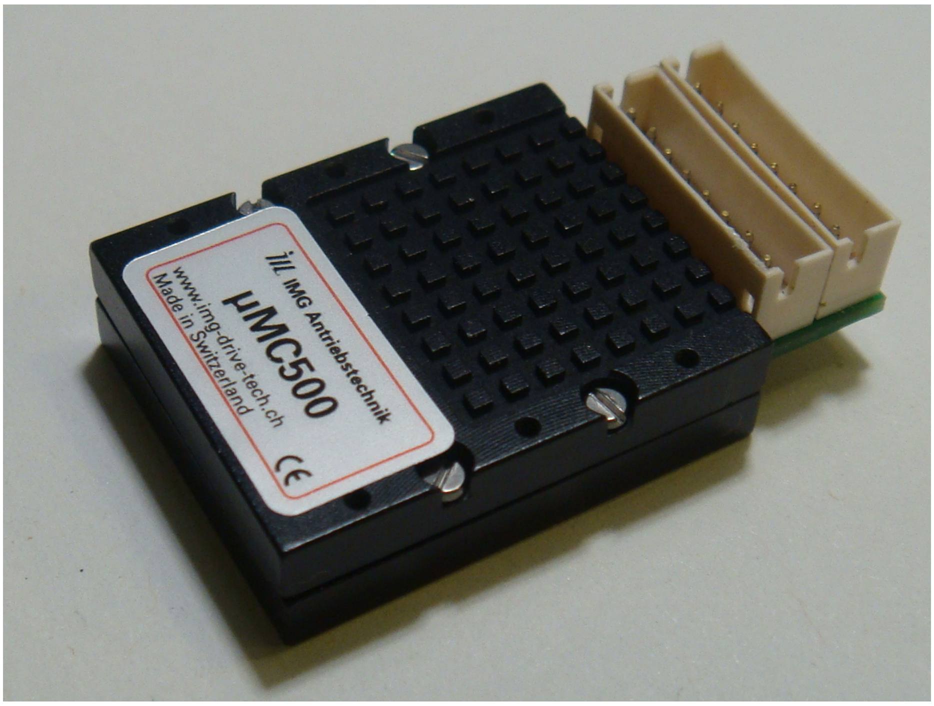
IMG Antriebstechnik GmbH

Zweiackerstrasse 67 CH-8053 Zürich Telefon / Fax: +41 44 381 03 74

www.img-drive-tech.ch

µMC500
Firmware Manual V6.00



Contents **Page**

1. Preface

The µMC500 comprises in its compact dimensions of no more than 31 x 18 x 7.1mm a complete single axis drive controller for small DC servo drives with a maximum of 24W electrical power.

3 different acceleration and deceleration ramps, positioning tracks up to +-31bit, a speed control with a dynamic range or +-14bit, 4 different current regulation and limitation modes in addition to adjustment options for all relevant parameters enable the integration of the µMC500 in virtually every system using a single-axis drive controller. Up to 9 of these drive controllers can be addressed individually via the RS485 bus. Utilizing this, in many cases systems with multiple axes can be synchronized.

The innovative CPM motion control engine generates its own space vector commutation by means of continuous positioning control of the drive. This results in the fact that this system does not need a separate speed controller. During positioning actions there is no need for a speed estimator for the acceleration and deceleration ramp, the required momentary speed for target runs is directly derived from the momentary position. This substantially facilitates parameter setting of the controller and gives the system intrinsic stability (patented principle).
Based on its operating principle the CPM motion control engine integrated in the µMC500 can also handle linear moving coil drives.

Further operating modes integrated in the µMC500:
- Stepper motor emulation in the shape of pulse and rotating direction signals on the inputs Index0 und Index1.
- Positioning to 8 freely definable target positions by means of pulse trains on input Index1; pulse timing freely definable.
- Controlled (rotating) speed in both directions with definable ramps and speeds by means of static signals on input Index0 und Index1.
- IxR-compensated speed control without incremental encoder (firmware version –A only)

Firmware update: The firmware of the µMC500 can be updated individually even within an RS485 network with several µMC500sconnected together, without the need of taking further measures (in-system / in-application). The boot loader accepts addressed, unencrypted HEX strings as well as encrypted CHX strings; see chapter 3, PC software STM7000.

We gladly develop customized firmware versions, customer-specific designs or adapted connector systems on request. The µMC500 can be configured in customer-specific versions as mere current regulator, e.g. for a magnetic powder brake, or can be configured as intelligent I/O module with power output stage, etc.

A short overview of features:

- Maximum positioning distance:	32 bit (-2147483648...2147483647)
- Dynamics speed controller:	1:+-16384
- Lowest speed:	10 increments/s
- Highest speed:	163840 increments/s
- Speed tolerance:	max. 2% static, 0% at load change
- Sampling rate positioning controller:	2,560 kHz
- Dynamics acceleration/deceleration ramps:	1:255 in 2 ranges
- Ramp profiles:	total 3: linear, square, none
- Index or zero point search:	5 different methods
- Status display:	2 LEDs
- Motor current regulation/limitation:	50mA...800mA in steps of 0.683mA
- Multistage power output and motor protection:	Short-circuit protection of the motor outputs against each other as well as against ground and supply, tracking error and overdrive monitoring, 4 current regulation modes
- Serial interface:	RS485 full and half duplex, UART (3.3 / 5V) 19200 or 115200Bd, 8 data, 1 stop, no parity as well as I^2C
- Module and device addresses on RS485 bus:	max. 9 (device addresses 1...9)
- Supply incremental encoder:	integrated, 5V / max. 50mA load

2. Command structure and responses on RS485 and UART

The commands exclusively consist of printable ASCII characters, which can be found on every keyboard. All commands can also be sent by means of a terminal program to a PC or host computer. The responses displayed on the terminal window appear as readable text.

2.1 Command structure for parameter writing

nks=xxxxxxxxx<LF>

<LF>=0x0a, whereas <CR>=0x0d will be ignored or aborted, accordingly.

Example: 4G0=200<LF> writes the rated speed value with 200 (200*10 = 2000 increments/s)
4X1=109471<LF> writes a position of +109471 increments relative to the current position

n: Device or module address, ASCII 1...9
k: Command group, ASCII – upper case letters A...Z likewise a..z
s: Sub index of the command group, ASCII – decimal figure 0..9 and upper case letters A...Z
x: Parameter value, ASCII – decimal numerous value with up to 10 digit and plus/minus sign

On each valid command with valid address there will be an immediate response, which is structured as follows:

na<LF>

Example: 4*<LF>

n: Device or module address (ASCII 1...9) of the responding module
a: "*" or "!" or "#", whereas * = command accepted, # = erroneous parameter (out of permissible value range) and ! = command cannot be accepted at the time, e.g. ramp is repeatedly defined new or ramp range with more than one nesting level. If no drive controller connected to the network corresponds to the sent address, or if the sent command is invalid, there will be no response.

2.2 Command structure for parameter reading

nks?<LF>

<LF>=0x0a, whereas <CR>=0x0d will be ignored or aborted, accordingly.

Example: 4G0?<LF> reads the rated speed value and sends e.g. 4:200<LF> as response
4Q0?<LF> reads the status 0 and sends e.g. 4:10010110<LF> as response

These examples are representative for reading out most of the parameters described in this manual, i.e. almost all commands explained in this manual can be read out by means of the question mark “?” instead of the equal character “=”, which is used for writing.

Sequencing
The individual commands are not accumulated in a ring buffer or run as a sequence. This implies that for each command there will be an immediate response (acknowledge) in the format described above. A new command can only be sent to one of the drive controllers in the RS485 network after the host computer has received this individual response.

Parameter of the serial interface:
- 19200 or 115200Bd (parameter setting via PC software STM7000)
- 8 data bits
- 1 start bit
- 1 stop bit
- No parity
- No XON/XOFF

3. PC software STM7000

The PC software STM7000 available for our drive controllers enables the user to easily access all parameters, status values and I/O signals. Furthermore, freely definable target runs can be performed, the firmware can be updated, parameter sets can be saved or several drive controllers connected to the RS485 interface can be identified and addressed individually.

