additional encoder; an incremental encoder is always required for DC motors. One exception is IxR control, as described in Chapter 3.4.5 „IxR control for MCDC“.

The velocity can be preset via the serial interface or from sequence programs, via an analog voltage preset or a PWM signal.

3.2.1 Velocity presetting via the serial interface

Controller structure for velocity control



In this operating mode the drive can be operated by velocity controlled with set-point presetting via RS232 or from a sequence program.

Basic settings

CONTMOD and SOR0 operating mode.

The controller parameters POR and I and the sampling rate can be adjusted for the velocity controller.

Command	Argument	Function	Description
POR	Value	Load Velocity Proportional Term	Load velocity controller amplification. **Value: 1 ... 255**
I	Value	Load Velocity Integral Term	Load velocity controller integral term. **Value: 1 ... 255**
SR	Value	Load Sampling Rate	Load sampling rate of the velocity controller as a multiple of the basic sampling time. **Value: 1 ... 20**

3 Functional Description

3.2 Velocity control

Velocity input

In BL motors the current velocity is determined in CONTMOD by evaluating the Hall sensor signals, which supply 3 000 pulses per revolution. In DC motors the velocity is determined using an incremental encoder whose resolution has to be set using the ENCRES command. DC motors without an incremental encoder can also be operated with limited accuracy in IxR mode (see Chapter 3.4.5 „IxR control for MCDC“).

Command	Argument	Function	Description
ENCRES	Value	Load Encoder Resolution	Load resolution of external encoder (4 times pulse/rev). **Value: 8 ... 65 535**

Additional settings

Movement limits

The LL command can also be used to define a movement range limit for velocity mode. The APL1 command activates monitoring of these limits.

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Current limitation

The current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.6.3 „Current controller and I²t current limitation“).

Motion control commands

An overview of all motion control commands is given in Chapter 7.4 „Motion control commands“.

Command	Argument	Function	Description
EN	-	Enable Drive	Activate drive
DI	-	Disable Drive	Deactivate drive
V	Value	Select Velocity Mode	Activate velocity mode and set specified value as target velocity (velocity control). Unit: rpm

Example:

- Drive motor at 100 rpm: `V100`

  In order to change the direction of rotation, simply assign a negative velocity value (e.g. V-100).
- Stop motor: `V0`

NOTE

Make sure that APL0 is set, if you do not want the drive to stop at the set range limits (LL)!

Also check that the maximum speed SP is not set below the desired target velocity.

3 Functional Description

3.2 Velocity control

Complex motion profiles

Reaching the given speed is indicated by a “v”, if “Notify Velocity“ has been set before starting the speed mode and ANSW1 or ANSW2 is set:

Command	Argument	Function	Description
NV	Value	Notify Velocity	A “v” is returned when the nominal speed is reached or passed through. **Value: –32 767 … 32 767**
NVOFF	-	Notify Velocity Off	Velocity command that has not yet been triggered is deactivated again.

3.2.2 Velocity presetting via an analog voltage or a PWM signal

In this operating mode, the drive velocity can be controlled with set value presetting via an analog voltage or a PWM signal.

Basic settings

CONTMOD mode and SOR1 (AnIn) or SOR2 (PWMIn).

The controller parameters POR, I and the sampling rate can be adjusted for the velocity controller. In addition, commands are available for configuring the analog velocity presetting.

Command	Argument	Function	Description
SP	Value	Load Maximum Speed	Load maximum speed (here: Target velocity at 10 V). Setting applies to all modes (except VOLTMOD) Unit: rpm
MV	Value	Minimum Velocity	Specifies the lowest velocity Unit: rpm
MAV	Value	Minimum Analog Voltage	Specifies the minimum start voltage Unit: rpm
ADL	-	Analog Direction Left	Positive voltages at the analog input result in anticlockwise rotation of the rotor
ADR	-	Analog Direction Right	Positive voltages at the analog input result in clockwise rotation of the rotor
DIRIN	-	Direction Input	Use fault pin as rotational direction input Low: … Left-hand rotation (corresponding to ADL command) High: … Right-hand rotation (corresponding to ADR command)
POR	Value	Load Velocity Proportional Term	Load velocity controller amplification. **Value: 1 … 255**
I	Value	Load Velocity Integral Term	Load velocity controller integral term. **Value: 1 … 255**
SR	Value	Load Sampling Rate	Load sampling rate of the velocity controller as a multiple of the basic sampling time. **Value: 1 … 20**

Velocity input

By default, in BL motors the current speed is determined by evaluating the Hall sensor signals. Additional incremental encoders cannot be connected to BL motors for analog velocity presetting.

In DC motors the velocity is solely determined using the incremental encoder. DC motors without an incremental encoder can also be operated with limited accuracy in IxR mode (see Chapter 3.4.5 „IxR control for MCDC“).

3 Functional Description

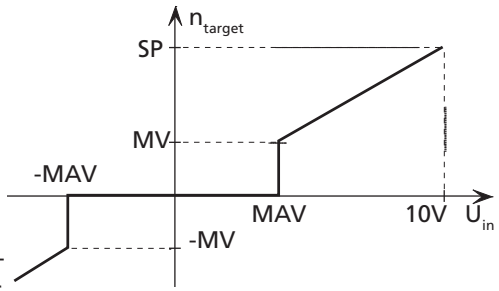
3.2 Velocity control

Target value input

Example:

The drive is only to start moving with voltages over 100 mV or below -100 mV at the analog input:

- MAV100



Advantage:

As 0 mV is usually difficult to set at the analog input, 0 rpm is also not easy to implement. The dead band produced by the minimum start voltage prevents the motor from starting as a result of small interference voltages.

Additional settings

Movement limits

The LL command can also be used to define a movement range limit for velocity mode. The APL1 command activates monitoring of these limits.

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Current limitation

The current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.6.3 „Current controller and I²t current limitation“).

Set-point presetting via pulse width signal (PWM) at the analog input (SOR2)

If SOR2 is set in APCMOD, the pulse duty factor of a PWM signal can be used as velocity target.

On delivery:

- Pulse duty factor > 50% ➔ clockwise rotation
- Pulse duty factor = 50% ➔ stoppage n = 0
- Pulse duty factor < 50% ➔ anti-clockwise rotation

The commands SP, MV, MAV, ADL and ADR can also be used here.

NOTE

Make sure that APL0 is set, if you do not want the drive to stop at the set range limits (LL)!

Input circuit

The input circuit at the analog input is designed as a differential amplifier. If the analog input is open, an undefined velocity can be set. The input must be connected to AGND with low-impedance or set to the voltage level of the AGND, in order to generate 0 rpm.

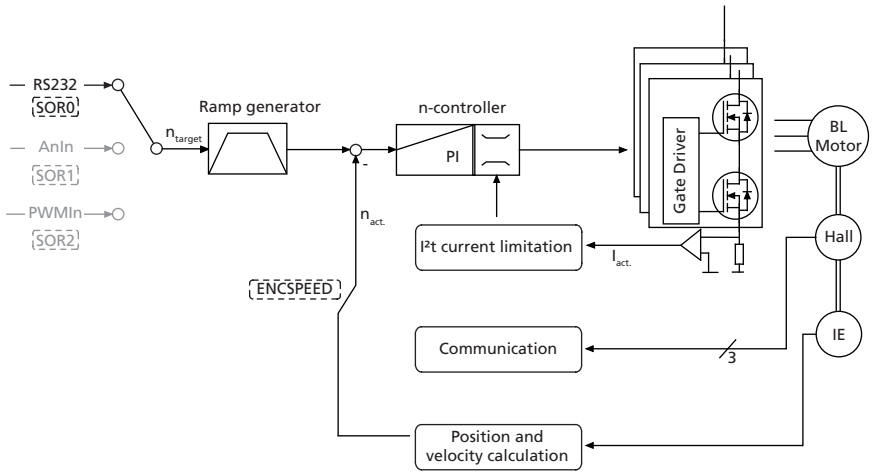
For a protective circuit example, see **Chapter 3.4 in the technical manual.**

3 Functional Description

3.2 Velocity control

3.2.3 External encoder as actual velocity value (ENCMOD) - not for MCDC

Velocity control with external encoder as actual value



In this operating mode the drive can be operated by velocity controlled with set-point presetting via RS232 or from a sequence program. The velocity is evaluated via an additional encoder, external or built onto the motor. In particular, this enables a specific load speed to be controlled by an incremental encoder at the output.

ENCMOD mode is available for BL motors only. The analog Hall sensors of the motors are also evaluated in ENCMOD mode for the motor commutation.

Basic settings

ENCMOD and SOR0 operating mode.

The controller parameters POR and I and the sampling rate can be adjusted for the velocity controller.

Command	Argument	Function	Description
POR	Value	Load Velocity Proportional Term	Load velocity controller amplification. **Value: 1 ... 255**
I	Value	Load Velocity Integral Term	Load velocity controller integral term. **Value: 1 ... 255**
SR	Value	Load Sampling Rate	Load sampling rate of the velocity controller as a multiple of the basic sampling time. **Value: 1 ... 20**

3 Functional Description

3.2 Velocity control

Velocity input

The external incremental encoder's resolution must be specified with 4 edge evaluation using the ENCRES parameter.

In addition to ENCMOD mode, velocity evaluation on the basis of the encoder must be activated using the ENCSPEED command.

Command	Argument	Function	Description
ENCMOD	-	Encoder Mode	Change to encoder mode (not for MCDC) An external encoder serves as position detector (the current position value is set to 0)
ENCSPEED	-	Encoder as speed sensor	Speed via encoder signals in encoder mode
HALLSPEED	-	Hall sensor as speed sensor	Speed via hall sensors in encoder mode
ENCRES	Value	Load Encoder Resolution	Load resolution of external encoder (4 times pulse/rev). **Value: 8 ... 65 535**

Additional settings

Movement limits

The LL command can also be used to define a movement range limit for velocity mode. The APL1 command activates monitoring of these limits.

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Current limitation

The current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.6.3 „Current controller and I²t current limitation“).

Motion control commands

An overview of all motion control commands is given in Chapter 7.4 „Motion control commands“.

Command	Argument	Function	Description
EN	-	Enable Drive	Activate drive
DI	-	Disable Drive	Deactivate drive
V	Value	Select Velocity Mode	Activate velocity mode and set specified value as target velocity (velocity control). Unit: rpm

Example:

- Drive motor at 100 rpm: `V100`

  In order to change the direction of rotation, simply assign a negative velocity value (e.g. V-100).
- Stop motor: `V0`

NOTE

Make sure that APL0 is set, if you do not want the drive to stop at the set range limits (LL)!

Also check that the maximum speed SP is not set below the desired target velocity.

3 Functional Description

3.2 Velocity control

Complex motion profiles

Reaching the given speed is indicated by a “v”, if “Notify Velocity“ has been set before starting the speed mode and ANSW1 or ANSW2 is set:

Command	Argument	Function	Description
NV	Value	Notify Velocity	A “v” is returned when the nominal speed is reached or passed through. **Value: –32 767 … 32 767**
NVOFF	-	Notify Velocity Off	Velocity command that has not yet been triggered is deactivated again.

3 Functional Description

3.3 Homing and limit switches

Guide

Homing on limit switches can be used to re-initialise the absolute position of an application after switching on.

After switching on, or by giving the GOHOSEQ command, previously defined homing is performed up to the set limit switch and then the actions defined for it are performed. The ramp generator settings for maximum acceleration and the movement limits are taken into account.

3.3.1 Limit switch connections and switching level

The connections

- AnIn
- Fault
- 3rd input
- 4th, 5th input (MCDC only)

can be used as reference and limit switch inputs.

In BL motors the zero crossing of the Hall sensor signals is also available as index pulse. The index pulse occurs once or twice per revolution depending on the motor type (two or four pole). The index pulse of an external encoder can also be connected to the fault pin, enabling the actual position to be exactly zeroed.

The AnIn and Fault connections are designed as interrupt inputs, which means that they are edge-triggered. All other inputs are not edge-triggered, so that the signal must be at least 500 µs to be reliably detected. The maximum reaction time to level changes at all inputs is 500 µs.

Digital input configuration

Command	Argument	Function	Description
SETPLC	-	Set PLC inputs	Digital inputs PLC-compatible (24 V level) Low: 0 V … 7.0 V High: 12.5 … V UB
SETTTL	-	Set TTL inputs	Digital inputs TTL-compatible (5 V level) Low: 0 V … 0.5 V High: 3 V … UB
REFIN	-	Reference Input	Fault pin as reference or limit switch input

The limit switch functions for the fault pin are only accepted if REFIN is activated (setting must be saved with SAVE)!

CAUTION!

Configure before applying a voltage

The electronics can be damaged if a voltage is applied to the fault pin while it is not configured as input.

► *Configure the fault pin as input first before applying external voltage!*

3 Functional Description

3.3 Homing and limit switches

3.3.2 Motion control commands

The function of the inputs and the homing behaviour are set using the FAULHABER commands described in Chapter 3.3.3 „Configuration of homing and limit switches“. A previously configured homing is then started with the following FAULHABER commands. An overview of all motion control commands is given in Chapter 7.4 „Motion control commands“.

Command	Argument	Function	Description
GOHOSEQ	-	Go Homing Sequence	Execute FAULHABER homing sequence. A homing sequence is executed (if programmed) irrespective of the current mode.
POHOSEQ	-	Power-On Homing Sequence	Start homing automatically after power-on. 1: Power-On Homing Sequence is activated 0: No homing after power-on
FHIX	-	Find Hall Index	The nearest index pulse in the preset direction of rotation is approached. For BX4 drives only
GOHIX	-	Go Hall Index	Move BL motor to Hall zero point (Hall index) and set actual position value to 0. Not for BX4 and MCDC drives
GOIX	-	Go Encoder Index	Move to the encoder index at the Fault pin and set actual position value to 0 (DC motor or ext. encoder).

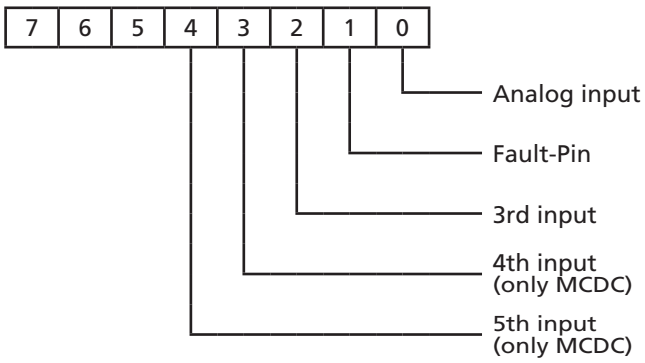
If the drive is already located in the limit switch when GOHOSEQ is invoked, first of all it moves out of the switch, in the opposite direction to that specified for HOSP. The same applies to the Power On Homing Sequence (POHOSEQ).

3 Functional Description

3.3 Homing and limit switches

3.3.3 Configuration of homing and limit switches

The following commands use the following bit mask for configuration of the limit switch functions:



Set or delete the bit at the position of the required input for each command and assign the resulting numeric value to the commands described below.

Polarity and limit switch function

Limit switches can respond to the rising or falling edge (or level).

In addition, the hard blocking function can be configured for the limit switches. The hard blocking function provides reliable protection against overshooting of the range limit switch. If the drive is located in an HB limit switch, then the direction of rotation set with HD will be blocked, i.e. the drive can only move further out of the limit switch.

The speed stays at 0 rpm, if the target velocity is preset in the wrong direction.

Command	Argument	Function	Description
HP	Bit mask	Hard Polarity	Define valid edge and polarity of respective limit switches: 1: Rising edge and high level effective. 0: Falling edge and low level effective.
HB	Bit mask	Hard Blocking	Activate Hard Blocking function for relevant limit switch.
HD	Bit mask	Hard Direction	Presetting of direction of rotation that is blocked with HB of respective limit switch. 1: Clockwise rotation blocked 0: Anticlockwise rotation blocked

Example:

- Setting of the Hard-Blocking function for Fault pin and 4th input: $2^1+2^3 = 2+8 = 10$ ➔ HB10

Definition of homing behaviour

In order to be able to execute a homing sequence with the command GOHOSEQ or as POHOSEQ, a homing sequence must be defined for a specific limit switch! Definition of the hard blocking behaviour is an additional option.

Command	Argument	Function	Description
SHA	Bit mask	Set Home Arming for Homing Sequence	Homing behaviour (GOHOSEQ): Set position value to 0 at edge of respective limit switch
SHL	Bit mask	Set Hard Limit for Homing Sequence	Homing behaviour (GOHOSEQ): Stop motor at edge of respective limit switch.
SHN	Bit mask	Set Hard Notify for Homing Sequence	Homing behaviour (GOHOSEQ): Send a character to RS232 at edge of respective limit switch.

These settings must be saved with SAVE so that they are available immediately after switching on!

Example:

- Homing with 3rd input as reference input (rising edge):
  - **HP4** Low level or falling edge was evaluated at AnIn and at the fault pin, the rising edge is evaluated at the 3rd input.
  - **SHA4** Activate a homing sequence for 3rd input (all others are in bit mask = 0) Action: Set Pos = 0 on reaching the limit switch
  - **SHL4** Activate a homing sequence for 3rd input (all others are in bit mask = 0) Action: Stop motor
  - **SHN4** Activate a homing sequence for 3rd input (all others are in bit mask = 0) Action: Notify via RS232

Homing Speed

Command	Argument	Function	Description
HOSP	Value	Load Homing Speed	Load speed and direction of rotation for homing (GOHOSEQ, GOHIX). Unit: rpm

Example:

- Homing with 100 rpm and negative direction of rotation:

```
HOSP-100
```

Direct programming via HA, HL and HN commands

These special commands can be used to define actions that are to be triggered at an edge of the relevant input, independently of a homing sequence. A programmed limit switch function will remain effective until the preselected edge occurs. The programming can be changed with a new command before an edge occurs.

Command	Argument	Function	Description
HA	Bit mask	Home Arming	Set position value to 0 and delete relevant HA bit at edge of respective limit switch. Setting is not saved
HL	Bit mask	Hard Limit	Stop motor and delete relevant HL bit at edge of respective limit switch. Setting is not saved.
HN	Bit mask	Hard Notify	Send a character to RS232 and delete relevant HN bit at edge of respective limit switch. Setting is not saved.

The settings are not saved with the SAVE command, therefore all configured limit switches are inactive again after power-on.

HL / SHL command:

Positioning mode

When the edge occurs, the motor positions itself on the reference mark with maximum acceleration.

Velocity controller mode

The motor is decelerated at the set acceleration value when the edge occurs, i.e. it goes beyond the reference mark. The reference mark can be precisely approached with a subsequent positioning command (command M).

3 Functional Description

3.3 Homing and limit switches

Advantage:

No abrupt motion changes.

Re. HN-/SHN command:

Hard Notify (HN) and Set Hard Notify (SHN) return values to the RS232 interface:

Connection	Return value
"AnIn"	h
"Fault"	f
"3rd input"	t
"4th input" (MCDC only)	w
"5th input" (MCDC only)	x

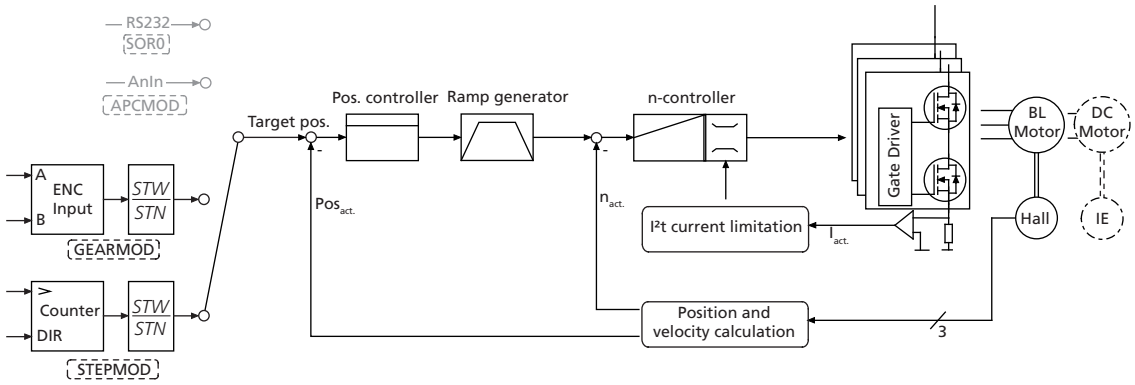
3 Functional Description

3.4 Enhanced operating modes

Use the CONTMOD command to revert from an enhanced operating mode to normal mode.

3.4.1 Stepper motor mode

Controller structure in stepper motor mode



In stepper motor mode the drive moves one programmable angle further for each pulse at the analog input, and thus simulates the function of a stepper motor.

There are a number of considerable advantages in comparison with a real stepper motor:

- The number of steps per revolution is freely programmable and of a very high resolution (encoder resolution)
- The individual step widths are freely programmable
- No detent torque
- The full dynamics of the motor can be used
- The motor is very quiet
- The motor monitors actual position so that no steps are "lost" (even with maximum dynamics)
- No motor current flows in settled state (actual position reached)
- High efficiency

Basic settings

In stepper motor mode, the analog input acts as frequency input. The error output must be configured as rotational direction input if the direction of rotation is to be changed via a digital signal.

Alternatively, the direction of rotation can also be preset via the commands ADL and ADR.

Command	Argument	Function	Description
STEPMOD	-	Stepper Motor Mode	Change to stepper motor mode
DIRIN	-	Direction Input	Fault pin as rotational direction input
ADL	-	Analog Direction Left	Positive voltages at the analog input result in anticlockwise rotation of the rotor
ADR	-	Analog Direction Right	Positive voltages at the analog input result in clockwise rotation of the rotor

3 Functional Description

3.4 Enhanced operating modes

Input

Maximum input frequency: 400 kHz

Level: 5 V TTL or 24 V PLC-compatible, depending on configuration.

The number of steps of the emulated stepper motor can be set to virtually any required settings using the following formula:

$$\text{Revolutions} = \text{pulses} \cdot \frac{STW}{STN}$$

Revolutions … revolutions generated on the drive
Pulses … number of pulses at the frequency input (= number of steps)

Command	Argument	Function	Description
STW	Value	Load Step Width	Load step width for step motor and gearing mode **Value: 1 ... 65 535**
STN	Value	Load Step Number	Load number of steps per revolution for step motor and gearing mode **Value: 1 ... 65 535**

Example:

Motor should turn 1/1000th of a revolution for each input signal:

- `STW1`
- `STN1000`

Additional settings

Movement limits

The range limits set with LL are also active in step motor mode with APL1.

Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Current limitation

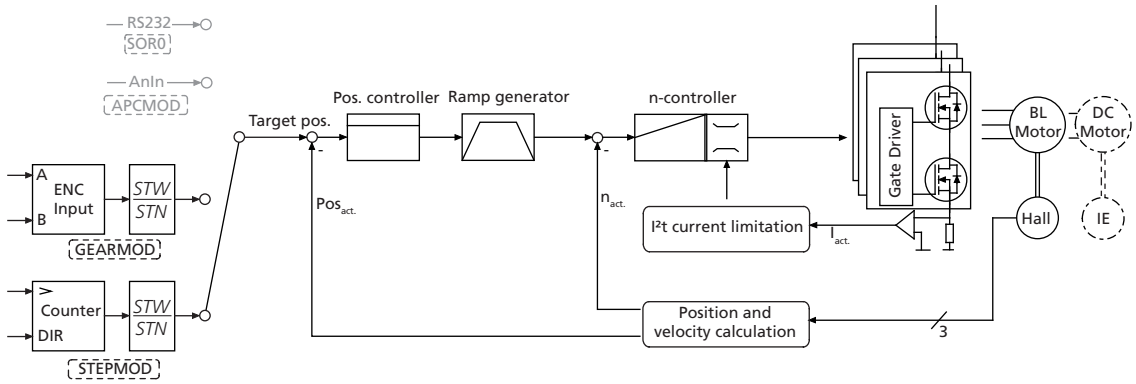
The current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.6.3 „Current controller and I²t current limitation“).

3 Functional Description

3.4 Enhanced operating modes

3.4.2 Gearing mode (electronic gear)

Controller structure in gearing mode



Gearing mode enables the use of an external encoder as set-point source for the position. This enables several drives to be synchronised. If the direction of rotation is to be changed by a digital signal, the function of the fault pin must be reconfigured as a rotational direction input.

Alternatively, the direction of rotation can also be preset via the commands ADL and ADR.

Basic settings

Command	Argument	Function	Description
GEARMOD	-	Gearing Mode	Change to gearing mode
DIRIN	-	Direction Input	Fault pin as rotational direction input

Input

The two channels of an external encoder are connected to connections AnIn and AGND, which may need to be connected to the 5 V encoder supply via a 2.7 kΩ pull-up resistor.

The gear ratio between the pulses per revolution (PPR) count of the external encoder and the resulting movement of the motor can be set using the following formula:

$$\text{Revolutions} = \text{pulses} \cdot \frac{STW}{STN}$$

Revolutions … revolutions generated on the drive
Pulses … actually counted pulses during four edge evaluation

Command	Argument	Function	Description
STW	Value	Load Step Width	Load step width for step motor and gearing mode **Value: 1 … 65 535**
STN	Value	Load Step Number	Load number of steps per revolution for step motor and gearing mode **Value: 1 … 65 535**

3 Functional Description

3.4 Enhanced operating modes

Example:

Motor has to move one revolution at 1 000 pulses of the external encoder:

- **STW1**
- **STN1000**

Additional settings

Movement limits

The range limits set with LL are also active in gearing mode with APL1.

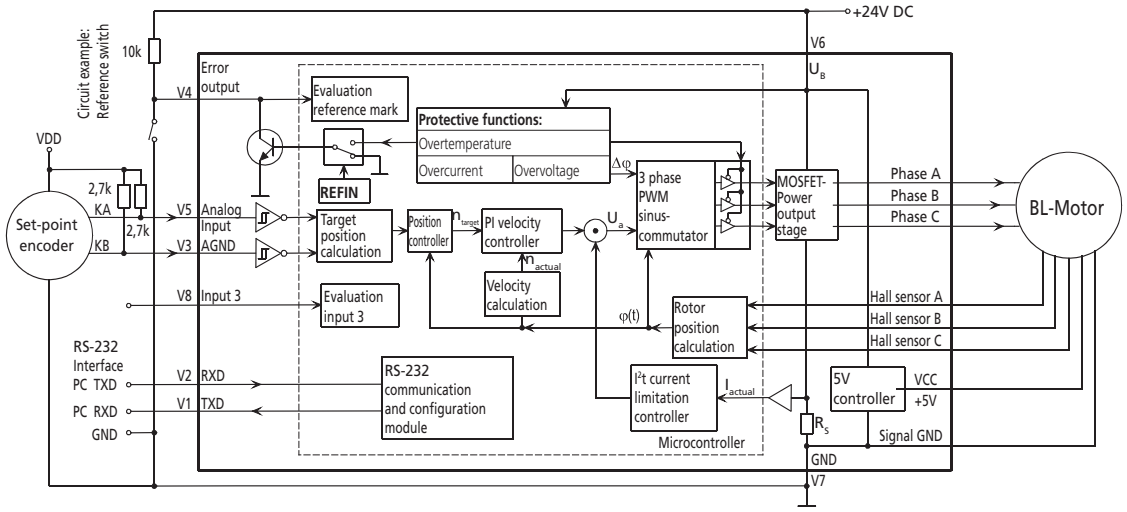
Ramp generator

The slopes of the acceleration and deceleration ramps, and the maximum speed can be defined using the AC, DEC and SP commands (see Chapter 3.6.1 „Ramp generator“).

Current limitation

The current limitation values LPC and LCC can be used to protect the drive against overload (see Chapter 3.6.3 „Current controller and I²t current limitation“).

Circuit example, gearing mode for MCBL 3003/06 S

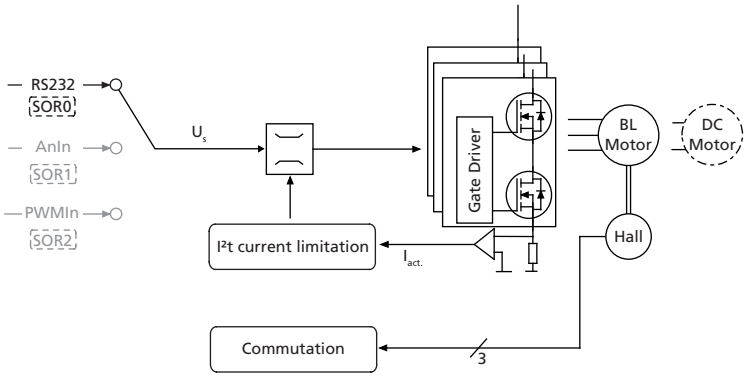


3 Functional Description

3.4 Enhanced operating modes

3.4.3 Voltage regulator mode

Controller structure in voltage regulator mode



In voltage regulator mode a motor voltage is output proportional to the preset value. Current limitation remains active.

With this mode, it is possible to use a higher level controller. The controller then serves only as a power amplifier.

Basic settings

Command	Argument	Function	Description
VOLTMOD	-	Set Voltage Mode	Activate Voltage Regulator Mode
U	Value	Set Output Voltage	Output motor voltage (corresponds to -Uv … +Uv) with SOR0 only **Value: –32 767 … 32 767**

Input

SOR0 (RS232)	SOR1 (AnIn)	SOR2 (PWMIn)	U_{MOT}
U-32767	-10 V	0 %	$-U_B$
U0	0 V	50 %	0
U32767	10 V	100 %	$+U_B$

Additional settings

Current limitation

The current limitation values LPC and LCC can be used to protect the drive against overload.

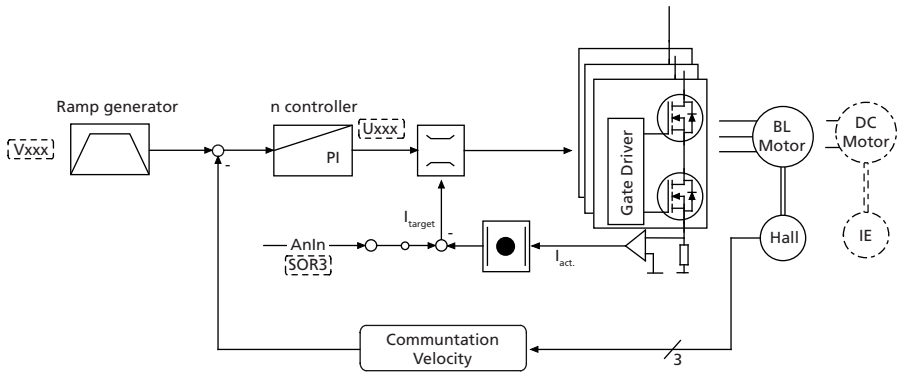
3 Functional Description

3.4 Enhanced operating modes

3.4.4 Current control with analog current presetting

Fixed direction of rotation (SOR3)

Controller structure for analog current presetting with fixed preset direction of rotation



You can switch to analog target current presetting with the SOR3 command. In this way, both in velocity mode and in voltage regulator mode, current amount can be limited proportional to the voltage at the analog input. The set current is weighted with the maximum current LPC.

The motor is activated either in velocity mode by a previously fixed target velocity, or in voltage regulator mode via a voltage value. The error output must be configured as rotational direction input if the direction of rotation is to be changed via a digital signal.

Basic settings

Command	Argument	Function	Description
SOR	3	Source for Velocity	3: Current target value via analog input
LPC	Value	Load Peak Current Limit	Load peak current (mA). **Value: 0 … 12 000**

Input

If 10 V are present at the analog input, the current is accordingly limited to the maximum current set with LPC.

Even if negative voltages are present at the analog input, the current is limited to the amount of the applied voltage. Negative target current presettings therefore have no effect on the direction of rotation!

SOR3 (AnIn)	I_{max}	n_{max}
-10 V	LPC	SP
0 V	0	SP
10 V	LPC	SP

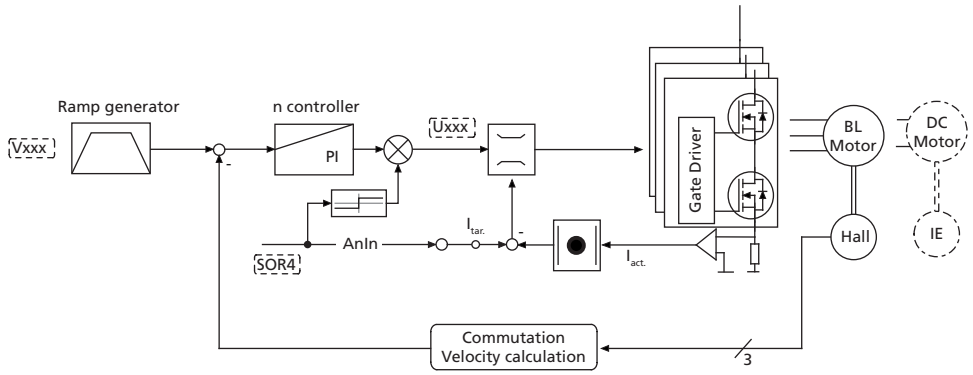
Warning! ***Risk of destruction***

In current control mode with analog current presetting the internal I²t current limitation is deactivated.