All commands can also be sent to the PC or host computer via terminal program, the responses displayed on the terminal window also appear as readable text

System requirements to run the STM7000:

- Windows 2000, XP, Vista, and Win7.
- 10MB free memory on HD
- Free COM1....COM8 port with RS485 full or half duplex - (4 - or 2 wire) interface or USB connection, respectively

RS485 interface

The RS485 interface can be realized with a USB to RS485 adapter or it can be an RS485 interface installed in the host computer. Since the µMC500 features a full duplex interface with half duplex ability (switching of data direction), a simple RS232 to RS485 converter without data direction switching mode can be connected. By means of this device, merely the TxD/RxD signals of approx.. +-10V of the RS232 interface are converted to differential 5V- A/B und Y/Z signals of the RS485 interface.

USB to RS485, as well as RS232 to RS485 adapters are produced by a variety of manufacturers and are made commercially available by most distributors.
Manufacturers: ADLINK, Spectra, Exsys, Trigress Security, Moxa, Athen, Maxxtro, IPC-DAS, etc.

For integration of an RS-485 interface in a custom designed host controller there is a large selection of RS485 transceivers available from different manufacturers: Fairchild Semiconductor, TI, ON Semiconductor, Linear Technology, Analog Devices, Maxim, etc., to mention only a few of them.

4. Ramp profiles and operating modes

Within the 9 different operating modes in the µMC500 the user can make a choice between 3 ramp profiles for positioning tasks, continuous speed control, e.g. for index search, stepper motor emulation, pulse train control via Index1, continuous speed in both directions by means of control signals on Index0 and Index1, "0 mode", as well as IxR-compensated speed control (firmware version A only).

	Command	Range x	Response
- Write ramp profile / operating mode:	nP0=x<LF>	0...7	n*<LF>
- Read ramp profile / operating mode:	nP0?<LF>		n:x<LF>

Example: 4P0=1<LF> for positioning with linear ramp or 4P0=4<LF> for continuous speed- speed control without target position, respectively, e.g. for index search (all examples for device with address 4)

To read the parameter the equal character "=" in the command string is replaced by a question mark "?".

x=0: 0-Modus (position is held, even if speed and new position are already defined).
x=1: Positioning with linear acceleration and deceleration ramp
x=2: Positioning without ramp
x=3: Positioning with squared ramp
x=4: Continuous move with linear acceleration and deceleration ramp for index search
x=5: Stepper motor mode, step signal to Index0, direction signal to Index1
x=6: (Firmware version A only: Tacho generator regulated speed; not available for µMC500, since there are no analog inputs, nor reference voltage outputs)
x=7: Firmware version A only: IxR-compensated speed control

The remaining operating modes, i.e. pulse train control via Index1 and continuous speed in both directions by means of control signals on Index0 and Index1 are not initiated via the commands for the ramp profile (nP0=xxx<LF>). These configurations are described in Chapter 5 "Signal-controlled operation".

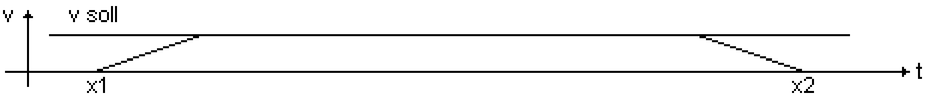


Figure 1: Positioning with linear ramp (mode 1): Rated speed reached

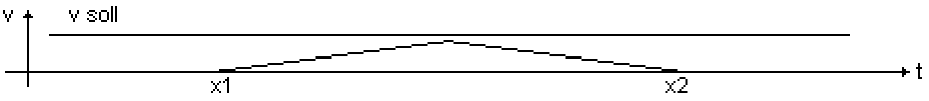


Figure 2: Positioning with linear ramp (mode 1): Rated speed not reached

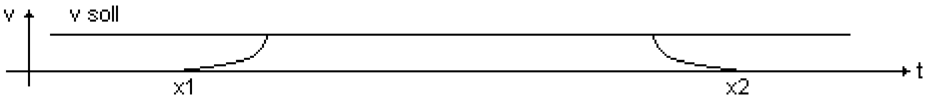


Figure 3: Positioning with squared ramp (mode 3)

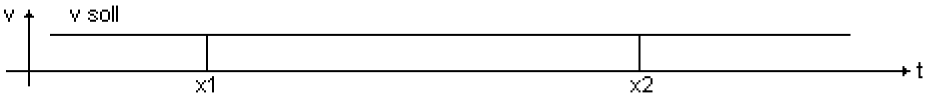


Figure 4: Positioning without ramp (mode 2)

A current positioning run can be interrupted at any time with a predefined deceleration ramp. For current positioning runs, the command for the interruption of the target run can be used in all 3 operating modes without restrictions. Pay attention to the fact that, in case of a target run interruption, the drive always decelerates with a linear ramp, independent from the ramp setting (linear, squared or abrupt [no ramp]) of the current positioning run. After an interruption, the positioning run can be continued by setting the new target position.

During a current positioning run, one new target position can be defined; the drive then decelerates with linear ramp to speed 0, i.e. with zero speed pass, if the new target position is in the opposite direction and then accelerates with the defined ramp setting towards the new target direction. Only a maximum of one new target position can be defined during a current positioning run.
Further target position definitions are only accepted after a zero speed pass, initiated by a previous new definition of the target position (one nesting level only).

4.1 Continuous speed control with linear ramp

The operating mode 4 offers to run any speed profile with linear acceleration and deceleration ramp. At changes of the rated speed the drive accelerates or decelerates with linear ramp to the new defined value. Negative speed values must be sent with a prefixed minus sign (-), positive speed values can be sent without a sign.

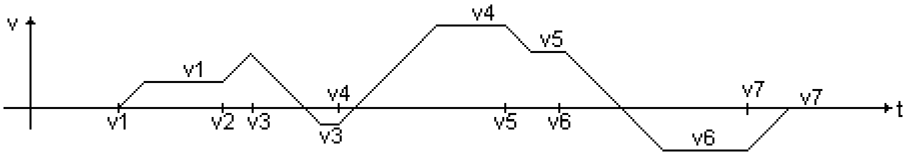


Figure 5: Speed profile in mode 4 with changing rated values

5. Signal-controlled operation

The servo controller µMC500 can also be configured to evaluate static or dynamic signals on the inputs Index0 and Index1. With this, target runs to a maximum of 8 positions, stepper motor emulation or continuous speed control without receiving further commands via serial interface is possible in stand-alone operation. The required parameters can, like all other data, be permanently stored in the parameter memory. In this way the data is, after a new power up, available again (see Chapter 19 “Saving and loading the parameters, default values”)

5.1 Stepper motor emulation via Index0 und 1

This operating mode evaluates the input Index0 for the step pulses and Index1 for a direction signal. The input Index0 reacts to the falling edge of the step signal and triggers a position change by one increment of the encoder each time. The maximum permissible frequency on the input Index0 amounts to 10kHz. The definition of rated speeds is of no importance in this operating mode.

Note: In stand-alone stepper motor mode without connection to a host computer via the serial interface, possible error states (e.g. loss of position due to active current limitation, etc.) are only displayed by means of the status LEDs, since with the µMC500V10 there are no further I/O signals available for error indication. Digital status signals for error notification are only available from hardware version µMC500V20 respectively V12 or higher.

The operating mode 5 for stepper motor emulation is written as follows:

	Command	Range x	Response
- Ramp profile / write operating mode:	nP0=5<LF>	-	n*<LF>

5.2 Continuous speed control via Index0 und Index1

This configuration enables the stand-alone operation of the µMC500 using static control signals on the inputs Index0 and Index1. In principle, the drive controller is operated internally in operating mode 4 in this configuration, i.e. with continuous speed control and linear acceleration and deceleration ramps. This implies that the speed and acceleration-relevant parameters, which are valid for mode 4, are used. The total of 4 possible logical states of the inputs Index0 und Index1 control the drive as follows:

Index0	Index1	Action
0	0	Drive deactivated, i.e. freely movable (stop); all internal error states, which occurred in the active control states are deleted resp. cleared.
0	1	Movement / rotation to the left (CCW), depending on the polarity of the drive / incremental encoder signals as well as sign of the rated speed value.
1	0	Movement / rotation to the right (CW), depending on the polarity of the drive / incremental encoder signals as well as sign of the rated speed value.
1	1	Drive is locked with stationary positioning control (hold)

To make this function work as stand-alone operation, the parameters rated value and ramp gradient, a current regulation mode including peak current values adapted to the application as well as the activation of this function must be written down and saved in the parameter memory. The necessary parameters for this function are listed in the following table.