3 Functional Description

3.4 Enhanced operating modes

Direction of rotation depending on current target value (SOR4)



You can switch to analog target current presetting with the SOR4 command. In this way, both in velocity mode and in voltage regulator mode, current amount can be limited proportional to the voltage at the analog input. The set current is weighted with the maximum current LPC.

The motor is activated either in velocity mode by a previously fixed target velocity, or in voltage regulator mode via a voltage value. The direction of rotation is determined from the sign of the current target value.

This mode corresponds to direct current control.

Basic settings

Command	Argument	Function	Description
SOR	4	Source for Velocity	4: Target current value via analog input with presetting of the direction of rotation via the sign of the set-point.
LPC	Value	Load Peak Current Limit	Load peak current (mA). **Value: 0 … 12 000**

Input

If 10 V are present at the analog input, the current is accordingly limited to the maximum current set with LPC.

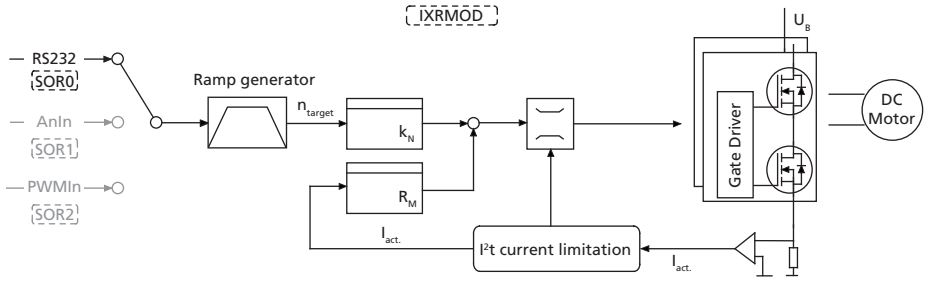
SOR4 (AnIn)	I_{max}	n_{max}
-10 V	LPC	-SP
0 V	0	SP
10 V	LPC	SP

3 Functional Description

3.4 Enhanced operating modes

3.4.5 IxR control for MCDC

Controller structure in IxR mode



For speed-controlled applications with DC motors without an encoder, an IxR control is available on the MCDC. In this mode, the motor speed is determined via an internal motor model. Consequently, the encoder and the associated wiring can be omitted.

However, control quality and accuracy are considerably restricted. This mode is mainly suited for higher speeds and larger motors in the FAULHABER range.

Basic settings

Command	Argument	Function	Description
IXRMOD	-	Set IxR Mode	Activate IxR control (MCDC only)
RM	Value	Load Motor Resistance	Load motor resistance RM according to specification in data sheet. Unit: mΩ
KN	Value	Load Speed Constant	Load speed constant kn in accordance with information in the data sheet. Unit: rpm/V

In stationary mode the following formula applies to the voltage at the DC motor: $U_M = R_M \times I_A + n / k_N$

As a result, at constant terminal voltage U_M the speed falls under load.

Vice versa, if R_M and k_N are known, the voltage applied to the motor can be increased depending on the target velocity and the measured motor current so that the voltage drop is approximately compensated at the winding resistor.

Setting rules

Synchronisation of the no-load speed via k_N.

Synchronisation of the velocity under load via R_M.

- Velocity increases under load: R_M is set too high
- Velocity drops too far under load: R_M is set too low

3 Functional Description

3.5 Special fault output functions

The error connection (fault pin) can be configured as input or output for different tasks:

Command	Function	Description
ERROUT	Error Output	Fault pin as error output (default)
ENCOUT	Encoder Output	Fault pin as pulse output (not MCDC)
DIGOUT	Digital Output	Fault pin as digital output. The output is set to low level.
DIRIN	Direction Input	Fault pin as rotational direction input ▶ Velocity control (see Chapter 3.2 „Velocity control“), ▶ Stepper motor mode (see Chapter 3.4.1 „Stepper motor mode“), ▶ Gearing mode (see Chapter 3.4.2 „Gearing mode (electronic gear)“), ▶ Voltage regulator mode (see Chapter 3.4.3 „Voltage regulator mode“). ▶ Current control with analog current presetting (see Chapter 3.4.4 „Current control with analog current presetting“).
REFIN	Reference Input	Fault pin as reference or limit switch input ▶ Homing and limit switches (see Chapter 3.3 „Homing and limit switches“)
POSOUT	Position Output	Fault pin as output for display of the condition: “target position reached".

Fault pin as error output

In ERROUT mode the output is set as soon as one of the following errors occurs:

- One of the set current limitation values (LPC, LCC) is exceeded
- Set maximum permissible speed deviation (DEV) is exceeded
- Overvoltage detected
- Maximum coil or MOSFET temperature exceeded

Additional settings

Delayed signalling

In order to hide the transient occurrence of errors during the acceleration phase, for example, an error delay can be set which specifies how long an error must be present before it is displayed at the error output:

Command	Argument	Function	Description
DCE	Value	Delayed Current Error	Delayed error output with ERROUT Value in 1/100 sec.

Example:

Wait 2 seconds before displaying error:

- DCE200

Error notification via RS232

If one of the above errors occurs, automatic notification with an “r” can be implemented by setting “Notify Error”, provided ANSW1 or ANSW2 is set:

Command	Argument	Function	Description
NE	0 - 1	Notify Error	Notification in the event of errors 1: An “r” is returned if an error occurs 0: No error notification

3 Functional Description

3.5 Special fault output functions

Fault pin as pulse output (not for MCDC):

In the ENCOUT mode the fault pin is used as pulse output, which outputs an adjustable number of pulses per revolution. The pulses are derived from the Hall sensor signals of the BL motors. Limited to max. 4000 pulses per second in 2 pole motors. Limited to max. 2000 pulses per second in 4 pole motors.

In MCBL 300x RS AES the LPN value is limited to 32.

Command	Argument	Function	Description
LPN	Value	Load Pulse Number	Preset pulse number for ENCOUT. **Value: 1 ... 255** **Value: 1 ... 32 in MCBL AES**

Example:

Output 16 pulses per revolution at the fault pin:

- LPN16

In the case of 5000 rpm, 5000/60 16 = 1333 pulses per second are output.

NOTE

For speeds that would generate more than the maximum possible pulse number at the set LPN value, the maximum number is output. The set pulses are precisely achieved, but the timing does not necessarily have to exactly agree (delays possible).

Position determination via pulse counting is therefore possible, provided that no change occurs in the direction of rotation and the maximum possible pulse number is not exceeded.

Fault pin as digital output

In DIGOUT mode, the error connection can be used as universal digital output. The digital output can be set or cleared via the following commands.

Command	Argument	Function	Description
CO	-	Clear Output	Set digital output DIGOUT to low level
SO	-	Set Output	Set digital output DIGOUT to high level
TO	-	Toggle Output	Switch to digital output DIGOUT

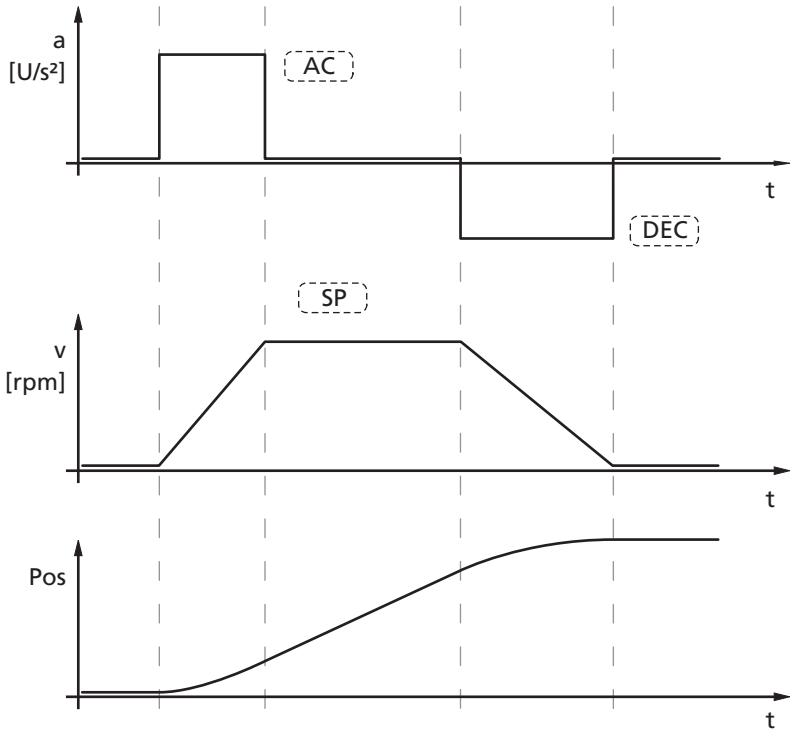
3 Functional Description

3.6 Technical information

3.6.1 Ramp generator

In all modes, apart from voltage regulator mode and current control, the set-point is controlled by the ramp generator.

Basic ramp generator function



This can be used to separately set the parameters for maximum acceleration (AC), maximum delay (DEC) and maximum speed (SP) for specific applications.

Basic settings

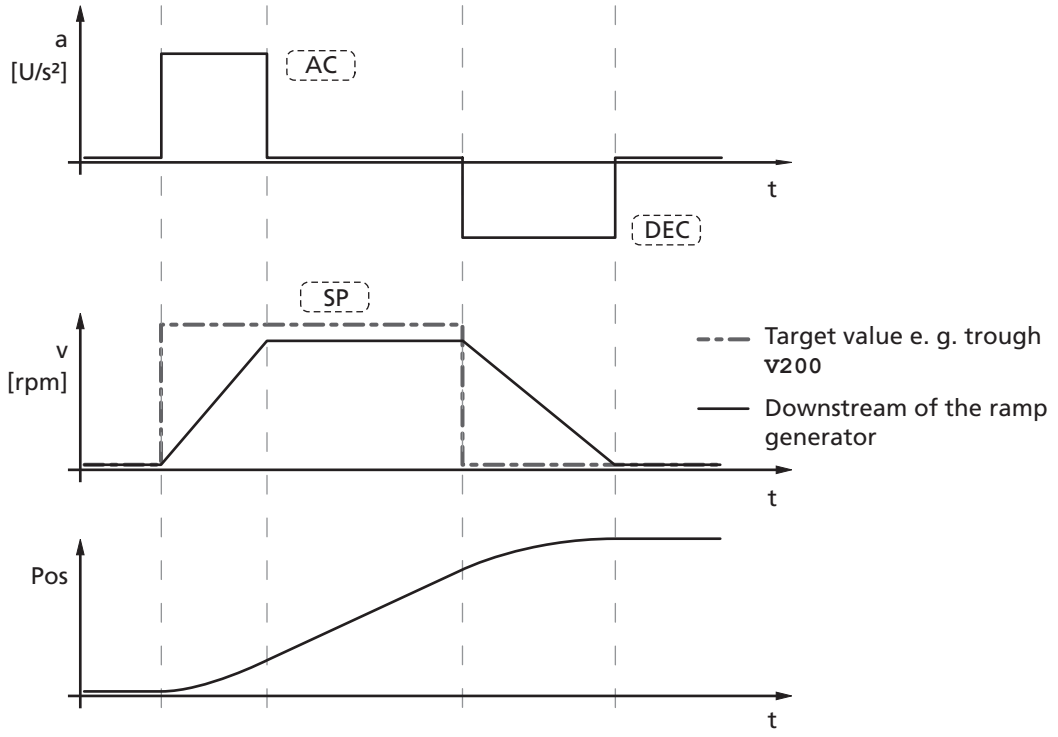
Command	Argument	Function	Description
AC	Value	Load Command Acceleration	Load acceleration value (1/s²). **Value: 0 ... 30 000**
DEC	Value	Load Command Acceleration	Load deceleration value (1/s²). **Value: 0 ... 30 000**
SP	Value	Load Maximum Speed	Load maximum speed (rpm). **Value: 0 ... 30 000**

3 Functional Description

3.6 Technical information

Ramp generator in velocity mode

Intervention of the ramp generator in velocity mode



In velocity mode the ramp generator acts like a filter on the target velocity. The target value is limited to the maximum speed value (SP) and target value changes are limited according to the deceleration and acceleration ramps (AC and DEC).

Notification of the higher level control

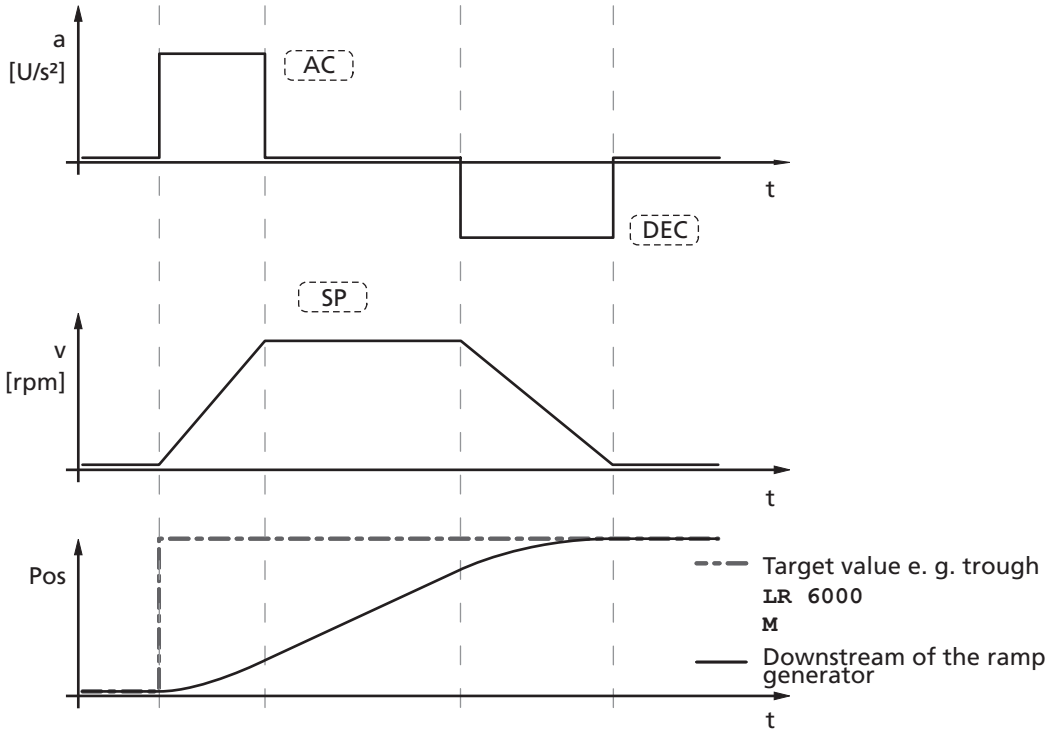
Reaching the given speed is indicated by a “v”, if “Notify Velocity“ has been set before starting the speed mode and ANSW1 or ANSW2 is set.

3 Functional Description

3.6 Technical information

Ramp generator in positioning mode

Intervention of the ramp generator in positioning mode



In positioning mode a preset speed is determined by the position controller from the difference between the target position and actual position.

In the ramp generator, the preset speed output by the position controller is limited to the maximum speed value (SP) and accelerations are limited according to the acceleration ramp (AC).

In positioning mode the deceleration process is not extended as, before reaching the limit position, the speed has to be reduced so that the target position can be reached without overshooting.

According to the equation of motion:

$2a\,s = v^2 \rightarrow v_{max} = \sqrt{2a\,s}$

a: Acceleration [m/s²]

v: Velocity [m/s]

s: remaining distance [m]

the maximum speed max n must be limited proportional to the remaining distance.

The allowable deceleration ramp, or rather the technically possible ramp depending on the motor and the inertia of the load, is set here using the parameter DEC.

Notification of the higher level control

Attainment of the target position or any intermediate position is indicated by a “p” on the serial interface if “Notify Position” is set before the start of positioning, provided that ANSW1 or ANSW2 is set.

Complex motion profiles

More complex motion profiles can be generated through appropriate presetting of new values (maximum speed, acceleration, end position) during positioning.

After a value change, simply execute a new motion start command (M). The commands NP and NV can be used to control the sequence.

The complex profile can be generated either by a higher level control or autonomously via a sequence program. Notes on design of the sequence programs are given in Chapter 6 „Sequence Programs“.

Command	Argument	Function	Description
NP	-/value	Notify Position	Without argument: A “p” is returned when the target position is attained. With argument: A “p” is returned if the specified position is over-travelled. **Value: $-1{,}8 \cdot 10^9$ … $1{,}8 \cdot 10^9$**
NPOFF	-	Notify Position Off	Notify Position command that has not yet been triggered is deactivated again.
NV	Value	Notify Velocity	A “v” is returned when the nominal speed is reached or passed through. **Value: –32 767 … 32 767**
NVOFF	-	Notify Velocity Off	Velocity command that has not yet been triggered is deactivated again.

Example:

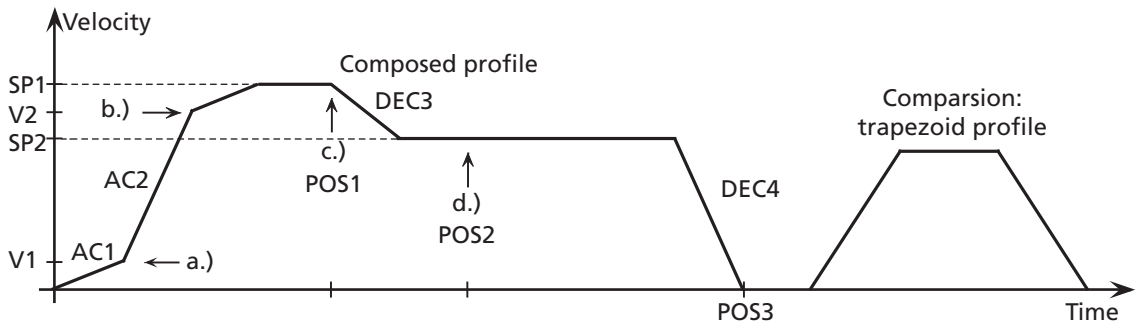
Complex speed profile with notify by the drive

Start	Update a)	Update b)	Update c)	Update d)
LA[POS3]	AC[AC2]	AC[AC1]	SP[SP2]	DEC[DEC4]
AC[AC1]	NV[V2]	NP[POS1]	DEC[DEC3]	NP[POS3]
SP[SP1]	M	M	NP[POS2]	M
NV[V1]			M	
M				

Drive response

V = V1	V = V2	Pos = Pos1	Pos = Pos2	Pos = Pos3
v	v	p	p	p

Example of complex motion profile in comparison with trapezoidal profile:



3 Functional Description

3.6 Technical information

3.6.2 Sinus commutation

The outstanding feature of FAULHABER motion controllers for brushless motors is their so-called sinus commutation. This means that the preset rotating field is always ideally positioned in relation to the rotor. As a result, torque fluctuations can be reduced to a minimum, even at very low speeds. In addition, the motor runs particularly quietly.

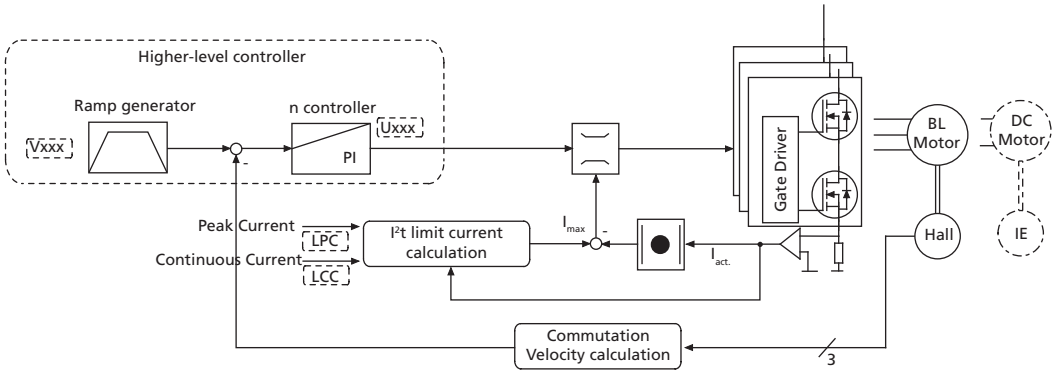
The sinus commutation is further enhanced by so-called flat-top modulation, which enables more modulation. As a result, higher no-load speeds are possible.

The SIN0 command can be used to set the system so that the sinus commutation switches to block commutation in the upper speed range. This full modulation enables the complete speed range of the motor to be utilised.

Command	Function	Description
SIN	Sinus commutation	0: Full control (block mode with full control) 1: Limited to sinusoidal form (basic setting)

3.6.3 Current controller and I²t current limitation

Intervention of the current limiting controller



The FAULHABER Motion Controllers are equipped with an integral current controller, which enables torque limitation.

The current controller operates as a limitation controller. Depending on the previous loading, the I²t current limitation limits to the allowable peak current or continuous current. As soon as the motor current exceeds the currently allowed maximum value the current controller limits the voltage.

Due to its design as a current limiting controller, current control in the thermally relaxed state has no effect on the dynamic of the velocity control. The time response of this limitation can be adjusted using the parameter CI.

The default values for CI limit the current to the allowable value after around 5ms.

3 Functional Description

3.6 Technical information

Basic settings

Command	Argument	Function	Description
LPC	Value	Load Peak Current Limit	Load peak current **Value: 0 … 12 000 mA**
LCC	Value	Load Continuous Current Limit	Load continuous current **Value: 0 … 12 000 mA**
CI	Value	Load Current Integral Term	Load integral term for current controller **Value: 1 … 255**

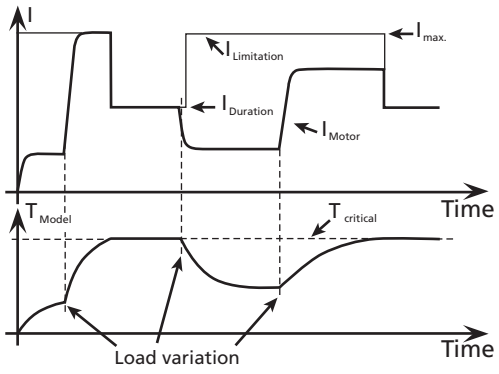
Mode of operation of the current controller

When the motor starts, the peak current is preset as the set-point for the current controller. As the load increases, the current in the motor constantly increases until it finally reaches the peak current. The current controller then comes into operation and limits the current to this set-point.

A thermal current model operating in parallel calculates a model temperature from the actually flowing current. If this model temperature exceeds a critical value, continuous current is switched to and the motor current is regulated to this. Only when the load becomes so small that the temperature falls below the critical model temperature is peak current permitted again.

The aim of this so-called I²t current limiting is not to heat the motor above the thermally allowable temperature by selecting a suitable continuous current. On the other hand, a high load should be temporarily possible in order to enable very dynamic movements.

Function of the I²t current limitation



3 Functional Description

3.6 Technical information

3.6.4 Overtemperature protection

If the MOSFET temperature of the external controllers or the coil temperature of the drives with integrated controller exceeds a preset limit value, the motor is switched off. The following conditions must be fulfilled in order to reactivate the motor:

- Temperature below a preset limit value
- Target velocity set to 0 rpm
- Actual motor speed less than 50 rpm

NOTE

Determining the coil temperature

The housing temperature is measured and the power loss concluded from the current measurement. The MOSFET or coil temperature is calculated from these values via a thermal model. In most applications, this method represents a thermal motor protection device.

3.6.5 Under-voltage monitoring

If the supply voltage falls below the lower voltage threshold, the power stage is switched off. The Motion Controller remains active. When the voltage returns within the permissible range, the power stage is switched on again immediately.

3.6.6 Overvoltage regulation

If the motor is operated as a generator, it produces energy. Usually power supply units are not able to feed this energy back into the power line. For this reason the supply voltage at the motor increases and, depending on the speed, the allowable maximum voltage can be exceeded.

In order to avoid irreparable damage to components, FAULHABER motion controllers for brushless motors contain a controller which adjusts the rotor displacement angle if a limit voltage (32 V) is exceeded. Motion controllers for DC motors contain a ballast circuit which is activated if a limit voltage (32 V) is exceeded. As a result, the energy generated in the motor is converted, and the voltage of the electronics remains limited to 32 V. This method protects the drive during generating operation and rapid braking.

3.6.7 Adjustment of the controller parameters

The controller parameters are already preset for common applications. However, in order to optimally adapt the controller to the respective application, the controller parameters must be optimized. Various theoretical and practical adjustment rules exist, but these will not be described in more detail here. A simple, practical method of adjusting the controller is explained below.

NOTE

Controller sampling rate

The digital controller operates at a sampling rate of 100 µs. When needed the sampling rate can be increased up to 2 ms via the command SR.

3 Functional Description

3.6 Technical information

The following controller parameters are available:

Command	Function	Description
POR	Load Velocity Proportional Term	Load velocity controller amplification. **Value: 1 ... 255**
I	Load Velocity Integral Term	Load velocity controller integral term. **Value: 1 ... 255**
PP	Load Position Proportional Term	Load position controller amplification. **Value: 1 ... 255**
PD	Load Position D-Term	Load position controller D-term. **Value: 1 ... 255**
SR	Load Sampling Rate	Setting of controller sampling rate. **Value: 1 ... 20 ms/10**

Possible procedure

a.) Set parameters of velocity controller:

1.) First of all you have to choose the right sampling rate for the velocity controller depending on the encoder resolution. With less encoder pulses, e.g. 64 pulses per revolution, you need a lower sampling rate, e.g. 1.8 ms = SR18. For higher resolutions, e.g. as exists for all BL motors (3000 pulses per revolution) – the maximum sampling rate can be set to SR1.

Set initial configuration:

- Controller amplification = 8; POR8
- Integral term = 20; I20
- Velocity at 1/3 of the maximum application velocity (example V1000)
- Set acceleration to highest tolerable value of the application (example AC10000)

2.) Increase controller amplification (step width 5, less subsequently); POR 13

3.) Preset velocity jump from 1/3 of maximum speed to 2/3 (example V2000)

4.) Velocity jump from 2/3 to 1/3 and monitor behaviour (example V1000)

5.) Repeat steps 2 to 4, until the controller becomes unstable. Then reduce controller amplification until stability is reliably ensured.

6.) Follow steps 2 to 5 with integral term.

b.) Set parameters of position controller:

1.) Set initial configuration

- Default value for P term: 8; PP8
- Default value for D term: 15; PD15

2.) Motion profiles appropriate for the application must now be specified.

If the system does not function stably with these settings, stability can be achieved by reducing the I term of the velocity controller or reducing the P term of the position controller.

3.) The P term of the position controller can now be increased until the system becomes unstable, in order to optimise the motion profile.

4.) The stability can then be reinstated using the following measures:

- Increasing the D term of the position controller (example: PD20)
- Reducing the I term of the velocity controller

3 Functional Description

3.6 Technical information

Special mode for position control

The SR command can be used to activate a special position control mode. To this end, the value 100 must be added to the required SR setting.

Example:

Required setting SR10 with special mode: `SR110`.

If this mode is activated, the parameter POR is successively reduced in a position-controlled application as soon as the drive is within the target corridor (can be set using the CORRIDOR command). This enables a much „gentler“ stoppage to be achieved after reaching the target position. As soon as the drive is out of the set target corridor position, POR is immediately increased again to the set value.

4 Protocol Description

Guide

An extensive set of ASCII commands is available for configuring and operating FAULHABER Motion Controllers. The structure of the command telegrams is described in the following.

Command frame

The ASCII commands have the following structure:

[Node No.]	Command	[Argument]	CR

The node number is optional and is only required if several drives are being operated on one interface.

The command consists of a letter character string.

The optional argument consists of an ASCII numeric value.

The end is always a CR character (Carriage Return, ASCII decimal code 13). Space characters are ignored, and no distinction is made between upper and lower case.

Response frame

The response to query commands or asynchronous events is also an ASCII character string, followed by a CR character (Carriage Return, ASCII decimal code 13) and an LF character (Line Feed, ASCII decimal code 10).

Response	CR	LF

NOTE

Response in bus mode

The response frames do not contain a node number. In bus mode you must therefore ensure that the response of the contacted node is received before a new command is sent!

Example:

Actual position queries:

- Transmit: POS[CR]
- Receive: 98956[CR][LF]

Drive nodes at 500 rpm:

- Transmit: V500[CR]

4 Protocol Description

Response behaviour settings

As a default, the send commands are not acknowledged. However, the ANSW command can be used to change the response behaviour:

Command	Argument	Function	Description
ANSW	Value	Answer Mode	0: No asynchronous responses 1: Allow asynchronous responses 2: All commands with confirmation and asynchronous responses 3: Debug mode, sent commands are returned 4-7: analogous to 0-3, but responses resulting from a command in the sequence program are not sent (cannot be set via Motion Manager)

If ANSW2 is set, you will receive an "OK" when the command has been successfully executed. If an execution error occurred you will receive one of the following character strings:

- "Unknown command"
- "Invalid parameter"
- "Command not available"
- "Overtemperature – drive disabled"

Example:

- Transmit: `V500[CR]`
- Receive: `OK[CR][LF]`

The SAVE/ EEPSAV command always responds with the character string "EEPROM writing done" after successful saving of the current settings in the data Flash memory, or with "Flash defect", if the save has failed.

CAUTION!

Simultaneous responses

If data is sent simultaneously by several devices, communication disturbance (interference) occurs.

- ▶ *No unaddressed query commands may be sent in network mode* (see Chapter 4.1 „Baud rate and node number")*, as otherwise all units will answer simultaneously and the message frames will mix.*
- ▶ *Asynchronous (sporadic) responses may not be sent simultaneously by several devices.*
- ▶ *Switch off command acknowledgement if using unaddressed send commands.*

Debug mode example:

- Activate debug mode: `ANW3`
- Transmit: `V100`
- Receive: `v,100: OK`

4 Protocol Description

4.1 Baud rate and node number

The serial interface must be configured as follows:

- 8 data bits
- 1 stop bit
- No Parity

The Xon/Xoff protocol must be used for rapid command sequences or transfer of sequence programs and parameter sets.

Baud rate

PC and controllers must be set to the same baud rate to enable them to communicate with each other.

If the baud rate of the controller has been changed, the baud rate of the PC or control must then also be set to the new baud rate.

The setting can be changed via the interface if a connection already exists with the drive node.

Command	Argument	Function	Description
BAUD	Value	Select baud rate	Specify transfer rate for RS232 interface **Value: Baud rate** 600 (not supported by Motion Manager) 1 200 2 400 4 800 9 600 (default) 19 200 38 400 57 600 115 200

Example:

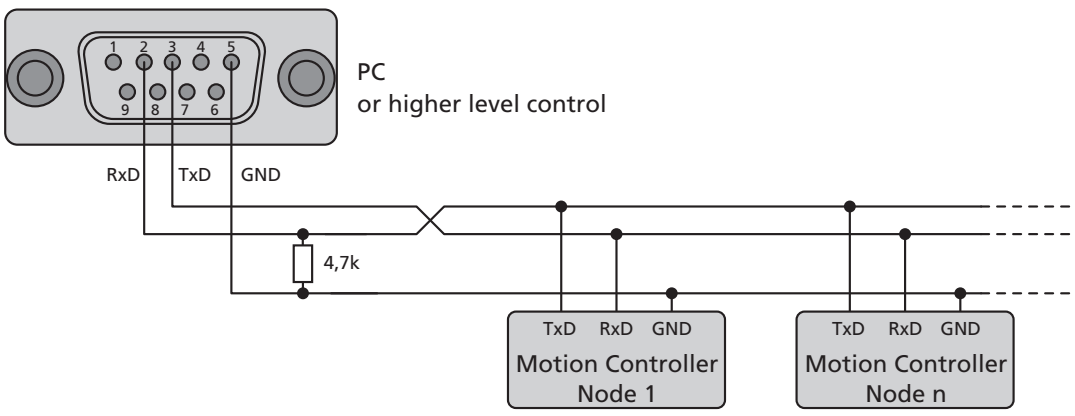
Change transfer rate to 19200 baud:

- BAUD 19200

Serial network and node number

Several motion controllers can be simultaneously operated at a standard RS232 interface. Notes on the wiring are given in the technical manual.

Interconnection of the serial interfaces of the higher level control and motion controllers in the network



4 Protocol Description

4.1 Baud rate and node number

For each drive to be individually actuated at the bus, each drive unit must have a unique node number between 1 and 255.

The devices are all delivered with node number 0. To prepare the units for network operation, they must first be individually connected to the PC and set to the required node address, e.g. with help of the FAULHABER Motion Manager.