	Command	Range y	Response	see Chapter
- Write rated speed:	nG0=yyyyy<LF>	-16384...16383	n*<LF>	6
- Write ramp gradient:	nJ0=yyy<LF>	0...255	n*<LF>	12
- Write range setting:	nJ1=y<LF>	0, 1	n*<LF>	12
- Write current regulation mode:	nB3=y<LF>	0...3	n*<LF>	17
- Write constant current I_{oc}:	nB0=yyy<LF>	0...1023 [mA]	n*<LF>	17

Continuation:	Command	Range y	Response	see Chapter
- Activate continuous speed via Index0 / 1:	nxS=y<LF>	1, 0	n*<LF>	5
- Save the parameter in non-volatile memory:	nE1=1<LF>	-	n*<LF>	19

The reference of the individual parameters to the corresponding chapter (see Chapter), in which the parameters are described more detailed, is only given once at the beginning of this User Manual to make it easier for the user to enter the matter around IMG drive controllers. In the subsequent chapters only the relevant functions or parameters are described without referring to further context, such as saving the parameters, etc.

Configuration example: 4G0=123<LF> (rated value 1230 incr./s); 4J0=5<LF> (ramp gradient value 5); 4J1=1<LF> (range for ramp gradient high); 4B3=1<LF> (current limitation snapping in position with restart); 4B0=100<LF> (peak current 100mA); 4xS=1<LF> (activation of the control for cont. movement via Index0/1); 4E1=1<LF> (save all parameters for stand-alone operation in non-volatile memory).

To read the parameters, the equal sign “=” in the command string is replaced by a question mark “?”

After this the µMC500 is ready to perform this function in stand-alone operation without a connection to the serial interface. It responds to the signals on Index0 and Index1 as described in the beginning of this section. The above example is valid for devices with address 4. The PC software STM7000 offers easy parameter settings for this function without the necessity of knowing the command structure: In the folder “Configure I/O control” these settings can be adapted to the application and can be saved.

Note: In stand-alone speed control mode without connection to a host computer via the serial interface, possible error states (e.g. loss of position due to active current limitation, etc.) are only displayed by means of the status LEDs, since with the µMC500V10 there are no further I/O signals available for error indication. Digital status signals for error notification are only available from hardware version µMC500V20 respectively V12 or higher.

Pulse-train-controlled target runs via Index0 und 1

The µMC500 can be configured for pulse-train-controlled target runs with a maximum of 8 freely definable positions in stand-alone mode. For this, the falling edges of the pulse train on input Index1 are counted and after the (freely definable) pulse filter time has elapsed, the drive performs the target runs. When 9 pulses (falling edges) have been received, an index or zero position search is triggered, which reacts on a falling edge on input Index0. The drive stops abruptly, the internal positioning values are set to zero (for index search refer to Chapter 8).

To make this function work as stand-alone operation the parameters rated value and ramp gradient, a current regulation mode including peak current values adapted to the application, the position values for 1 to 8 pulses, the pulse filter time (pulse acknowledge time) as well as the activation of this function must be written down and saved in the parameter memory. The necessary parameters for this function are listed in the following table.

	Command	Range y	Response
- Write position 0 (for 1 pulse):	nx0=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Write position 1 (for 2 pulses):	nx1=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Write position 2 (for 3 pulses): and so on ...	nx2=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Write position 7 (for 8 pulses):	nx7=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Write pulse filter (acknowledge) time (10ms units):	nxT=yyyy<LF>	0...1000 [10ms]	n*<LF>
- Write pulse-train-controlled target run (activate/deactivate):	nxP=y<LF>	1, 0	n*<LF>

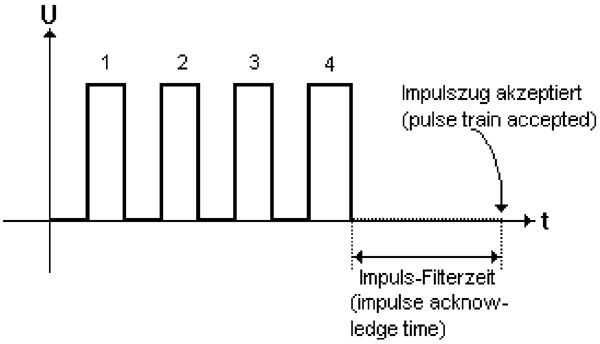


Figure 6: Pulse train – control signal on input Index1
for stand-alone operation

Note: In stand-alone pulse-train-controlled operating mode without connection to a host computer via the serial interface, possible error states (e.g. loss of position due to active current limitation, etc.) are only displayed by means of the status LEDs, since with the µMC500V10 there are no further I/O signals available for error indication. Digital status signals for error notification are only available from hardware version µMC500V20 respectively V12 or higher.

6. Rated speed

The rated speed represents the rotational or linear speed value that will be kept during the target run after completed acceleration phase until the deceleration phase begins. In mode 4, i.e. with continuous speed control without target position this is the end speed, which will be reached after completed acceleration or deceleration ramp.

- For the CPM motion control engine this value is defined as increments*10 per second [10* I/s] with a prefixed plus or minus sign.
- In mode for continuous speed without target position (mode 4) this is the end speed, which is reached after completed acceleration or deceleration.
- The minimum speed amounts to 10 increments/s, the maximum is 163840 increments/s. A predefined rated speed of e.g. 379 generates an incremental encoder frequency of exactly 3790Hz in a closed control loop. The static deviation is specified with a maximum of 2% (dependent of the tolerance of the controller internal precision RC oszillator, running at 24MHz)
- The direction of movement in mode 4 can be defined by means of the separate command nG2=x<LF> instead of speed values with prefixed plus or minus sign.
- In case of target runs in the modes 1, 2 und 3 the prefixed sign of the rated speed or the explicit definition by means of the command nG2=x<LF> has no influence on the direction of movement; this is exclusively determined by the shortest distance to the target position.

Note: Since the rated speed in the CPM motion control engine is processed in increments per second, the speed is inversely proportional to the resolution of the incremental encoder:
-> The smaller the resolution of the incremental encoder, the higher the (rotational) speed at a given rated value
-> The higher the resolution of the incremental encoder, the lower the (rotational) speed at a given rated value

Examples for a given rated speed value of 100, i.e. 1000 increments per second
Command nG0=100<LF>:
- With a small servo drive equipped with an incremental encoder having only 10 increments per revolution this rated value generates a speed of 100 rev./s
- With a servo drive equipped with an incremental encoder having 2000 increments per revolution this rated value generates a speed of 0.5 rev./s, i.e. half a revolution per second.

	Command	Range x,y	Response
- Write rated speed:	nG0=xxxxx<LF>	-16384...16383	n*<LF>
- Read rated speed: :	nG0?<LF>	-16384...16383	n:yyyyy<LF>
- Write direction of movement:	nG2=x<LF>	0, 1	n*<LF>

All rated speed values in [increments/s *10]

Example: 4G0=123<LF> generates an end speed of 1230 increments per second
(Example for device with address 4)

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

7. Target positions

7.1 Absolute target position

This value defines the target position in increments. The drive will run immediately after receipt of this command, with predefined ramp and speed, to this position.
During current positioning runs, a new absolute target position can be defined. Then the drive decelerates with linear ramp to speed 0 (with zero speed pass, if the new target position is in the opposite direction) and finally accelerates again with the predefined ramp mode towards the new target direction. Only a maximum of one new target position can be defined during a current positioning run. A new target position can only be defined after zero speed pass (triggered by a previous new definition of the target position): Only one nesting level can be processed. Repetitive sending of identical target position values during a current target run is possible at any time, the drive will not be lead through zero speed pass.

	Command	Range x,y	Response
- Write absolute target position:	nX0=xxxxxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Read absolute target position:	nX0?<LF>	-2147483648...2147483647	n:yyyyyyyy<LF>
- Read setpoint position:	nX3?<LF>	-2147483648...2147483647	n:yyyyyyyy<LF>
- Read actual position:	nX2?<LF>	-2147483648...2147483647	n:yyyyyyyy<LF>

Example: 4X0=67219<LF> sets the target position to 67219 increments
(Example for device with address 4)

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

Explanation of the terms target position, setpoint position, actual position:
- Target position: The position value to which the drive is supposed to run.
- Setpoint position: The momentary setpoint position for the positioning controller, which changes constantly during a current target run.
- Actual position: The momentary actual value of the position resp. incremental sensor, which shows the real, physical momentary position of the drive.