In order to address the individual drives in the network, the node number must be specified before each ASCII command to be sent (e.g. 3V100). Commands without a node number are adopted by all drive nodes in the network (Broadcast).

Command	Argument	Function	Description
NODEADR	Value	Define Node Address	Set node number **Value. Node number 1 ... 255**
NET	0-1	Set Network Mode	Activate RS232 multiplex mode for network operation. 0: No network operation, single drive on an RS232 1: Network operation activated

Example:

Set drive unit to node number 3:

- `NODEADR3`

Example:

Activate network operation:

- `NET1`

4 Protocol Description

4.2 Trace Function

An efficient trace function is available via an additional binary interface. This allows up to 2 values to be read out online in a resolution of up to 3 ms.

In order to be able to use the binary interface, it must first have been opened for the desired node with the command BINSEND1.

Command	Argument	Function	Description
BINSEND	0 – 1	Open Binary Interface	1 = Open binary interface 0 = Close binary interface

Trace configuration

1. Setting of binary transmit mode for parameter 1 (curve 1):
   2 binary characters are sent in direct succession: [Command][Mode1]
   The relevant value is switched to, depending on the value of Mode1.
   Command:
   200: Set binary transmit mode for parameter 1
   Mode1:
   0: Actual velocity [Integer16, rpm]
   1: Target velocity [Integer16, rpm]
   2: Controller output [Integer16]
   4: Motor current [Integer16, mA]
   44: Housing temperature [Unsigned16, °C]
   46: Coil temperature [Unsigned16, °C]
   200: Current position [Integer32, Inc]
   201: Target position [Integer32, Inc]
2. Setting of binary transmit mode for parameter 2 (curve 2):
   2 binary characters are sent in direct succession: [Command][Mode2]
   The relevant value is switched to, depending on the value of Mode2.
   Command:
   202: Set binary transmit mode for parameter 2
   Mode2:
   0: Actual velocity [Integer16, rpm]
   1: Target velocity [Integer16, rpm]
   2: Controller output [Integer16]
   4: Motor current [Integer16, mA]
   44: Housing temperature [Unsigned16, °C]
   46: Coil temperature [Unsigned16, °C]
   200: Current position [Integer32, Inc]
   201: Target position [Integer32, Inc]
   255: No second parameter is sent (basic setting for power-on)

4 Protocol Description

4.2 Trace Function

Data request

A binary character is sent: [Request]

Depending on the set modes (Commands 200 and 202), 3,5,7 or 9 bytes are sent back to the PC.

Request: 201: Request a data package

Following a mode adjustment it is necessary to wait for at least 2 ms before requesting valid data.

Received data (after request 201):

1.) Mode1 between 0 and 15, Mode2 at 255 (inactive)

➔ 3 byte … 1st byte: Low byte data

2nd byte: High byte data

3rd byte: Time code

The data are in Integer16 format.

2.) Mode1 between 16 and 199, Mode2 at 255 (inactive)

➔ 3 byte … coding as for 1.)

The data are in Unsigned16 format.

3.) Mode1 between 200 and 255, Mode2 at 255 (inactive)

➔ 5 byte … 1st byte: Lowest byte data

2nd byte: Second byte data

3rd byte: Third byte data

4th byte: Highest byte data

5th byte: Time code

The data are in Integer32 format.

4.) Mode1 corresponding to 1.), 2.) or 3.) and Mode2 less than 255:

➔ 5 - 9 Byte …Byte 1 to 2 (4): Data bytes of Mode1

Byte 3 (5) to 4 (6) (8): Data bytes of Mode2

Byte 5 (7) (9): Time code

The data bytes of Mode2 are coded as for Mode1.

The time code corresponds to a multiple of the time basis of 1 ms and defines the time interval to the last transmission.

5 Commissioning

Guide

To make the basic settings for commissioning, the drive unit must be connected to a PC or higher level control via the serial interface.

NOTE

Connection of the RS232 interface is described in the technical manual.

For the communication setup, ensure that the same transfer rate is set for all nodes (see Chapter 5.2.1 „Connection Parameters“).

FAULHABER Motion Management provides a convenient device configuration option using graphic dialogs.

The configuration can also be carried out using your own PC program, a terminal program or a PCS program.

5 Commissioning

5.1 Basic settings

In the case of external motion controllers, several basic settings have to be made during the initial start-up to adjust the controller to the connected motor.

If drive units are integrated, these basic settings are made in the factory so it is only necessary to adjust to the respective application.

CAUTION! ***Risk of destruction!***

Failure to observe these basic settings can result in destruction of components!

► *The basic settings described in the following must be noted and observed*

The following basic settings must be made fro external motion controllers:

- Motor type or motor data (KN, RM) of the connected motor
- Resolution of an external encoder (ENCRES), if used
- Current limitation values (LCC, LPC), adjusted to the motor type and application
- Controller parameters (POR, I, PP, PD), adjusted to the motor type and application

In addition, FAULHABER Motion Manager can be used to synchronise the Hall sensor signals for smooth start-up and optimisation of the phase angle for the best efficiency.

The configuration must then be adjusted to the respective application for all motion controllers (integrated and external). In particular, the following basic settings are important:

- Mode
- Current limiting values
- Controller parameters
- Function of the digital inputs/outputs

Warning! ***Risk of destruction***

If using the Fault Pin as input (REFIN, DIRIN), the desired function must be programmed before applying external voltage!

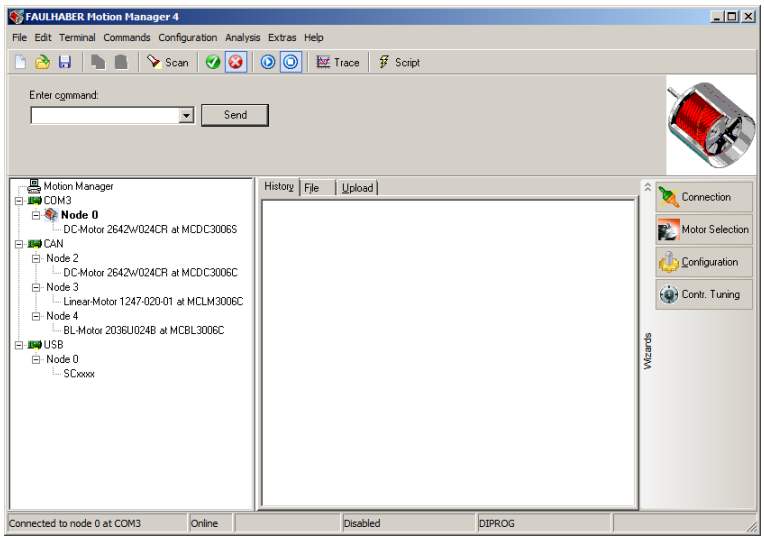
Configuration of these parameters with the help of the FAULHABER Motion Manager is explained in greater detail in the following chapter.

5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

"FAULHABER Motion Manager" PC software provides a simple option for configuring the drive unit and for performing initial tests and optimisation.

The software is available for Microsoft Windows and can be downloaded free of charge from the FAULHABER internet site: www.faulhaber.com.



Motion control systems with electronics built onto the motor are already pre-parameterised in the factory.

Motion controllers with an externally connected motor must be equipped with current limitation values suitable for the motor and suitable controller parameters before being started up.

The motor selection wizard is available for selection of the motor and corresponding suitable basic parameters from V 4.6 and higher of the Motion Manager.

Other settings, e.g. for the function of the fault pin, can be made under the "Configuration – Drive functions" menu item, where a convenient dialog is provided (see Chapter 5.2.3 „Configuration"). The configuration dialog is also available for direct access in the Wizard bar on the right-hand side of the Motion Manager window.

V 4.6 and higher of the Motion Manager also provides a Tuning Wizard, with which the controller parameters of the speed and positioning controller can be adjusted to the application.

5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

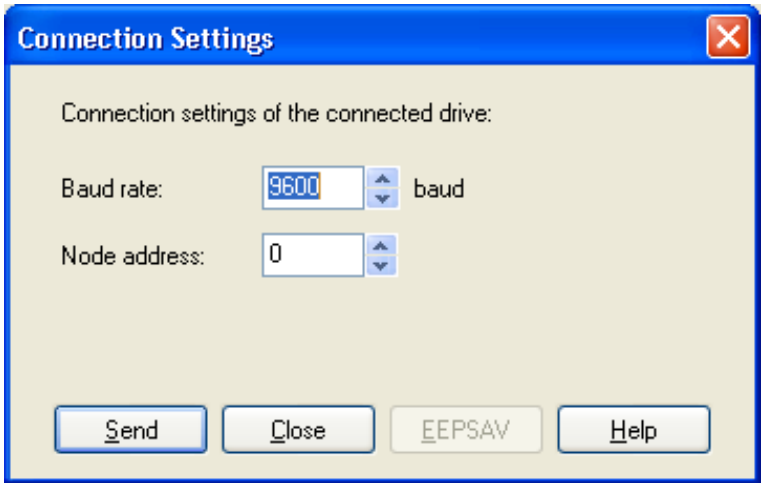
5.2.1 Connection Parameters

If no drive nodes were found after starting the Motion Manager, a connection wizard appears which, following selection of the „Motion Controller with RS232 interface“ product group, can be used to set the PC COM port and the transfer rate. The connection wizard can also be started at any time via the Wizard bar.

Setup wizard (Step 1: Select the product group)

The menu item „Configuration - Connection Parameters…“ can be used to set the transfer rate and node No. of a connected drive unit.

Configuration of the node number and transfer rate



5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

5.2.2 Motor selection

External motion controllers must be adjusted to the connected motor. The motor wizard is provided for this purpose; it can be opened via the Wizard bar at the right-hand edge of the screen.

After selecting the required FAULHABER motor from a list and setting the sensor type used, as well as entering an inertia factor for the load to be operated, in addition to the motor and current limiting values, suitable controller parameters are also determined and transferred to the drive.

5.2.3 Configuration

An extensive configuration dialog is available under the „Configuration – Drive Functions..“ menu item.

No settings are transferred to the drive until the „Send“ button is pressed. The current state of the drive is also read back and the dialog is updated accordingly. Invalid combinations of settings are corrected at the same time, as they are not accepted by the drive.

The settings are permanently saved in the drive using the „EEPSAV“ button.

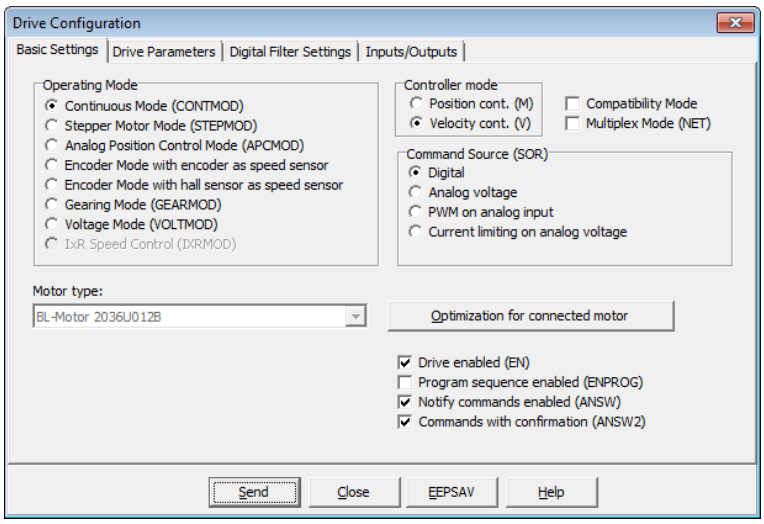
5.2.4 Basic settings

Within the scope of the commissioning,

- the motor type
- the type of operation
- and the type of set-point presetting

are set in the Basic Settings tab.

Basic settings for the motor and encoder type



The Motor Wizard has already set useful default settings for the selected motor/ sensor combination, which can be further refined and adjusted to the required application on the following dialog pages.

5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

Encoder type and optimisation

A button, with which the optimisation wizard can be started, is available for externally connected BL motors with analog Hall sensors for optimisation of Hall sensor signals and the phase angle to the connected motor.

NOTE

Ensure that the motor can freely rotate before starting the encode optimisation.

No special motor adjustments are necessary for DC motors. Apart from setting the current limitation and controller parameters, in this case it is only necessary to set the resolution of a built on incremental encoder in the "Drive Parameters" tab, if not already set via the motor wizard.

Controller mode

Faulhaber motion controllers support both main types of operation

- Position control as servo drive.
- Velocity control

The controller mode is partly automatically selected depending on the chosen operating mode.

Operating mode

In addition to the controller mode, variations of the operation can also be selected.

The following options are available:

CONTMOD

Default setting for the selected controller mode.

For BL motors the actual velocity and actual position in CONTMOD are determined by the motor's Hall sensors.

For DC motors the actual velocity and actual position are determined by the motor's incremental encoder (corresponds to ENCMOD)

CONTMOD for position control: see Chapter 3.1.1 „Set value presetting via the serial interface"

CONTMOD for velocity control: see Chapter 3.2.1 „Velocity presetting via the serial interface"

STEPMOD

Position control

The target position is derived from the number of steps at the AnIn input.

STEPMOD, see Chapter 3.4.1 „Stepper motor mode",

APCMOD

Position control

The target position is preset by an analog voltage at the AnIn input.

APCMOD, see Chapter 3.1.2 „Analog positioning mode (APCMOD)"

ENCMOD with ENCSPEED

Position control or velocity control with evaluation of the external encoder for the actual velocity too

ENCMOD for position control: see Chapter 3.1.3 „External encoder as actual position value (ENCMOD) - not for MCDC"

ENCMOD for velocity control: see Chapter 3.2.3 „External encoder as actual velocity value (ENCMOD) - not for MCDC"

ENCMOD with HALLSPEED

Position control with evaluation of an external encoder and the Hall signals for the actual speed of BL motors

ENCMOD for position control: see Chapter 3.1.3 „External encoder as actual position value (ENCMOD) - not for MCDC“

GEARMOD

Position control

The target position is determined using the number of steps of an external encoder

GEARMOD, see Chapter 3.4.2 „Gearing mode (electronic gear)“

VOLTMOD

Direct presetting of a voltage amplitude at the motor

VOLTMOD, see Chapter 3.4.3 „Voltage regulator mode“

IxRMOD

Velocity control without sensors for DC motors

IxRMOD, see Chapter 3.4.5 „IxR control for MCDC“

Set-point presetting

The set-value presetting must be chosen to match the selected type of operation and controller mode.

The following are supported:

- Presetting via the serial interface or from a sequence program
- Set-point presetting for position or velocity via an analog voltage
- Set-point presetting for position or velocity via a PWM voltage
- Set-point presetting for the limit current via an analog voltage

Power-on state

In the default state the drive's power stage is initially inactive after power-on.

The power stage can be automatically activated after power-on by selecting the „Drive enabled (EN)“ checkbox.

In the default setting, a sequence program is not worked through after the drive is switched on (power-on).

A sequence program stored in the drive can be automatically started immediately after power-on by selecting the „sequence program enabled (ENPROG)“ checkbox.

Communication settings

The „Multiplex mode (NET)“ checkbox is used to activate the selected drive for network mode.

The Asynchronous Responses (ANSW) checkbox can be used to suppress asynchronous responses of the selected drive. They are enabled in the default state.

Use the „Commands with confirmation (ANSW2)“ checkbox to suppress the confirmation frames for the commands sent to the drive. They are activated in the default state.

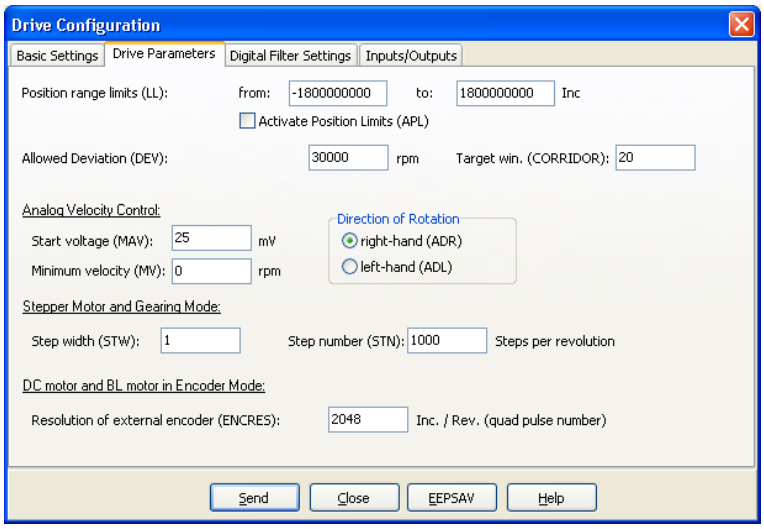
5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

5.2.5 Drive parameters

The Drive Parameters tab is used to make additional settings for the encoder and chosen type of operation.

Additional settings for the chosen type of operation



Encoder resolution

If an incremental encoder attached to the motor is to be evaluated its effective resolution for 4 edge evaluation must be given.

Set-point presetting in stepper or gearing mode

For set-point presetting in stepper mode and in gearing mode the conversion from step count of the external presetting to number of motor revolutions must be given.

Example:

Motor has to perform one revolution at 1000 pulses of the external encoder or at 1000 steps:

- `STW1`
- `STN1000`

Detailed notes on using these parameters are given in the chapters with the functional description of stepper and gearing mode (Chapter 3.4.1 „Stepper motor mode“ and Chapter 3.4.2 „Gearing mode (electronic gear)“).

Velocity presetting via an analog voltage

For presetting a velocity via an analog voltage, a threshold value (MAV) can be preset, from which the target value is evaluated starting with the minimum velocity (MV).

Detailed notes on using this parameter is given in Chapter 3.2.2 „Velocity presetting via an analog voltage or a PWM signal“.

5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

Positioning range limits

In various types of operation the movement range can be monitored and limited. The limits of this movement rage can be given in increments of the actual position using the parameter LL.

Range monitoring is activated by the APL1 command.

Maximum allowable velocity deviation and target corridor

The parameter CORRIDOR defines a range by which the target position within which the „Target position reached“ flag is set. If required, the target position is signalled asynchronously by a notify.

Within this corridor the D term of the position controller is active and the ramp generator is inactive.

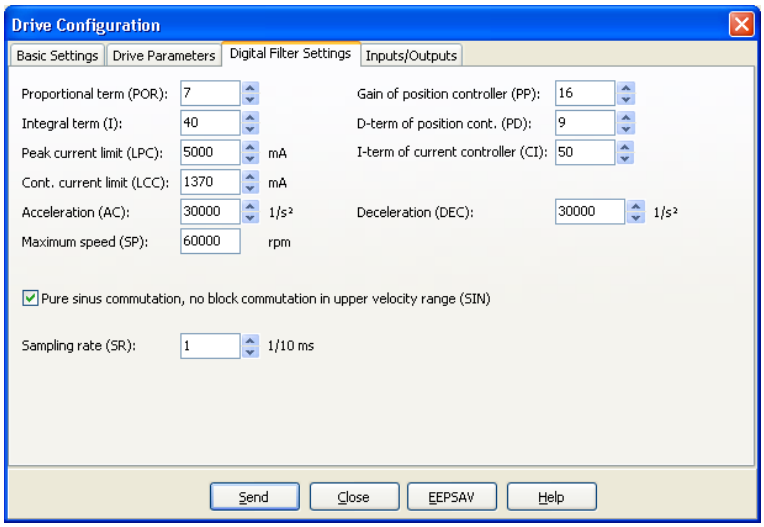
The parameter DEV can be used to preset a maximum allowable controller deviation for the velocity controller. If this barrier is exceeded for longer than set using the parameter DCE in the Inputs and Outputs tab, an error is signalled via the fault pin or on the serial interface.

5.2.6 Controller settings

The changes to the default set controller and current limitation parameters can be made in the „Controller Parameters“ tab of the drive configuration dialog.

In addition, under the „Configuration – Controller Parameters…“ menu item, there is another dialog in which the online parameters can be changed and the result can be observed directly or can be recorded using the trace function in Motion Manager.

Settings for the controller



5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

Voltage output

By default the motion controller for BL motors uses pure sinus commutation. This

means the motor runs with the lowest possible losses and noise.

Alternatively, at higher velocities it is possible to also allow overriding of the output signals similar to block commutation. The maximum achievable velocity is then 7% higher.

NOTE

On changing between pure sinus commutation and operation with block commutation in the upper velocity range the controller amplification also changes by 7%.

Current controller (LCC, LPC, CI)

The parameter LCC can be used to give the thermally allowable continuous current for the application.

Motors and the motion controller can be overloaded within certain limits. Therefore, higher currents can be allowed for dynamic processes. The maximum current value is given by the parameter LPC.

Depending on the drive's load, the internal current monitoring limits the output current to the peak current (LPC) or the allowable continuous current (LCC).

CAUTION!

Risk of destruction!

The thermally allowable continuous current (LCC) should never be given above the thermally allowable continuous current of the motor according to the data sheet.

The maximum peak current (LPC) may never be given above the maximum peak output current of the installed electronics.

The current controller of the motion controller operates as a current limiting controller and therefore in an unlimited case has no effect on the dynamics of the velocity control. The speed of the limitation can be set using the parameter CI. If using the default values for your motor, the current is limited to the allowable value after around 5ms.

If a Faulhaber motor was selected on the basic settings page, parameters are already set here with which the motor can be safely operated.

Further information is given in Chapter 3.6.3 „Current controller and I²t current limitation“.

Velocity controller (I, POR, SR)

The velocity controller is implemented as a PI controller. The sampling time SR can be set as multiples of 100µs, the proportional amplification POR and the integral term I.

If a Faulhaber motor was selected on the basic settings page, parameters are already set here with which the motor can be safely operated.

If the motor is exposed to additional loads, the inertia of the load must be compensated for by a higher proportional term and if necessary slower sampling; in most applications the integral term can remain unchanged.

Further information on setting and adjustment is given in Chapter 3.6.7 „Adjustment of the controller parameters“.

Ramp generator (AC, DEC, SP)

The ramp generator limits the velocity change at the input of the velocity controller using the parameters AC and DEC and the maximum preset speed using the parameter SP.

The parameters AC and SP can be freely selected depending on the application; the parameter DEC is used to specify the deceleration behaviour in positioning mode. For large loads, the deceleration ramp must be limited using the parameter DEC to achieve dead beat (overshoot-free) run-in in the target position.

Further information on setting and adjustment is given in Chapter 3.6.1 „Ramp generator“.

Position controller (PP, PD)

The position controller is implemented as a proportional controller. An additional D term also acts within the target corridor only (see Drive Parameters tab).

The proportional term uses the position deviation in increments to calculate the maximum preset die velocity for the secondary velocity controller. The ramp generator is used to additionally limit the acceleration and maximum velocity.

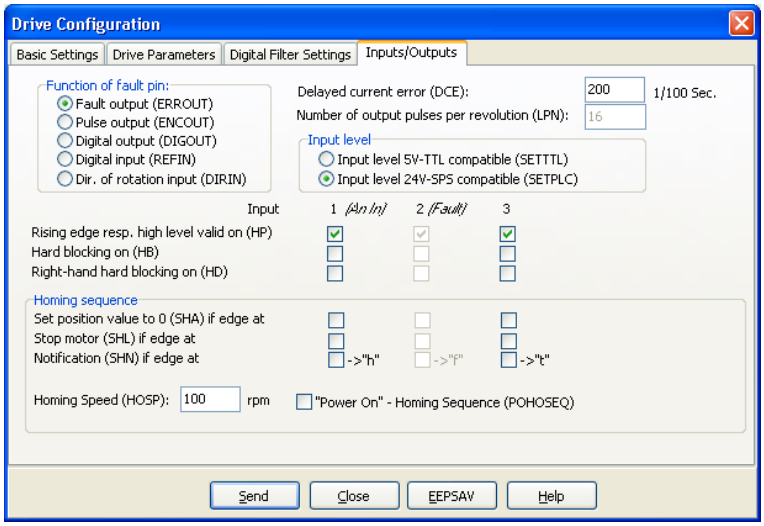
Dead beat run-in in the target position can be preferentially achieved by adjusting the deceleration ramp to the load. For a well-attenuated transient condition in the limit position, the parameter PP must be reduced proportionally to the load inertia.

Further information on setting and adjustment is given in Chapter 3.6.7 „Adjustment of the controller parameters“.

5.2.7 I/O protective circuit and use

The function of the inputs and outputs and homing can be defined in the „Inputs/Outputs“ tab of the drive configuration dialog.

Configuration of the inputs and outputs



Input level and edge

The switching thresholds of the digital inputs are either directly TTL compatible or are adjusted to the switching level of 24V PCS outputs.

In addition, it is also possible to select which level is to be used as the active level for each input and to what extent the input is to be used as a limit switch (HB/HD).

Function of the fault pin

The fault pin can be used both as an input and as an output.

CAUTION!

Do not connect 24V to the fault pin, if the fault pin is configured as a digital output (ERROUT / DIGOUT / ENCOUT)!

The other settings for the 2nd input can only be made if the fault pin is configured as the reference or rotational direction input.

For the default function as a fault output, the parameter DCE can be used to specify a delay time to suppress the response to individual short overcurrent pulses.

For the function as pulse output, the number of pulses per revolution of the motor can be set using the parameter LPN.

In the POSOUT function the output displays the entry into the target corridor as a digital signal (low means target position is reached).

Homing

Use as a reference switch can be set for each of the available inputs.

To this end, either the actual position can be set to 0 by an edge at the selected input (SHA), the motor can be stopped (SHL) or a message can be set to the higher level control (SHN). The actions can be combined.

Homing defined in this way can be executed by the GOHOSEQ command or automatically after switching on if POHOSEQ is set.

5.2.8 Data set management

Save parameters

The settings of a drive can be saved as a backup or as a file for configuration of other drives.

The Motion Manager offers the option of reading out the current drive configuration and saving it as a parameter file.

Transfer parameters to the drive

Previously saved parameter files can be opened in Motion Manager, edited if necessary and transferred to the drive.

Note: Execute the SAVE or EEPSAV command to permanently save a transferred parameter set in the drive.

5 Commissioning

5.2 Configuration using FAULHABER Motion Manager

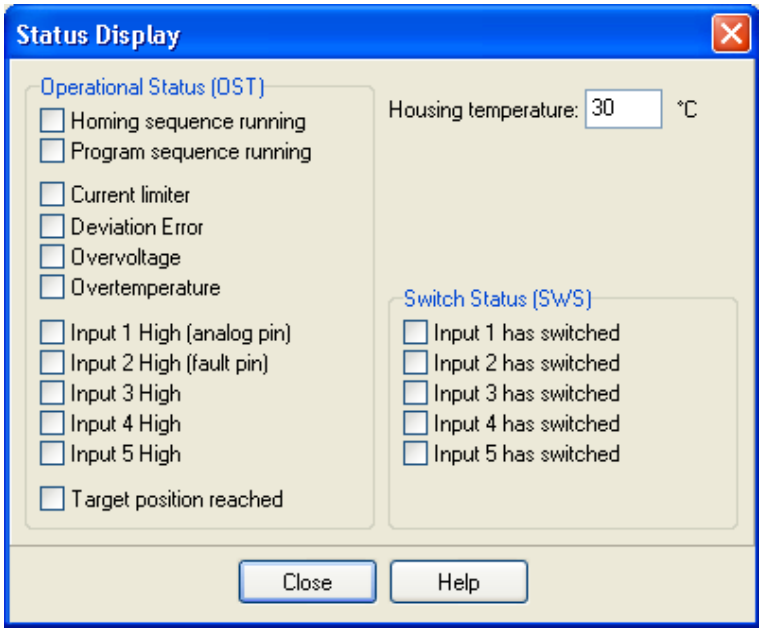
5.2.9 Diagnosis

The status display is used for continuous checking of the main operating states.

Internal states, error flags and the state of the digital inputs are signalled. In addition, the internally measured housing temperature is also displayed here.

The display is updated by Motion Manager by means of cyclical querying of the internal states.

Display of the operating state



Internal states

Partially autonomous states of the motion controller are displayed. These are the course of homing and an active sequence program.

Other internal states are on the one hand the error flag and the housing temperature.

The current limitation flag is set if the maximum current has been set to the continuous current (LCC) by the i^2t monitoring.

States of digital inputs

The state of the digital inputs is displayed as On or Off depending on the level setting

Status of the limit switches

The display indicates whether one of the limit switches has switched, even if the assigned input is already back in the idle state.

Motion Manager provides a trace function as an additional diagnosis tool with which the internal parameters can be graphically recorded. This enables the dynamic behaviour of the drive to be monitored, which is useful, e.g. for optimisation of the controller parameters.

6 Sequence Programs

Sequence programs that are stored directly in the data flash memory of the controller and executed from there can be created for stand-alone applications or for partially autonomous sequences.

The sequence programs can be created and transferred with the FAULHABER Motion Manager, but it is also possible to use a standard text editor and to subsequently transfer the programs with the Motion Manager or a terminal program.

During a program sequence commands can still be sent via the RS232. Almost all ASCII commands can be used in motion programs.

The command PROGSEQ can also be used in the network with a preceding node number. The subsequent command must be send also with a preceding node number. The addressed node stores all received instructions thereby, between the commands PROGSEQ and END.

Command	Argument	Function	Description
PROGSEQ […] END	–	Program Sequence	Defines the start and end of the sequence program. All commands sent to PROGSEQ are not executed, but transferred to the sequence program memory. An END marks the end of the sequence program. All commands after END are directly executed again. There is no SAVE command necessary for saving the program sequence. Command must not be executed more than 10,000 times, as otherwise the function of the Flash memory can no longer be guaranteed. These commands do not have to be entered in the FAULHABER Motion Manager, as they are automatically attached by the "Transfer program file…" function. **Note:** The Xon/Xoff protocol must be used to transfer lengthy program sequences
GPROGSEQ	-/1	Get Program Sequence	Reads out and sends back the stored program sequence. Each program line is output in lower case letters, ending with a CR character. At the end of the program, the "end:" line is sent with details of the program length in bytes followed by a CR and LF character. **GPROGSEQ1:** Reads out the program sequence and indicates at which program line the program counter is currently located ("PC--")
ENPROG	–	Enable Program	Execution of the program is released, i.e. the sequence is started. This status can be permanently stored with SAVE/EEPSAV, so that the drive starts up with the stored program sequence immediately after power-on.
DIPROG	–	Disable Program	Deactivate program execution.
RESUME	–	Resume	Continue program sequence after DIPROG at the point at which it was interrupted.
MEM	–	Memory	Return available program memory in Word.

Control of sequence programs

There are a number of additional commands for controlling programs which are only useful within sequence programs and are consequently only available there.

The following commands stop the sequence until the relevant position is reached:

- NP … Notify Position

  The sequence stops at the next M or V command, until the relevant position is reached.

- HN … Hard Notify

  The sequence stops at the GOHOSEQ command or at the next M or V command, until the limit switch is overtravelled.

6 Sequence Programs

- NV … Notify Velocity

  The sequence stops at the next M or V command, until the relevant speed is reached.

- GOHIX … Go Hall Index

  The sequence stops at the GOHIX command, until the Hall null position is reached.

If there are several Notify conditions, the first fulfilled condition effects continuation of the program.