7.2 Relative target position

Setting this value allows the drive to run to a target position that is relative to the momentary position.
The advantage of this is that the host computer does not need to calculate this target position by itself, e.g. if the drive needs to be positioned from a virtual zero point.
The drive will run immediately after receipt of this command, with predefined ramp and speed, to this position.

	Command	Range x,y	Response
- Write relative target position:	nX1=xxxxxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Read relative target position:	Not possible		
- Read setpoint position:	nX3?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>
- Read actual position:	nX2?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>

Example: 4X1=5421<LF> sets the new target position to momentary position + 5421 increments
(Example for device with address 4)

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

Explanation of the terms target position, setpoint position, actual position:
- Target position: The position value to which the drive is supposed to run.
- Setpoint position: The momentary setpoint position for the positioning controller, which changes constantly during a current target run.
- Actual position: The momentary actual value of the incremental sensor, which shows the real, physical momentary position of the drive.

7.3 Set position without target run

Writing this command allows to assign a new numeric value to the drive in its momentary position without causing the drive to run to this position.

	Command	Range x,y	Response
- Set position without target run:	nX5=xxxxxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Read setpoint position:	nX3?<LF>	-2147483648...2147483647	n:yyyyyyyy*<LF>
- Read actual position:	nX2?<LF>	-2147483648...2147483647	n:yyyyyyyy*<LF>

Example: 4X5=7890<LF> sets the position without target run to 7890 increments
(Example for device with address 4)

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

Explanation of the terms target position, setpoint position, actual position:
- Target position: The position value to which the drive is supposed to run.
- Setpoint position: The momentary setpoint position for the positioning controller, which changes constantly during a current target run.
- Actual position: The momentary actual value of the incremental sensor, which shows the real, physical momentary position of the drive.

8. Index and zero point detection

In most positioning systems a reference, index or zero point serves as reference position for all subsequent target positions the drive will run to. This point is searched during the initialization phase of the system and defines the internal numeric zero point.

The following index search procedures were integrated in the µMC500:
- Index detection without pre-index on input Index0 optional with rising or falling signal edge
- Index detection with pre-index on input Index0 optional with rising or falling signal edge
- Index detection with pre-index on input Index1 and zero point on Index0, both optional with falling or rising signal edge
- Motor current index: Index search on the basis of mechanical end stops

The zero point search can be substantially accelerated with 2 different index search speeds in different directions.

8.1 Zero point detection on input Index0 without pre-index switch

Causes an abrupt stop of the drive on detection of the first falling or rising edge on input Index0. The numeric position, the ramp or operating mode, accordingly, as well as the rated speed are set to 0 after index detection.
To make use of this function, additionally mode 4 (nP0=4<LF> for continuous speed control) and a suitable speed and direction of movement (nG0=xxx<LF> respectively nG0=-xxx<LF>) need to be defined.

	Command	Range x	Response
- Zero point detection without pre-index via input signal on Index0:	nY0=x<LF>	0 or 1	n*<LF>
- Write operating mode for index search:	nP0=x<LF>	4	n*<LF>
- Write rated speed:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

The command parameter x (1 or 0, e.g. 4Y0=1<LF>) is used to define the signal edge that shall cause an index recognition:
x=1: Input Index0 reacts to the positive, rising signal edge.
x=0: Input Index0 reacts to the negative, falling signal edge.
If no parameter is set (e.g. 4Y0=<LF>) Index0 triggers on negative edges.

8.2 Zero point detection on input Index0 with pre-index switch on input Index0

This zero point detection causes the drive to decelerate with linear ramp from rated speed to index search speed (possibly in opposite direction) on detection of the first falling or rising signal edge on input Index0.
On occurrence of the second zero point signal, edge the motor stops abruptly, all positioning-relevant parameters and the rated speed are set to 0. The following configurations with pre-index - or zero-point - switches, accordingly, can be evaluated with this method:

- 2 parallel-wired switches for pre-index and zero point are installed in such a way that, when run over and run against, 2 separate signal edges are produced.
- 1 index switch and definition of a negative index search speed: By means of a “one and only” zero point switch, the pre-index and the zero point signal edges are produced with only one switch.

This applies to both configurations: The drive runs with high speed against the pre-index or zero point switch, accordingly, and runs over it. The pre-index or zero point switch produces the first signal edge and causes the drive to decelerate to index search speed or speed 0 and then to accelerate to index search speed in opposite direction in case of a negative index search speed.
Then the drive runs (onwards or back) towards the zero point switch with a much smaller index search speed until the switch produces the second signal edge upon actuation. The drive stops abruptly.
After triggering a zero point recognition - irrespective of the presence of an index switch – the numeric position, the rated speed as well as the ramp or operating mode, accordingly, are set to 0.
To make use of this function mode 4 (nP0=4<LF> for continuous speed control) and a suitable speed and direction of movement in the direction of the zero point switch (nG0=xxx<LF> respectively nG0=-xxx<LF>) as well as a corresponding, usually much smaller index search speed (nG1=xxx<LF> resp. nG1=-xxx<LF>) need to be defined.

	Command	Range x,y	Response
- Index search with pre-index via input signal on Index0	nY1=xy<LF>	0 or 1	n*<LF>
- Write operating mode for index search:	nP0=x<LF>	4	n*<LF>
- Write index search speed:	nG1=xxxxx<LF>	-16384...16383	n*<LF>
- Write rated speed:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

The command parameters x and y (1 or 0, e.g. 4Y1=01<LF>) are used to define the signal edge that shall cause a pre-index or an index recognition, accordingly:
x=1: Input Index0 reacts to the positive, rising edge for the first pre-index signal
x=0: Input Index0 reacts to the negative, falling edge for the first pre-index signal
y=1: Input Index0 reacts to the positive, rising edge for the second index signal
y=0: Input Index0 reacts to the negative, falling edge for the second index signal

If no parameter is set (e.g. 4Y1=<LF>) Index0 triggers in both cases on negative edges.

8.3 Zero point detection with pre-index on input Index1 and zero point on input Index0

This mode offers a combined use of an incremental encoder index (one pulse per revolution) in connection with a pre-index switch. To achieve this, the incremental encoder index signal (comes up every revolution) is connected to input Index0 and the pre-index switch is connected to input Index1 of the µMC500.
During an index search the index signals of the incremental encoder on input Index0 are ignored until the pre-index switch connected to input Index1 produces a signal edge. Then the drive switches to index search speed and the input Index0 is activated to receive a signal edge from the incremental sensor. From now a signal edge from the incremental encoder index causes the drive to stop abruptly.
After triggering a zero point recognition - irrespective of the presence of an index switch – the numeric position, the rated speed as well as the ramp or operating mode, accordingly, are set to 0.
To make use of this function mode 4 (nP0=4<LF> for continuous speed control) and a suitable speed and direction of movement in the direction of the zero point switch (nG0=xxx<LF> respectively nG0=-xxx<LF>) as well as a corresponding, usually much smaller index search speed (nG1=xxx<LF> resp. nG1=-xxx<LF>) need to be defined.

	Command	Range x,y	Response
- Write input Index0 for zero point detection and Index1 for pre-index detection:	nY2=xy<LF>	0 or 1	n*<LF>
- Write operating mode for index search:	nP0=x<LF>	4	n*<LF>
- Write index search speed:	nG1=xxxxx<LF>	-16384...16383	n*<LF>
- Write rated speed:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

The command parameters x and y (1 or 0, e.g. 4Y2=01<LF>) are used to define the signal edge that shall cause a pre-index or an index recognition, accordingly:
x=1: Index1 reacts to positive, rising signal edge
x=0: Index1 reacts to negative, falling signal edge
y=1: Index0 reacts to positive, rising signal edge
y=0: Index0 reacts to negative, falling signal edge

If no parameter is set (e.g. 4Y2=<LF>) Index0 and Index1 trigger on negative edges.

8.4 Index detection by means of motor current

This method enables zero point detection without using any index switch, if for design reasons, such a switch cannot be integrated. In this case, the zero position is defined by a mechanical end stop which causes the momentary actual value for the motor current to rise above a defined threshold when the drive runs against the stop. After activation of this operating mode all positioning-relevant parameters are reset as soon as the actual current value rises above the rated value. The drive stops abruptly, the numerical position, the rated speed and the ramp or operating mode, accordingly, are set to 0:

To make use of this function mode 4 (nP0=4<LF> for continuous speed control) and a suitable speed and direction of movement in the direction of the mechanical stop (nG0=xxx<LF> respectively nG0=-xxx<LF>) as well as a corresponding rated and maximum current for the index detection (nB0=xxx<LF> in [mA]) need to be defined.

	Command	Range x	Response
- Write index detection via drive current:	nY3=1<LF>	-	n*<LF>
- Write maximum rated detection current:	nB0=xxx<LF>	0...800 [mA]	n*<LF>
- Write operating mode for index search:	nP0=x<LF>	4	n*<LF>
- Write rated speed:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

9. Capturing the momentary position

This function allows to capturing the momentary actual position at the time there is a falling or rising edge on input Index1. As soon as this position value has been captured, this function will automatically be deactivated. Initialization of the input Index1 and reading of the captured momentary position, which was saved at the time of the signal edge, is done by means of the following commands:

	Command	Range x,y	Response
- Activate input Index1:	nY5=x<LF>	0 or 1	n*<LF>
- Read captured position:	nX4?<LF>	-2147483648...2147483647	n:yyyyyyyyy<LF>

Example: 4Y5=1<LF> initializes input Index1 for querying the momentary position on rising edge, 4Y5=0<LF> on falling edge. (Example for device with address 4)

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

10. Setting signal-triggered target position

With the definition of this position value, a target run can be triggered by a signal edge on input Index1.
To achieve this, the target position the drive is supposed to run to and the activation of this function must be written. Furthermore, the desired ramp or operating mode for target runs (e.g. nP0=1<LF> for linear ramp) and a suitable speed (nG0=xxx<LF>) must be defined.

	Command	Range x, y	Response
- Write signal-triggered target position:	nX6=xxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Activate input Index1 for target run:	nY4=x<LF>	1 or 0	n*<LF>
- Write operating mode for index search:	nP0=x<LF>	1..3	n*<LF>
- Write rated speed:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

The parameter x (1 or 0, e.g. 4Y4=1<LF>) is used to define the signal edge that shall trigger the target run:
x=1: Input Index1 reacts to the positive, rising signal edge.
x=0: Input Index1 reacts to the negative, falling signal edge.
If no parameter is set (e.g. 4Y4=<LF>) the input triggers on negative edges.

11. Interruption of a current target run

The target run interruption command can be used in all 3 ramp and operating modes. Please note that a target run interruption will decelerate the drive with linear ramp. After an interruption, the positioning process can be resumed by setting a new target position.

	Command	Range x,y	Response
- Interrupt positioning process:	nG3=1<LF>	-	n*<LF>

12. Gradient of the acceleration and deceleration ramp

In the µMC500 all modes have a ramp function except ramp mode 2 (positioning without ramp). The gradient of the ramp can be adjusted in 2 ranges of 255 steps each.

Please note that the gradient and range setting of the ramp in the modes with linear ramp (positioning with linear ramp and continuous speed control) can only be changed while the drive is not moving. For example, if you try to change the ramp during a positioning run in mode 1, a response containing an exclamation mark „!“ will appear. In case of positioning runs with squared (track-dependent) ramp the adjustment of the gradient and the range setting can be made at any time. In mode 2 these settings have no effect.

	Command	Range x,y	Response
- Write ramp gradient:	nJ0=xxx<LF>	0...255	n*<LF>
- Write range setting:	nJ1=x<LF>	0, 1	n*<LF>

Example: 4J0=10<LF> sets the ramp gradient to value 10; 4J1=1<LF> sets the range setting to 1 (example for device with address 4)

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

The time the drive takes to accelerate to end speed, or to decelerate to 0 is calculated as follows (this applies to positioning with linear ramp in mode 1, i.e. nP0=1<LF> and to continuous movement without target position in mode 4, i.e. . nP0=4<LF>):

t = V *range / Tsample*ramp

t:	Acceleration time [s] until rated speed value is reached
V:	Rated speed value (command nG0=xxxx<LF>)
ramp:	Ramp gradient (command nJ0=xxx<LF>)
range:	Range setting of the ramp (command nJ1=x<LF>) whereas: range=1 at nJ1=1<LF> respectively range=256 at nJ1=0<LF>
Tsample:	Sampling rate (2560Hz)

Example with nG0=2560<LF>, nJ0=1<LF>, nJ1=1<LF>:
Acceleration time t = 2560*1/2560*1 = **1s**

Example with nG0=1000<LF>, nJ0=5<LF>, nJ1=1<LF>:
Acceleration time t = 1000*1/2560*5 = **0.078s**

Example with nG0=5000<LF>, nJ0=2<LF>, nJ1=1<LF>:
Acceleration time t = 5000*1/2560*2 = **0.97s**

Example with nG0=5000<LF>, nJ0=5<LF>, nJ1=1<LF>:
Acceleration time t = 5000*1/2560*5 = **0.39s**

Example with nG0=5000<LF>, nJ0=1<LF>, nJ1=1<LF>:
Acceleration time t = 5000*1/2560*1 = **1.95s**

Example with nG0=400<LF>, nJ0=10<LF>, nJ1=0<LF>:
Acceleration time t = 400*256/2560*10 = **4s**

13. Read status 0, 1 and 3

13.1 Status 0

The **Status 0** gives information which actions the positioning and speed controller currently performs, if there is an acceleration or a deceleration ramp currently run, if the predefined target position is reached and some more information. The information is conveyed in form of an ASCII string containing 10 characters, whereas the first character is the responding module / device, the second character is a colon (:) and the remaining 8 bytes show the status information, e.g. 9:00011010<LF>

	Command	Range x,y	Response
- Read internal status:	nQ0?<LF>	00000000...11111111	n:yyyyyyyy<LF>
Order of the bytes:			1..........8

The individual bytes (y) have the following significance if 1:

Byte	Significance
1	Position 0 reached
2	Motor current limitation / regulation was or is actually active
3	Tracking error; setpoint and actual position deviate by x increments (adjustable)
4	Drive has reached rated speed
5	Drive stands still
6	Position initiated via input Index1 reached
7	Set value for direction of rotation
8	Defined target position reached

13.2 Status 1

The **Status 1** gives information whether the electronic overload fuse, the short-circuit or motor protection is active, or if the temperature of the power stage is too high, etc. The information is conveyed in form of an ASCII string containing 10 characters, whereas the first character is the responding module, the second character is a colon (:) and the remaining 8 bytes show the status information, e.g. 9:10010010<LF>.
The **Status 1** can – except for shutdown due to overtemperature or low voltage – be reset with the command nZ1<LF>.

	Command	Range x,y	Response
- Read status word 1:	nQ1?<LF>	00000000 to 11111111	n:yyyyyyyy<LF>
Order of the bytes:		1............8 1..........8	1...........8
- Delete status 1:	nZ1=1<LF>	-	n*<LF>

The individual bytes (y) have the following significance if 1:

Byte	Significance
1	Over - temperature power stage (>110”C)
2	Supply voltage <8V +-10% (supply voltage too low)
3	Electronic overload fuse active; IoM>0.8A @ t > 1s
4	Electronic short-circuit protection active; IoM>3A @ t > 10us
5	Actual position not equal to setpoint position due to active current limitation or motor protection
6	Overdrive monitoring; PWM current value >= permissible value (adjustable)
	Positioning controller / power stage deactivated due to overdrive / short-circuit / motor protection or deliberate deactivation
8	Over - voltage, supply voltage >32V (supply voltage too high)

Example response from address 4: 4:00101000<LF>

13.3 Status 3

The **Status 3** is a summary of all errors in Status 1 and amounts to 0, if there are no error conditions.
The information is conveyed in form of an ASCII string containing 3 characters, whereas the first character is the responding module, the second character is a colon (:) and the last byte shows the status information, e.g. 9:3<LF>.

	Command	Range x,y	Response
- Read status 3:	nQ3?<LF>	0...7	n:y<LF>

x=0: No error conditions
x>0: Error conditions detected, maximum value 7

14. Reset, deactivating, reactivating

With the command nZ0=1<LF> all positioning-relevant parameters, i.e. setpoint position, actual position, target position, ramp or operating modes, accordingly, rated speed, overdrive, motor protection and current limitation states as well as all relevant internal registers are reset to 0. If there are no permanent error conditions, all error-related status bits in status 0 and 1 are also reset with this command which results in the drive remaining in stationary positioning control in mode 0. After this, all required parameters, such as rated speed, ramp profile, etc. must be written anew or loaded from the non-volatile parameter set, accordingly.