Additional commands for use within sequence programs:

Command	Argument	Function	Description
DELAY	Value	Delay	Stop sequence for a defined time Argument: in 1/100 seconds **Value: 0 … 65 535**
TIMEOUT	Value	Timeout	With Notify commands, only wait for the specified time and then continue the sequence again. Can also be used via RS232: Send an "o" if Notify condition has not been fulfilled. Argument: in 1/100 seconds **Value: 0 … 65 535**
JMP	Adr	Jump	Jump to specified address. (Can also be used via RS232). **Address: 0 … 255**
JMPGx	Adr	Jump if greater than x	Jump to the specified address if result of last query command is greater than variable x (A, B, C). **Address: 0 … 255**
JMPLx	Adr	Jump if less than x	Jump to the specified address if result of last query command is less than variable x (A, B, C). **Address: 0 … 255**
JMPEx	Adr	Jump if equal to x	Jump to specified address if result of last query command is equal to variable x (A, B, C). **Address: 0 … 255**
JPH	Adr	Jump if Hard-Input activated	Jump to the specified address if the analog input is active (HP determines the polarity). **Address: 0 … 255**
JPF	Adr	Jump if Hard-Input activated	Jump to the specified address if the Fault Pin input is active (HP determines the polarity). Fault Pin must be configured as input (REFIN). **Address: 0 … 255**
JPT	Adr	Jump if 3rd input activated	Jump to the specified address if the 3rd input is active (HP determines the polarity). **Address: 0 … 255**
JPD (MCDC only)	Adr	Jump if 4th input activated	Jump to the specified address if the 4th input is active (HP determines the polarity). **Address: 0 … 255**
JPE (MCDC only)	Adr	Jump if 5th input activated	Jump to the specified address if the 5th input is active (HP determines the polarity). **Address: 0 … 255**
SETx	Value	Set Variable x	Set variable x (A, B, C) to the specified value. Value: Int32 Without argument: Result of last query command is loaded into the variable. **Value: –2 147 483 648 … 2 147 483 647**
GETx	–	Get Variable x	Query content of variable x (A, B, C).
ADDx	Value	Add to Variable x	Add or subtract variable x (A, B, C) with given value. **Value: –2 147 483 648 … 2 147 483 647**
SETARGx	–	Set argument	Set value of variable x (A, B, C) as argument for the next command (if no argument is given there).
DxJNZ	Adr	Decrement x, Jump if not Zero	Decrease the value of variable x (A, B, C) by one and jump to specified address if the value is not 0. **Address: 0 … 255**
ERI	Adr	Error Interrupt	An error interrupt is activated from execution of this command. This means that if an error subsequently occurs (overvoltage, current limitation, …), then the sequence branches to the specified address. The error handling mode is ended if a JMP or RETI command is executed. **Address: 0 … 255**

6 Sequence Programs

Command	Argument	Function	Description
RETI	-	Return Error Interrupt	Return from an error handling routine. **Important:** the interrupted command is not continued, even if it was not completed at the time of interruption!
DIERI	-	Disable Error Interrupt	The ERI command is deactivated, i.e. in the event of an error the program does not jump to the error handling routine.
CALL	Adr	Call Subroutine	Call a subroutine at specified address. Address: 0…255
RET	-	Return from Subroutine	Return from a subroutine. Please note that only one subroutine level is possible, i.e. no subroutines can be called within subroutines!
A	Adr	Define Address	Definition of current position as entry address for jump commands. Address: 0…255

Response behaviour settings

As a default, the send commands are not acknowledged. However, the ANSW command can be used to change the response behaviour:

Command	Argument	Function	Description
ANSW	Value	Answer Mode	0: No asynchronous responses 1: Allow asynchronous responses 2: All commands with confirmation and asynchronous responses 3: Debug mode, sent commands are returned 4-7: analogous to 0-3, but responses resulting from a command in the sequence program are not sent (cannot be set via Motion Manager)

Explanations of the commands and functions

Jump commands

The program sequence can be specifically controlled with the jump commands.

The JMP command can also be used from the RS232. This is useful in cases where different program routines are to be called from the computer.

Example:

```
A1
JMP1          ;Endless loop
A2            ;Program sequence 2 (can only be called by JMP2 from the RS232)
LA10000
NP
M
JMP1          ;Return to endless loop
A3            ;Program sequence 3 (can only be called by JMP3 from the RS232)
LA-10000
NP
M
JMP1          ;Return to endless loop
```

The program sequences according to A2 or A3 can only be called by a JMP2 or JMP3 command from the RS232. A JMP2 from the RS232 results in the drive moving to position 10 000 and stopping there.

6 Sequence Programs

The DxJNZ commands serve to form loops with a predefined number of cycles.

Example:

Move by the same relative position 5 times.

```
SETA5       ;Set variable A to the value 5
A2          ;Define jump address 2
LR100       ;Load relative position
NP          ;Notify Position
M           ;Start positioning
DAJNZ2      ;Decrease A by 1 and jump to address 2, provided that variable A is not yet 0.
```

The commands JPH, JPF and JPT enable jumps that are only executed if the relevant input is active. This means that programs can be called via external switches.

The commands JMPGx, JMPLx, JMPEx enable jumps that refer to the result of the last query command.

Example:

```
SETA 100
GN
JMPLA3
```

The command JMPLA3 jumps to address 3 if the velocity value returned with GN is less than 100 rpm (value of variable A).

Entry addresses are defined via command A. In the case of a jump, the sequence is continued at this point.

The value range for jump commands extends from 0 to 255. Accordingly, a maximum of 256 different entry points can be defined with JMP, JPx, ERI and CALL.

Error Interrupt

During execution of the ERI command, nothing happens initially. Only if an error situation subsequently occurs does the sequence jump immediately to the specified address. This enables sensible continuation of the program in the event of error.

The RETI command enables you to return to the position at which the sequence was interrupted. Please note that the interrupted command is no longer executed, but is continued with the next command.

No new error interruption can take place within the error handling routine. The error handling status is cancelled as soon as the RETI or JMP command is executed. After this, the commands are interrupted again if an error occurs. It should therefore be ensured that the error situation disappears in the error handling routine. Otherwise, the error handling call will be repeated.

Homing

The HN/SHN command enables you to stop the sequence until the limit switch is reached. In order to correctly execute the GOHOSEQ command within a sequence, it is essential to set the SHN command accordingly when defining the homing sequence. This is necessary particularly if you wish to use the Power-On Homing sequence (POHOSEQ1).

6 Sequence Programs

Notify commands

Notify commands enable you to generate complicated motion profiles.

Example:

```
LA100000
SP5000
AC50
NV1000
M
AC100
NV2000
M
AC50
NP
M
```

With this sequence, the acceleration is increased during boot-up at 1 000 rpm. It is decreased again at 2 000 rpm.

The NP command without argument stops the sequence until the target position is reached.

The CALL command

The CALL command enables subroutines to be called from different points, any number of times. You can only jump back from a subroutine again with the RET command.

All commands are permitted within a subroutine except for a repeated CALL command.

General information

If a sequence program is completely processed (no jump at the end of a program), then an “n” is sent to the RS232, if ANSW1 or ANSW2 is set.

In order to generate an endless program (useful for standalone operation), a jump command is required at the end of the program.

Memory size

The sequence programs are stored in binary coding in the Flash memory; 2 bytes are stored for each command, and 0 to 4 bytes for the argument. The maximum memory size available for sequence programs is 6 656 bytes (3 328 words).

6 Sequence Programs

Example:

Positioning routines called via RS232.

The program enables the calling of different routines from the RS232 interface:

- JMP2: Homing Sequence. First move to a limit switch and then to the Hall sensor zero point (Hall index), in order to obtain the most precise reference point possible.
- JMP3: Move to position 0 and stop there.
- JMP4: Attempt to approach a position with low current limitation. As there may be an obstacle in the way in the application, the target position may not be attained. The motor should be stopped after 5 seconds, in any event. (Further evaluation occurs in the higher level control).
- JMP5: 1 000 cycles with following sequence: 10 revolutions forwards, 1 second pause, 5 revolutions back again and then 0.5 seconds pause.

Configuration:

```
SOR0          ;Digital velocity presetting via RS232
LR0           ;Set current position as target position
M             ;Switch to position control (Motion 0)
SHA1          ;Homing Sequence with Notify at AnIn
SHN1
SHL1
HOSP200       ;Homing speed 200 rpm
HP1           ;Rising edge at limit switch effective
ENPROG        ;Start motion program after power-on
ANSW0         ;No asynchronous responses
EEPSAV        ;Save configuration
```

Program:

```
A1
JMP1          ;Endless loop
A2            ;Entry point for homing sequence (JMP2)
GOHOSEQ       ;Homing to reference switch
GOHIX         ;Subsequent homing to Hall sensor zero point (Hall index)
JMP1          ;Return to endless loop
A3            ;Entry point for routine 1 (JMP3)
LA0           ;Set target position to 0
NP            ;Notify at target position (sequence stops until target position is reached)
M             ;Start positioning
JMP1          ;Return to endless loop
A4            ;Entry point for routine 2 (JMP4)
LPC500        ;Set current limitation values to 500 mA (continuous current # peak current)
LA1000000
NP
```

6 Sequence Programs

```
TIMEOUT500 ;Continue sequence after 5 sec., even if position has not yet been attained
M          ;Start positioning
V0         ;Stop motor
LR0
M          ;Switch back to positioning mode
JMP1       ;Return to endless loop
A5         ;Entry point for routine 3 (JMP5)
SETA1000   ;Predefine variable A
A6         ;Entry point for loop
LR30000
NP
M
DELAY100
LR-15000
NP
M
DELAY50
DAJNZ6     ;Repeat loop 1000 times
JMP1       ;Return to endless loop
```

The individual routines are called from the serial interface by sending the commands “JMP2”, “JMP3”, etc.

If the sequence is to wait until the end of a motion command (M, GOHOSEQ, etc.), a Notify (NP or SHN1 in the Homing Sequence configuration) must be set first of all.

7 Parameter Description

Guide

All ASCII commands that are available for operation of the FAULHABER Motion Controllers are listed below.

The structure of the ASCII commands is explained in Chapter 4 „Protocol Description“.

7.1 Basic setting commands

The commands listed here are used for the configuration of basic setting parameters.

7.1.1 Commands for special operating modes

Command	Argument	Function	Description
SOR	0 – 4	Source for Velocity	Source for velocity presetting 0: Serial interface (default) 1: Voltage at analog input 2: PWM signal at analog input 3: Current target value via analog input 4: Target current value via analog input with presetting of the direction of rotation via input polarity
CONTMOD	–	Continuous Mode	Switch back to normal mode from an enhanced mode
STEPMOD	–	Stepper Motor Mode	Change to stepper motor mode
APCMOD	–	Analog Position Control Mode	Change to position control via analog voltage
ENCMOD	–	Encoder Mode	Change to encoder mode (not for MCDC) An external encoder serves as position detector (the current position value is set to 0)
HALLSPEED	–	Hall sensor as speed sensor	Speed via Hall sensors in encoder mode (not for MCDC)
ENCSPEED	–	Encoder as speed sensor	Speed via encoder signals in encoder mode (not for MCDC)
GEARMOD	–	Gearing Mode	Change to gearing mode
VOLTMOD	–	Set Voltage Mode	Activate Voltage Regulator Mode
IXRMOD	–	Set IxR Mode	Activate IxR control (MCDC only)

7 Parameter Description

7.1 Basic setting commands

7.1.2 Parameters for basic setting

Command	Argument	Function	Description
ENCRES	Value	Load Encoder Resolution	Load resolution of external encoder (4 times pulse/rev). **Value: 8 … 65 535**
KN	Value	Load Speed Constant	Load speed constant K_n in accordance with information in the data sheet. Unit: rpm/V. **Value: 0 … 16 383**
RM	Value	Load Motor Resistance	Load motor resistance RM according to specification in data sheet. Unit: mΩ. **Value: 10 … 320 000**
STW	Value	Load Step Width	Load step width for step motor and gearing mode **Value: 1 … 65 535**
STN	Value	Load Step Number	Load number of steps per revolution for step motor and gearing mode **Value: 1 … 65 535**
MV	Value	Minimum Velocity	Presetting of minimum velocity in rpm for specification via analog voltage (SOR1, SOR2) **Value: 0 … 30 000**
MAV	Value	Minimum Analog Voltage	Presetting of minimum start voltage in mV for presetting speed via analog voltage (SOR1, SOR2) **Value: 0 … 10 000**
ADL	–	Analog Direction Left	Positive voltages at the analog input result in anticlockwise rotation of the rotor (SOR1, SOR2)
ADR	–	Analog Direction Right	Positive voltages at the analog input result in clockwise rotation of the rotor (SOR1, SOR2)
SIN	0 – 1	Sinus commutation	1: No block commutation within the upper velocity range (default) 0: Block commutation within the upper velocity range (full modulation) (not for MCDC)
NET	0 – 1	Set Network Mode	Activate RS232 multiplex mode for network operation. 0: No network operation, single drive on an RS232 1: Network operation activated
BAUD	Value	Select baud rate	Specify transfer rate for RS232 interface For value, see Chapter 4.1 „Baud rate and node number“
NODEADR	Value	Define Node Address	Set node number **Value: 0 … 255**
ANSW	0 – 7	Answer Mode	0: No asynchronous responses 1: Allow asynchronous responses 2: All commands with confirmation and asynchronous responses 3: Debug mode, sent commands are returned (cannot be used if configuring with Motion Manager!) 4-7: analogous to 0-3, but responses resulting from a command in the sequence program are not sent (cannot be set via Motion Manager)
POLNUM	2,4	Pole Number	Number of magnetic poles of the connected motor (not for MCDC) 2: Two-pole motor 4: Four-pole motor (e.g. BX4)
SENSTYP	4	Sensor type	Setting of the connected AES encoder. (only for MCBL AES) 4: AES-4096 Further types available on request

7 Parameter Description

7.1 Basic setting commands

7.1.3 General parameters

Command	Argument	Function	Description
LL	Value	Load Position Range Limits	Load limit positions (the drive cannot be moved out of these limits). Positive values specify the upper limit and negative values the lower. The range limits are only active if APL1 is. **Value: –1.8 · 10^9 … 1.8 · 10^9**
APL	0 – 1	Activate/Deactivate Position Limits	Activate range limits (LL) (valid for all operating modes except VOLTMOD). 1: Position limits activated 0: Position limits deactivated
SP	Value	Load Maximum Speed	Load maximum speed. Setting applies to all modes (rpm). **Value: 0 … 30 000**
AC	Value	Load Command Acceleration	Load acceleration value (1/s²). **Value: 0 … 30 000**
DEC	Value	Load Command Deceleration	Load deceleration value (1/s²). **Value: 0 … 30 000**
SR	Value	Load Sampling Rate	Load sampling rate of the velocity controller as a multiple of the basic controller sampling rate according to the data sheet.**Value: 1 … 20**
POR	Value	Load Velocity Proportional Term	Load velocity controller amplification. **Value: 1 … 255**
I	Value	Load Velocity Integral Term	Load velocity controller integral term. **Value: 1 … 255**
PP	Value	Load Position Proportional Term	Load position controller amplification. **Value: 1 … 255**
PD	Value	Load Position Differential Term	Load position controller D-term. **Value: 1 … 255**
CI	Value	Load Current Integral Term	Load integral term for current controller. **Value: 1 … 255**
LPC	Value	Load Peak Current Limit	Load peak current (mA). **Value: 0 … 12 000**
LCC	Value	Load Continuous Current Limit	Load continuous current (mA). **Value: 0 … 12 000**
DEV	Value	Load Deviation	Load maximum permissible deviation of actual velocity from target velocity (deviation) **Value: 0 … 30 000**
CORRIDOR	Value	Load Corridor	Window around the target position. **Value: 1 … 32 767**

7.1.4 Configuration of fault pin and digital inputs

Command	Argument	Function	Description
ERROUT	–	Error Output	Fault pin as error output
ENCOUT	–	Encoder Output	Fault pin as pulse output (not for MCDC):
DIGOUT	–	Digital Output	Fault pin as digital output. The output is set to low level.
POSOUT	–	Position Output	Fault pin as digital output for display of the condition: “target position reached".
DIRIN	–	Direction Output	Fault pin as rotational direction input
REFIN	–	Reference Input	Fault pin as reference or limit switch input
DCE	Value	Delayed Current Error	Delayed error output for ERROUT in 1/100 sec. **Value: 0 … 65 535**
LPN	Value	Load Pulse Number	Preset pulse number for ENCOUT. **Value: 1 … 255**
CO	–	Clear Output	Set digital output DIGOUT to low level
SO	–	Set Output	Set digital output DIGOUT to high level
TO	–	Toggle Output	Switch to digital output DIGOUT
SETPLC	–	Set PLC inputs	Digital inputs PLC-compatible (24 V level)
SETTTL	–	Set TTL inputs	Digital inputs TTL-compatible (5 V level)

7 Parameter Description

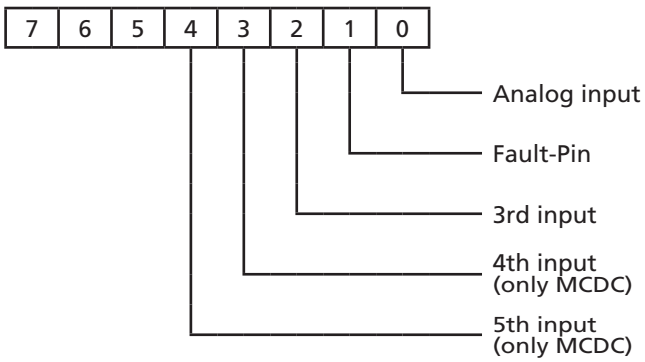
7.1 Basic setting commands

7.1.5 Configuration of homing and limit switches in

Command	Argument	Function	Description
HP	Bit mask	Hard Polarity	Define valid edge and polarity of respective limit switches: 1: Rising edge and high level effective. 0: Falling edge and low level effective.
HB	Bit mask	Hard Blocking	Activate Hard Blocking function for relevant limit switch.
HD	Bit mask	Hard Direction	Presetting of direction of rotation that is blocked with HB of respective limit switch. 1: Clockwise rotation blocked 0: Anticlockwise rotation blocked
SHA	Bit mask	Set Home Arming for Homing Sequence	Homing behaviour (GOHOSEQ): Set position value to 0 at edge of respective limit switch.
SHL	Bit mask	Set Hard Limit for Homing Sequence	Homing behaviour (GOHOSEQ): Stop motor at edge of respective limit switch.
SHN	Bit mask	Set Hard Notify for Homing Sequence	Homing behaviour (GOHOSEQ): Send a character to RS232 at edge of respective limit switch.
HOSP	Bit mask	Load Homing Speed	Load speed and direction of rotation for homing (GOHOSEQ, GOHIX, GOIX). **Value: –30 000 … 30 000 rpm**
POHOSEQ	0 – 1	Power-On Homing Sequence	Start homing automatically after power-on. 0: No homing after power-on 1: Power-On Homing Sequence is activated
HA	Bit mask	Home Arming	Set position value to 0 and delete relevant HA bit at edge of respective limit switch. Setting is not saved.
HL	Bit mask	Hard Limit	Stop motor and delete relevant HL bit at edge of respective limit switch. Setting is not saved.
HN	Bit mask	Hard Notify	Send a character to RS232 and delete relevant HN bit at edge of respective limit switch. Setting is not saved.