The command nZ1=1<LF> causes a reset of all relevant drive protection flags, which are triggered by a control point that is too high (PWM value) or a tracking error that is too large (deviation setpoint position – actual position). These flags only become active if the values for the current limitation or regulation of the drive (in connection with the supply voltage), have been set too high, i.e. if the current regulator cannot act as limiter. See Chapter 17 “Current regulator, current limitation, overdrive monitoring, tracking error, motor protection“. The reset of the motor protection flags leaves all other parameters in their set values. After resetting these flags, the positioning controller is reactivated and thus allows to run to new target positions immediately after this, provided there aren’t any further or new error conditions.

The command nZ2=1<LF> deactivates the drive in such a way that it can be freely moved by external forces, whereas the actual position is continuously reported.
A subsequent reactivation initiated by means of the command nZ3=1<LF> loads the setpoint position and target positon together with the currently reported actual position value and then restarts positioning control of the drive.

	Command	Range x,y	Response
- Trigger complete reset:	nZ0=1<LF>	-	n*<LF>
- Reset drive protection flags:	nZ1=1<LF>	-	n*<LF>
- Deactivate positioning control:	nZ2=1<LF>	-	n*<LF>
- Reactivate positioning control:	nZ3=1<LF>	-	n*<LF>

15. Reading input values of Index0 and 1, supply voltage, actual current and temperature

The input values can be read with the following commands:

	Command	Range x	Response	
- Read Index0:	nI0?<LF>	0, 1	n:x<LF>	
- Read Index1:	nI1?<LF>	0, 1	n:xLF>	
- Read supply voltage:	nI5?<LF>	5,00..32,00	n:xx.xxV<LF>	[V]
- Read actual current of the drive:	nI6?<LF>	0.05..0.80	n:x.xxA<LF>	[A]
- Read temperature:	nI9?<LF>	0..100	n:xxC<LF>	[°C]
- Read analog voltage 0: (µMC500V20 and V12 only)	nIA?<LF>	0..1023	n:xxxx<LF>	

Remark regarding nI0<LF>: „I“ is he upper case „i“ and denotes „Input“

15.1 Write output value

The output value can be written with the following command:

	Command	Range x	Response
- Write output 0: (hardware version µMC500V20 and V12 only)	nO0=x<LF>	0, 1	n:x<LF>

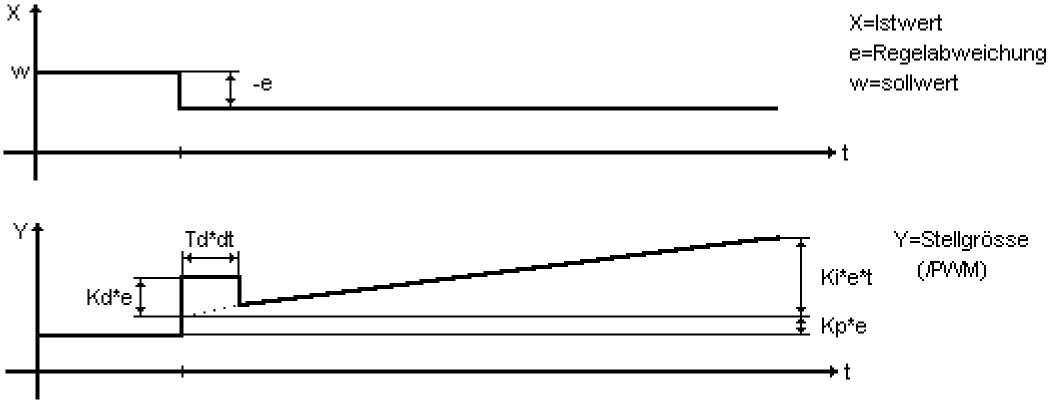
To read the status of the output the equal character “=” in the command string is replaced by a question mark “?”.

Attention: After writing of the output 0 has been carried out, this output is not available as status output for error messages anymore!

16. Transfer functions of the positioning controller

In order to ensure that the positioning / speed controller provides optimum results for each application with regard to transient response and control deviation, all 6 parameters in the µMC500 can be freely defined:
Proportional component Kpp, differential component Kdp, Integral component Kip, time constant of the differential component Tdp as well as the limitation of the integral component Rip can be adapted in wide ranges to optimize the behavior of the control loop. The sampling rate is permanently set 2.560kHz.
The CPM motion control engine integrated in the µMC500 produces its own space vector commutation by means of a continuous positioning control of the drive, which results in the fact that this system doesn't need a separate speed controller. Likewise, for positioning processes no speed estimator for acceleration / deceleration ramps is necessary, the required momentary speed for target runs is directly derived from the momentary position.
This substantially facilitates the parameter setting for the controller and gives the system intrinsic stability (patented principle).

The figure demonstrates the control behavior of a PID controller by means of a jump function of the actual value:



The following commands allow access to the control parameters of the CPM motion control engine:

	Command	Range x	Response	Default
- Write proportional component Kpp [1]:	nA0=xxx<LF>	0...255	n*<LF>	5
- Write integral component [1]:	nA3=xxx<LF>	0...255	n*<LF>	10
- Write differential component Kdp [1]:	nA1=xxx<LF>	0...255	n*<LF>	10
- Write time constant differential component Tdp [400us]:	nA2=xxx<LF>	1...255	n*<LF>	2
- Write Rip limitation [1]:	nA4=xxx<LF>	1...127	n*<LF>	3
- Sampling rate [Hz]:	-	-	-	2.560kHz

All 6 parameters can be changed at any time irrespective of the momentary operating mode.
The default values indicated in this table are factory settings, which are stored in the parameter memory and loaded during power up. These are selected in such a way that a wide spectrum of drive and incremental encoder combinations should principally be stable at first power up (uncritical „low-performance setting“).

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

17. Current regulator, current limitation, overdrive monitoring, tracking error, motor protection

In the µMC500 there are totally 4 current limitation and current regulation modes integrated with parameters that can be adjusted in a wide range. These ensure a comprehensive protection for drive, mechanics and power stage. In addition to this, there is a separate short-circuit and over temperature protection for the power stage, the global over temperature monitor, the overdrive protection and the tracking monitor.

The current regulation and protection mechanisms can be summarized as follows.

- Current protection mode 0: If the drive current exceeds the rated value (parameter constant current I_{oc}) the drive will be stopped, i.e. for example a current target run will be abruptly interrupted. The trigger time amounts to (actual current – rated current)x400µs/1A. This condition is stored in Status1 and is displayed on the status LEDs.

- Current protection mode 1: If the drive current exceeds the rated value (parameter constant current I_{oc}) the drive's current will be limited (snapping in position), similar to exceeding the momentary torque of a stepper motor at low speed. During a current positioning run the drive will try to repetitively restart with a frequency between 2...20Hz. Depending on the momentary speed and the flexibility of the mechanical system. The repetitive trigger time amounts to (actual current – rated current)x400µs/1A. This condition is stored in Status1 and is displayed on the status LEDs.

- Current protection mode 2: Constant current regulation wit freely definable regulation parameters for the proportional and the integral component as well as the constant current I_{oc}. Due to its integrating behavior this current regulator allows for a short time high peak currents during acceleration and deceleration ramps without acting as limiter. As long as the current regulation is active, this condition is stored in Status1 and is displayed on the status LEDs.

- Current protection mode 3: I^2*t – Current regulation with adjustable peak current I_{om}, constant current I_{oc} and time constant τI_{oq}. The time constant τI_{oq} can be adjusted in the range of 10ms...100s.
  If the constant current I_{oc} is exceeded the current in the drive will primarily be limited to the peak current I_{om}, which, as a rule, is a multiple of the constant current I_{oc}. Subsequently, the difference of the drive current to the constant current I_{oc} is squared and transferred to a numeric integrator for time calculation.
  The integrator calculates the time-depending reduction of the rated current down to constant current I_{oc}.
  After complete lapse of the I^2*t time constant τI_{oq} only the (lower) constant current I_{oc} flows in the drive.
  This is the drive current which is permissible to flow continuously under all temperature-relevant conditions, i.e. to keep permanent thermal stability of the servo drive.
  Due to its integrating behavior, this current regulator allows for a short time high peak currents during acceleration and deceleration ramps without acting as limiter.
  As long as the current regulation is active, this condition is stored in Status1 and is displayed on the status LEDs.