Bit mask of the limit switches

The resulting decimal value must be transferred to the commands given here.



7 Parameter Description

7.2 Query commands for basic setting

7.2.1 Operating modes and general parameters

Command	Argument	Function	Description		
CST	–	Configuration Status	Set operating mode. Return value is binary encoded (LSB = Bit 0): Bit 0, reserved Bit 1-2, automatic responses 0: ANSW0 (no automatic responses) 1: ANSW1 (asynchronous responses) 2: ANSW2 (additional command acknowledgements) 3: ANSW3 (Debug) Bit 3-5, Velocity presetting: 0: SOR0 (RS232 interface) 1: SOR1 (Analog voltage) 2: SOR2 (PWM signal) 3: SOR3 (current limitation value) 4: SOR4 (current limitation value with presetting of rotational direction via input polarity) Bit 6, reserved Bit 7-9, FAULHABER mode: 0: CONTMOD 1: STEPMOD 2: APCMOD 3: ENCMOD/HALLSPEED 4: ENCMOD/ENCSPEED 5: GEARMOD 6: VOLTMOD 7: IXRMOD Bit 10, power amplifier: 0: Disabled (DI) 1: Enabled (EN) Bit 11, position controller: 0: Switched off 1: Switched on Bit 12, analog direction of rotation: 0: ADL 1: ADR Bit 13, Position Limits APL: 0: deactivated 1: activated Bit 14, sinus commutation SIN: 0: Allow block commutation 1: Do not allow block commutation Bit 15, network operation 0: NET0 (Single device on an RS232) 1: NET1 (Multiplex mode activated)		
GMOD	–	Get Mode	MCDC	MCBL	Set FAULHABER mode
			D	c	CONTIMOD
			S	s	STEPMOD
			A	a	APCMOD
			–	h	ENCMOD
			–	e	ENCSPEED
			G	g	GEARMOD
			V	v	VOLTMOD
			I	–	IxRMOD
GENCRES	–	Get Encoder Resolution	Set encode resolution (ENCRES)		
GKN	–	Get Speed Constant	Motor speed constant Unit: rpm/V (KN)		
GRM	–	Get Motor Resistance	Motor resistance Unit: mΩ (RM).		
GSTW	–	Get Step Width	Set step width (STW)		

7 Parameter Description

7.2 Query commands for basic setting

Command	Argument	Function	Description
GSTN	–	Get Step Number	Set number of steps per revolution (STN)
GMV	–	Get Minimum Velocity	Set minimum velocity in rpm (MV)
GMAV	–	Get minimum analog voltage	Set minimum start voltage value Unit; mV (MAV)
GPL	–	Get Positive Limit	Set positive limit position (LL)
GNL	–	Get Negative Limit	Set negative limit position (LL)
GSP	–	Get Maximum Speed	Set maximum speed in rpm (SP)
GAC	–	Get Acceleration	Set acceleration value Unit: 1/s² (AC)
GDEC	–	Get Deceleration	Set deceleration value in 1/s² (DEC)
GSR	–	Get Sampling Rate	Set sampling rate of the speed controller Unit: ms/10 (SR)
GPOR	–	Get Velocity Proportional Term	Set amplification value of the speed controller (POR)
GI	–	Get Velocity Integral Term	Set integral term of the speed controller (I)
GPP	–	Get Position Proportional Term	Set amplification value of the position controller (POR)
GPD	–	Get Position D term	Set D component of the position controller (PD)
GCI	–	Get Current Integral Term	Set integral term of the current controller (CI)
GPC	–	Get Peak Current	Set peak current Unit: mA (LPC)
GCC	–	Get Continuous Current	Set continuous current in mA (LCC)
GDEV	–	Get Deviation	Set deviation value (DEV)
GCORRIDOR	–	Get Corridor	Set window around the target position (CORRIDOR)
GNODEADR	–	Get Node Address	Set node number (NODEADR)

7 Parameter Description

7.2 Query commands for basic setting

7.2.2 Configuration of fault pin and digital inputs

Command	Argument	Function	Description
IOC	–	I/O Configuration	Set input/output configuration. Return value binary coded (LSB=Bit 0): Bit 0-7, Hard Blocking: 0-31: Function active for input 1-5 Bit 8-15, Hard Polarity: 0-31: Rising edge at input 1-5 Bit 16-23, Hard Direction: 0-31: Clockwise movement blocked at input 1-5 Bit 24, state of digital output: 0: Low 1: High Bit 25, level of digital inputs: 0: TTL level (5V) 1: PLC LEVEL (24 V) Bit 26-28, function of fault pin: 0: ERROUT 1: ENCOUT 2: DIGOUT 3: DIRIN 4: REFIN
GDCE	–	Get Delayed Current Error	Set value of the error output delay (DCE)
GPN	–	Get Pulse Number	Set pulse number (LPN)

7.2.3 Configuration of homing

Command	Argument	Function	Description
HOC	–	Homing Configuration	Set homing configuration. Return value binary coded (LSB = Bit 0): Bit 0-7, SHA setting Bit 8-15, SHN setting Bit 16-23, SHL setting Bit 24, Power-On Homing Sequence 0: deactivated 1: activated (homing after power-on)
GHOSP	–	Get Homing Speed	Set homing speed Unit: rpm (HOSP)

7 Parameter Description

7.3 Miscellaneous commands

Command	Argument	Function	Description
NE	0 – 1	Notify Error	Notification in the event of errors 1: An "r" is returned if an error occurs 0: No error notification
SAVE EEPSAV		Save Parameters	Save current parameters and configuration setting to Flash memory. The drive will also start with these settings when next switched on. **Attention: Command must not be executed more than 10,000 times, as otherwise the function of the Flash memory can no longer be guaranteed.**
RESET		Reset	Restart drive node.
RN		Reset Node	Set application parameters to original values (ROM values) (current, acceleration, controller parameters, maximum speed, limit positions…) Communication parameters, operating mode and hardware configuration are retained
FCONFIG		Factory Configuration	All configurations and values are reset to the standard delivery status. After this command the drive performs a reset. **Attention: Customer-specific factory settings are also lost, programmed sequence programs are retained!** **The command can be executed a maximum 10000 times.**

7.4 Motion control commands

Command	Argument	Function	Description
DI	–	Disable Drive	Deactivate drive
EN	–	Enable Drive	Activate drive
M	–	Initiate Motion	Activate position control and start positioning
LA	Value	Load Absolute Position	Load new absolute target position **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**
LR	Value	Load Relative Position	Load new relative target position, in relation to last started target position. The resulting absolute target position must lie between the values given below. **Value: $-2.14 \cdot 10^9$ and $2.14 \cdot 10^9$**
NP	–/value	Notify Position	Without argument: A "p" is returned when the target position is attained. With argument: A "p" is returned if the specified position is over-travelled.
NPOFF	–	Notify Position Off	Notify Position command that has not yet been triggered is deactivated again.
V	Value	Select Velocity Mode	Activate velocity mode and set specified value as target velocity (velocity control). Unit: rpm
NV	Value	Notify Velocity	A "v" is returned when the nominal speed is reached or passed through. **Value: –30 000 … 30 000**
NVOFF	–	Notify Velocity Off	Velocity command that has not yet been triggered is deactivated again.
U	Value	Set Output Voltage	Output motor voltage (corresponds to -Uv…+Uv) for SOR0 only in VOLTMOD. **Value: –32 767 … 32 767**
GOHOSEQ	–	Go Homing Sequence	Execute FAULHABER homing sequence. A homing sequence is executed (if programmed) irrespective of the current mode.
FHIX	-	Find Hall Index	BL 4-pol Motor auf Hall-Nullpunkt (Hall-Index) fahren und Istpositionswert auf 0 setzen. Bei 4-pol Motoren sind innerhalb einer Umdrehung zwei Hall-Nullpunkte jeweils gegenüber vorhanden. Es wird jeweils der nächstliegende Index angefahren. (nur für BL 4-pol)
GOHIX	–	Go Hall Index	Move BL motor to Hall zero point (Hall index) and set actual position value to 0 (nur für BL 2-pol)
GOIX	–	Go Encoder Index	Move to the encoder index at the Fault pin and set actual position value to 0 (DC motor or ext. encoder).
HO	–/value	Define Home Position	Without argument: Set actual position to 0. With argument: Set actual position to specified value. **Value: $-1.8 \cdot 10^9$ … $1.8 \cdot 10^9$**

7 Parameter Description

7.5 General query commands

Command	Argument	Function	Description
GTYP	–	Get Controller Type	Query designation (name) of the controller
GSER	–	Get Serial Number	Query the serial number
VER	–	Get Version	Current software version
POS	–	Get Actual Position	Current actual position
TPOS	–	Get Target Position	Target position
GV	–	Get Velocity	Current target velocity in rpm
GN	–	Get N	Current target velocity in rpm
GU	–	Get PWM Voltage	Set PWM value in VOLTMOD
GRU	–	Get Real PWM Voltage	Current controller output value
GCL	–	Get Current Limit	Current limitation current in mA
GRC	–	Get Real Current	Current actual current in mA
TEM	–	Get Temperature	Current housing temperature in °C
GADV	Value	Get Analog Voltage	Read out the voltage applied at the given input (value). 1: Voltage at AnIn 3: Voltage at 3rd In 4: Voltage at 4th In (MCDC only) 5: Voltage at 5th In (MCDC only) Scaling: 1000 digits = 1 V **Return value input 1: –10 000 … 10 000** **Return value input 3, 4, 5: 0 … 10 000**
OST	–	Operation Status	Display current operating status. Return value binary coded (LSB=Bit 0): Bit 0: Homing running Bit 1: Program sequence running Bit 2: Program sequence stopped because of DELAY command Bit 3: Program sequence stopped because of NOTIFY command Bit 4: Current limitation active Bit 5: Deviation error Bit 6: Overvoltage Bit 7: Overtemperature Bit 8: Status input 1 Bit 9: Status input 2 Bit 10: Status input 3 Bit 11: Status input 4 Bit 12: Status input 5 Bit 13 – 15: Reserved for further inputs Bit 16: Position attained Bit 17: Limitation to continuous current
SWS	–	Switch Status	Temporary limit switch settings. Return value binary coded (LSB=Bit 0): Bit 0-7: HA setting Bit 8-15: HN setting Bit 16-23: HL setting Bit 24-31: Information which limit switch has already switched (is reset on resetting the respective input)

7 Parameter Description

7.6 Commands for sequence programs

Commands for generating and executing sequence programs:

Command	Argument	Function	Description
PROGSEQ […] END	–	Program Sequence	Defines the start and end of the sequence program. All commands sent to PROGSEQ are not executed, but transferred to the sequence program memory. An END marks the end of the sequence program. All commands after END are directly executed again. There is no SAVE command necessary for saving the program sequence. Command must not be executed more than 10,000 times, as otherwise the function of the Flash memory can no longer be guaranteed. These commands do not have to be entered in the FAULHABER Motion Manager, as they are automatically attached by the “Transfer program file…” function. **Note:** The Xon/Xoff protocol must be used to transfer lengthy program sequences
GPROGSEQ	-/1	Get Program Sequence	Reads out and sends back the stored program sequence. Each program line is output in lower case letters, ending with a CR character. At the end of the program, the "end:" line is sent with details of the program length in bytes followed by a CR and LF character. **GPROGSEQ1:** Reads out the program sequence and indicates at which program line the program counter is currently located ("PC--")
ENPROG	–	Enable Program	Execution of the program is released, i.e. the sequence is started. This status can be permanently stored with SAVE/EEPSAV, so that the drive starts up with the stored program sequence immediately after power-on.
DIPROG	–	Disable Program	Deactivate program execution.
RESUME	–	Resume	Continue program sequence after DIPROG at the point at which it was interrupted.
MEM	–	Memory	Return available program memory in Word.

7 Parameter Description

7.6 Commands for sequence programs

Additional commands for use within sequence programs:

Command	Argument	Function	Description
DELAY	Value	Delay	Stop sequence for a defined time Argument: in 1/100 seconds **Value: 0 … 65 535**
TIMEOUT	Value	Timeout	With Notify commands, only wait for the specified time and then continue the sequence again. Can also be used via RS232: Send an "o" if Notify condition has not been fulfilled. Argument: in 1/100 seconds **Value: 0 … 65 535**
JMP	Adr	Jump	Jump to the given address (can also be used via RS232). **Address: 0 … 255**
JMPGx	Adr	Jump if greater than x	Jump to the specified address if result of last query command is greater than variable x (A, B, C). **Address: 0 … 255**
JMPLx	Adr	Jump if less than x	Jump to the specified address if result of last query command is less than variable x (A, B, C). **Address: 0 … 255**
JMPEx	Adr	Jump if equal to x	Jump to specified address if result of last query command is equal to variable x (A, B, C). **Address: 0 … 255**
JPH	Adr	Jump if Hard-Input activated	Jump to the specified address if the analog input is active (HP determines the polarity). **Address: 0 … 255**
JPF	Adr	Jump if Hard-Input activated	Jump to the specified address if the Fault Pin input is active (HP determines the polarity). Fault Pin must be configured as input (REFIN). **Address: 0 … 255**
JPT	Adr	Jump if 3rd input activated	Jump to the specified address if the 3rd input is active (HP determines the polarity). **Address: 0 … 255**
JPD (MCDC only)	Adr	Jump if 4th input activated	Jump to the specified address if the 4th input is active (HP determines the polarity). **Address: 0 … 255**
JPE (MCDC only)	Adr	Jump if 5th input activated	Jump to the specified address if the 5th input is active (HP determines the polarity). **Address: 0 … 255**
SETx	Value	Set Variable x	Set variable x (A, B, C) to the specified value. **Value: Int32** Without argument: Result of last query command is loaded into the variable. **Value: –2 147 483 648 … 2 147 483 647**
GETx	–	Get Variable x	Query content of variable x (A, B, C).
ADDx	Value	Add to Variable x	Add or subtract variable x (A, B, C) with given value. **Value: –2 147 483 648 … 2 147 483 647**
SETARGx	–	Set argument	Set value of variable x (A, B, C) as argument for the next command (if no argument is given there).
DxJNZ	Adr	Decrement x, Jump if not Zero	Decrease the value of variable x (A, B, C) by one and jump to specified address if the value is not 0. **Address: 0 … 255**
ERI	Adr	Error Interrupt	An error interrupt is activated from execution of this command. This means that if an error subsequently occurs (overvoltage, current limitation, …), then the sequence branches to the specified address. The error handling mode is ended if a JMP or RETI command is executed. **Address: 0 … 255**
RETI	–	Return Error Interrupt	Return from an error handling routine. **Important:** the interrupted command is not continued, even if it was not completed at the time of interruption!
DIERI	–	Disable Error Interrupt	The ERI command is deactivated, i.e. in the event of an error the program does not jump to the error handling routine.
CALL	Adr	Call Subroutine	Call a subroutine at specified address. **Address: 0 … 255**
RET	–	Return from Subroutine	Return from a subroutine. Please note that only one subroutine level is possible, i.e. no subroutines can be called within subroutines!
A	Adr	Define Address	Definition of current position as entry address for jump commands. **Address: 0 … 255**

FAULHABER

MA7000.05029 English, 2nd issue, 12.2011

DR. FRITZ FAULHABER GMBH & CO. KG
Antriebssysteme

Daimlerstraße 23/25
71101 Schönaich · Germany
Tel. +49(0)7031/638-0
Fax +49(0)7031/638-100
info@faulhaber.de
www.faulhaber.com