- Overdrive / control point monitor: The power stage control signal (PWM signal) and the corresponding filter time can be adapted to the requirements in a wide range. When the control point is exceeded and after the filter time has elapsed, the power stage will be deactivated.
  This condition is stored in Status1 and is displayed on the status LEDs.

- Tracking error monitor: Monitors the deviation between the setpoint position and the actual position in conjunction with a filter time. Both parameters can be adapted to the requirements in a wide range.
  When the tracking error is exceeded and after the filter time has elapsed, the power stage will be deactivated.
  This condition is stored in Status1 and is displayed on the status LEDs.

- Overvoltage and low voltage monitor: Supply voltages out of the permissible range deactivate the power stage.
  In case of over voltage events, the drive will be short-circuited in the power stage within < 10µs to avoid too high supply voltages, if the supply over voltage is generated by the servo drive itself.
  This condition is stored in Status1 and is displayed on the status LEDs.

- Global overtermperature monitor: Protects the complete module from operating temperatures >100°C; at temperatures above this limit the power stage will be deactivated.
  This condition is stored in Status1 and is displayed on the status LEDs.

- Short-circuit protection: Protects the drive's power stage against short-circuits against each other, against ground and against power supply. Furthermore, the power stage is equipped with an autonomous

overtemperature and overload protection.
This condition is stored in Status1 and is displayed on the status LEDs.

The commands that allow access to these parameters are listed in the following:

	Command	Range x	Response	Default
- Current protection mode:	nB3=x<LF>	0...3	n*<LF>	1
- Constant current I_{oc}:	nB0=xxx<LF>	0...1023 [mA]	n*<LF>	200mA
- Peak current I_{om} für I^2*t :	nB1=xxx<LF>	0...1023 [mA]	n*<LF>	800mA
- Time constant τI_{oq} für I^2*t :	nB2=xxx<LF>	0...10000 [10ms]	n*<LF>	10s
- Max. control point for overdrive monitor:	nZ4=xxx<LF>	0...255 [PWM]	n*<LF>	250
- Filter time for overdrive monitor:	nZ5=xxx<LF>	0...10000 [10ms]	n*<LF>	5s
- Max. tracking error:	nZ6=xxx<LF>	0...2147483647[I]	n*<LF>	63 incr.
- Filter time for tracking error:	nZ7=xxx<LF>	0...10000 [10ms]	n*<LF>	5s
- Current regulator proportional component Kpc:	nB4=xx<LF>	0...32	n*<LF>	1
- Current regulator integral component Kic	nB5=xx<LF>	0...32	n*<LF>	10

The default values indicated in this table are factory settings, which are stored in the parameter memory and loaded during power up.

To read the parameters the equal sign “=” in the command string is replaced by a question mark “?”

18. Activate 16bit - CRC (cyclic redundancy check)

The integration of CRC sums enable continuous verification or all transferred data on the serial line. This makes sense if the data must be securely transferred via slip rings, long cables (>>10m) and / or in environments with strong electromagnetic interference.

As factory default, the CRC evaluation is deactivated. However, it can be activated and stored as non-volatile data in the parameter set by means of the command described below. After activation it is mandatory to link all commands sent to the controller to this CRC sum to enable the controller to check the validity of the data and to receive acknowledgement. Corrupt or missing check sums or commands result in data not being accepted and in not receiving a response (inexistent communication).

Please bear in mind that there is no command to deactivate the CRC check. After being saved in the parameter set, this setting can only be undone by loading the default values – this, of course asks again for the correct CRC.

If the CRC is activated, all commands described in this document must have the following structure:

nks=ppppppp&yyyyyy<LF> for writing and nks?&yyyyyy<LF> for reading, accordingly.

The character ampersand „&“ marks the begin of the check sum. The numeric value after the character & (that means yyyyyy), is the CRC check sum; it must be an ASCII numeric value with a maximum of 5 digits plus prefixed sign (i.e. in a range of -32768 to 32767).

	Command	Range y	Response
- Activate CRC:	n&<LF>	-	n*&yyyyyy<LF>

Whereas n*&yyyyyy<LF> is the response of the activated CRC. Based on the fact that the CRC is calculated over the entire character string including “&”, the resulting CRC sum also depends on the device / module address.

Examples for commands without and with CRC:

Command without CRC	Command with CRC	Response
4P0=4<LF>	4P0=4&6903<LF>	4*&-19220<LF>
4G0=1267<LF>	4G0=1267&-22022<LF>	4*&-19220<LF>
4A0=20<LF>	4A0=20&17550<LF>	4*&-19220<LF>
4Q0?<LF>	4Q0?&4813<LF>	4:10001011&-19867<LF>

The CRC values in these examples are effectively applicable, i.e. true values for a module with address 4.

19. Saving and loading the parameters, default values

All parameters can be saved in the non-volatile memory range by means of the commands described below. The parameters are automatically reloaded after power up. This function is essential for the initiation of autonomous operating modes since, after completed configuration, there is no further communication via the serial interface.

All parameters listed in the following overview are saved:

- Parameters of the positioning controller
- Parameters of the current regulator
- Parameters of the overdrive, motor and mechanics protection
- Ramp / operating modes
- Parameters and modes for autonomous operation
- Rated speed and ramp values
- The current (actual) position
- The device / module address
- The baud rate of the serial interface, 19200Bd or 115200Bd

The saved parameters can also be explicitly loaded during operation, Values, possibly already written before via the serial interface, are then overwritten with those from the non-volatile parameter set.
In case of erroneous parameter settings, hard-coded default values (factory settings) can be loaded, which should principally provide for a stable operation and a sufficient drive protection for typical applications.
The values of the factory settings (default) are listed in the table that summarizes all commands, see table 1, page 30. The factory settings can, on request, be adapted to customer-specific requirements. Attention: The loading procedure of the hard-coded factory settings / default values always involves also a saving instance of these hard-coded factory settings in the non-volatile FLASH memory.

The following commands are used to save and load the factory settings and the user-specific parameters:

	Command	Response	
- Save parameter set:	nE1=1<LF>	n*<LF>	
- Load parameter set:	nE2=1<LF>	n*<LF>	
- Load (default) factor settings:	nE3=1<LF>	n*<LF>	
- Read serial number:	nE4?<LF>	n:S/N yy-nnnnn<LF>	e.g. S/N 10-02537

The serial number cannot be changed by the user (password-protected value).

Example: 4E1=1<LF> saves all current values in the parameter set 1 (the µMC500 possesses a single set only)

The live span of the non-volatile parameter FLASH memory allows for typically 250'000, minimum 20'000 erase / write cycles for 10 years data preservation. Though this live span seems to be more than sufficient for a typical application, the number of parameter saving instances should be kept to the necessary minimum. We recommend you to avoid cyclic saving instances with high repetition rates.

20. Writing and saving address of the controller

The individual address „n“ of each drive controller is determined via the serial interface, i.e. the µMC500 is not equipped with hardware switches, such as DIP switches or similar devices for address setting.

It is important to pay attention to the fact that the command for address assignment is a broadcast command, which is accepted by all drive controllers bearing any module address from 1 to 9. Consequently, this command must not be sent in an RS485 network array with several connected drive controllers, since all drive controllers would adopt the same module address. A individual communication would not be possible anymore.

For that reason, the address assignment needs to be performed on a single, isolated specimen connected to an RS485 bus, before this drive controller is integrated in an RS485 network array with several modules. Only after performing this procedure can the drive controller be addressed individually.

In addition to this, it is also important to save all parameters (nE1=1<LF>) after assigning the module address in order to make the individual address setting available after a new power up.

The factory setting for the module address of all µMC500 is 4, see examples of all commands in this document. The device / module address can be adjusted from 1 to 9, the easiest way to do this is using the PC software STM7000.

The command to determine the module / device address runs as follows:

	Command	Range x	Response
- Write module / device address:	nAdress=x<LF>	1...9	n*<LF>
- Save parameter set:	nE1=1<LF>	-	n*<LF>

Immediately after receipt and evaluation of this command the drive controller only responses to commands with the correct address anymore. Afterwards, do not forget to save all parameters with nE1=1<LF>.

Example: 4Adress=1<LF> changes the address of the connected module, which had any address before, to 1.
nE1=1<LF> saves the parameter set including module / device address.

21. Firmware update

The firmware of the µMC500 can be updated individually even within an RS485 network with several µMC500s, without the need of taking further measures (in-system / in-application - programming). The boot loader accepts addressed, unencrypted HEX strings as well as encrypted CHX strings.
The PC software STM7000 allows for an easy firmware update of the currently selected target module. To achieve this, the individual HEX or CHX lines of the firmware are extended by the target address, e.g. 9:04207B0012202F22DE<LF> for a HEX line to target address 9.
With a single-line initiated transfer, the receipt of each individual line will be acknowledged by the target module (9*<LF>), then the next line follows with acknowledgement, etc.
In case of a multiple line transfer of up to a maximum of 15 lines the receipt is only acknowledged after the 15th line. This procedure allows for a much quicker update process in systems with a UBS to RS485 adapter. However, the loading speed depends to a large extent on the corresponding configuration and the system-internal load of the computer. The PC software STM7000 decides, according to the version of the boot loader in the drive controller, whether a single or a multiple line transfer of HEX or CHX lines to the target module is possible. Whenever feasible, the multiple line transfer is preferred. The loading time for a multiple line transfer is typically about 5..15s.

IMG Antriebstechnik principally provides firmware updates as well as customer-specific variants as downloads with encrypted CHX files. The CHX files have a key length of 2048 bits and are only decrypted within the boot loader in the drive controller.
Example of a CHX line with target address 9: 9:ºÇï±¬↓←Ñ»Ñ³¬š‰ù¹ÂÞ›Î¡°‹´«Ë§æðó¯¦íó´Ú¬éÈ…öì<LF>
Information on the boot loader or the encryption program are principally not published.

However, it is the aim of IMG Antriebstechnik to establish, in the long run, an open-source community for its drive controllers, following the example of Linux. For this, when the time is ripe, the hardware will be published in the form of block diagrams, which enable the user to individually build their own firmware concepts with unencrypted downloads. More information and legal conditions follow.

22. Initial operation, examples

22.1 Continuous movement with 1290Hz incremental encoder frequency

4P0=4<LF>	Write continuous speed control without target position, mode 4.
4G0=129<LF>	Set rated speed to 129, i.e. 1290 increments per second.

The drive accelerates immediately after 4G0=129<LF> has been written to rated speed and runs at this speed until a new rated speed value is sent.

22.2 Run to target position 1456 increments with speed 150 and squared acceleration / deceleration ramp:

4P0=3<LF>	Write squared (track-dependent) ramp, mode 3.
4G0=150<LF>	Set rated speed to 150, i.e. 1500 incr./s.
4J0=3<LF>	Define a ramp gradient of 3.
4X0=1456<LF>	Write target position 1456 increments.

The positioning run starts immediately after the target position has been written.
Subsequent target runs are triggered by sending new target positions.

Table 1: Summary of the commands

	Command	Writing range ("=")	Reading range ("?")	Default
Positioning controller				
Proportional comp. pos. controller [1]:	A0	0..255	0..255	20
Differential comp. pos. controller [1]:	A1	0..255	0..255	10
Time constant diff. comp.Td [1/400us]:	A2	1..255	1..255	2
Integral comp. pos. controller [1]:	A3	0...255	0..255	10
Prelimitation I-component Ri [1]:	A4	0...127	0..127	3
Current regulator				
Constant current I_{oc} [mA]	B0	0..1023	0..1023	200mA
Current limit I_{om} for I^2*t [mA]	B1	0..1023	0..1023	800mA
Time constant τI_{oq} for I^2*t [10ms]	B2	0..10000	0..10000	10s
Current limitation / regulation mode:	B3	0..3	0..3	1
Prop. comp. Kpc current regulator [1]:	B4	0..32	0..32	1
Integral comp. Kic current regulator [1]:	B5	0..32	0..32	10
Sampl. rate current regulator [1/400us]:	B6	0..255	0..255	1
Speed				
Rated speed [incr./s *10]:	G0	-16384..16383	-16384..16383	0
Index search speed [incr./s *10]:	G1	-16384..16383	-16384..16383	0
Direction of movement explicit	G2	0, 1	0, 1	0
Abortion of target run	G3	1	-	-
Ramp				
Ramp gradient [1]:	J0	1..255	1..255	1
Gradient range:	J1	0, 1	0, 1	1
Index search, target run, momentary position				
Index on Index0, abrupt stop:	Y0	0, 1	0, 1	0
Index on Index0 w. pre-index on Index0:	Y1	00, 01, 10, 11	0, 1	00
Index on Index0 w. pre-index on Index1:	Y2	00, 01, 10, 11	0, 1	00
Index by drive current detection:	Y3	1	0, 1	0
Index1- activate triggered target run:	Y4	0, 1	0, 1	0
Index1 for query of momentary position:	Y5	0, 1	0, 1	0
Ramp / operating mode				
Operating mode:	P0	0..7	0..7	0
Status				
Read status 0:	Q0	-	00000000 to 11111111	
Read status 1:	Q1	-	00000000 to 11111111	
Read status 3:	Q3	-	0..7	
Target and momentary positions				
Target position absolute:	X0	-2147483648 to 2147483647	-2147483648 to 2147483647	0
Target position relative:	X1	-2147483648 to 2147483647	-	0
Actual value momentary position:	X2	-	-2147483648 to 2147483647	0
Setpoint value momentary position:	X3	-	-2147483648 to 2147483647	0
Queried momentary position:	X4	-	-2147483648 to 2147483647	0
Target position without target run:	X5	-2147483648 to 2147483647	-2147483648 to 2147483647	0
Index1-triggered target position:	X6	-2147483648 to 2147483647	-2147483648 to 2147483647	0

	Command	Writing range ("=")	Reading range ("?")	Default
Signal-controlled, autonomous operation				
Target pos. 0 for 1 pulse on Index1:	x0	-2147483648 to 2147483647	-2147483648 to 2147483647	0
Target pos. 1 for 2 pulses on Index1:	x1	-2147483648 to 2147483647	-2147483648 to 2147483647	100
Target pos. 2 for 3 pulses on Index1: etc. etc.	x2	-2147483648 to 2147483647	-2147483648 to 2147483647	200
Target pos. 7 for 8 pulses on Index1:	x7	-2147483648 to 2147483647	-2147483648 to 2147483647	700
Filter time for pulse train [10ms]:	xT	0..1000	0..1000	1s
Pulse train triggered target run:	xP	0, 1	0, 1	0
Continuous speed via Index0/1:	xS	0, 1	0, 1	0
Reset, drive protection, re/deactivate				
Complete reset:	Z0	1	-	-
Reset drive protection flags:	Z1	1	-	-
Deactivate positioning control:	Z2	1	0, 1	0
Reactivate positioning control:	Z3	1	0, 1	-
Control point for overdrive monitor:	Z4	0..255	0..255	250
Filter time of overdrive monitor:	Z5	0..10000	0..10000	5s
Tracking error:	Z6	0...2147483647	0...2147483647	63Inc.
Filter time tracking error:	Z7	0..10000	0..10000	5s
Activate CRC for command string:	Z8	1	-	off
Parameter set, default values, serial number				
Save parameter set:	E1	1	-	-
Load parameter set:	E2	1	-	-
Load (default) factory settings:	E3	1	-	-

Read serial number:	E4	-	S/N yy-nnnnn
			yy=year of production
			nnnnn=running serial number
				S/N yy-nnnnn

Read input values				
Read Index0:	I0	-	0, 1	-
Read Index1:	I1	-	0, 1	-
Read analog voltage (V20 only):	IA	-	0..1023	-
Read supply voltage:	I5	-	xx.xxV	-
Read actual current of drive:	I6	-	x.xxA	-
Read temperature:	I9	-	xxC	-

Remark regarding nI0<LF>:
„I“ is the upper case „i“ and denotes „Input“

Write output values				
Output 0 (hardware V20 and V12 only):	O0	0, 1	0, 1	-
Module type, version, date, manufacturer				
Read Info string:	^	-	µMC500 <Version> <Date> IMG Antriebstechnik Switzerland	