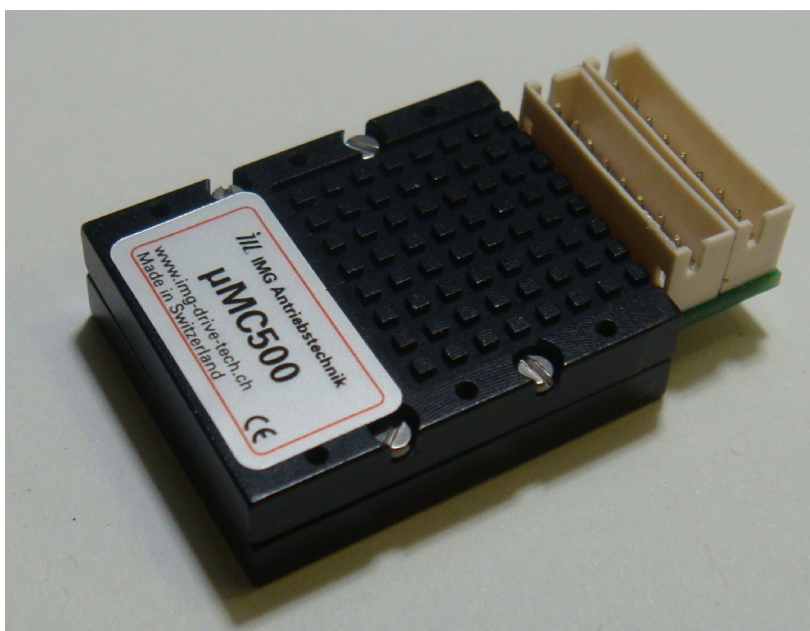


***il* IMG Antriebstechnik GmbH**

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

www.img-drive-tech.ch

μMC500 Firmware manual V6.00



	Seite
1. Vorwort	3
2. Kommandostruktur und Rückmeldungen auf RS485 und UART	4
3. PC-Software STM7000	5
4. Rampenprofile und Betriebsmodi	6
5. Signalgesteuerter Betrieb	8
6. Sollgeschwindigkeit	11
7. Zielpositionen	12
8. Index- und Nullpunktdetektion	14
9. Abtasten der Momentanposition	17
10. Signalgetriggerte Zielposition setzen	17
11. Unterbrechen einer laufenden Zielfahrt	17
12. Steilheit der Beschleunigungs- und Bremsrampe	18
13. Status 0, 1 und 3 lesen	19
14. Reset, Deaktivieren, Reaktivieren	21
15. Eingangswerte von Index0 und 1, Betriebsspannung, Ist-Strom und Temperatur lesen	22
16. Übertragungsfunktionen des Lagereglers	23
17. Stromregler, Strombegrenzung, Übersteuerungsüberwachung, Schleppfehler, Motorschutz	24
18. 16bit - CRC (cyclic redundancy check) aktivieren	26
19. Sichern und Laden der Parameter, Defaultwerte	27
20. Adresse des Reglers schreiben und sichern	28
21. Firmwareupdate	28
22. Erste Inbetriebnahme, Beispiele	29
Tabelle 1: Zusammenfassung Kommandi	30

1. Vorwort

Der µMC500 integriert in seinen kompakten Massen von nur 31 x 18 x 7.1mm einen vollwertigen, einachsigen Antriebsregler für kleine DC-Servoantriebe mit maximal 24W elektrischer Leistung.

3 verschiedene Beschleunigungs- und Bremsrampen, Positionierwege bis +-31bit, eine Geschwindigkeitsregelung mit einem Dynamikbereich von +-14bit, 4 verschiedene Stromregel- und Begrenzungsmodi sowie Einstellmöglichkeiten für alle beteiligten Parameter ermöglichen die Integration des µMC500 in fast jedem System mit einachsiger Antriebsregelung. Maximal 9 dieser Antriebsregler können über den RS485-Bus einzeln angesprochen werden; in vielen Anwendungsfällen können damit auch mehrachsige Systeme synchronisiert werden.

Die innovative CPM motion control engine erzeugt seine eigene space vector - Kommutation mittels kontinuierlicher Lageregelung des Antriebs; dies hat zur Folge, dass dieses System ohne separierten Geschwindigkeits-Regler auskommt. Bei Positioniervorgängen wird kein Geschwindigkeits - Estimator für die Beschleunigungs- / Bremsrampe benötigt; die erforderliche Momentangeschwindigkeit wird bei Zielfahrten direkt aus der Momentanposition abgeleitet. Dies vereinfacht die Parametrisierung des Reglers erheblich und verleiht dem System intrinsische Stabilität (Patentiertes Verfahren).

Aufgrund ihres Wirkungsprinzips beherrscht die im µMC500 integrierte CPM motion control engine auch lineare moving-coil - Antriebe.

Weitere im µMC500 integrierte Betriebsmodi:

- Schrittmotoremulation in Form von Impuls- und Drehrichtungssignalen an den Eingängen Index0 und Index1.
- Anfahren von 8 frei definierbaren Positionen mittels Impulspulsen an Eingang Index1; Impulstiming frei definierbar.
- Geregelt Geschwindigkeit/Drehzahl in beide Richtungen mittels statischer Signale an Index0 und Index1 mit frei definierbaren Rampen und Geschwindigkeiten.
- IxR kompensierte Drehzahlregelung ohne Inkrementalgeber (nur Firmware Version -A)

Firmwareupdate: die Firmware im µMC500 kann ohne weitere Massnahmen auch innerhalb eines RS485-Netzwerks mit mehreren µMC500 individuell aktualisiert werden (in system / in application); der Bootloader akzeptiert adressierte, unverschlüsselte HEX- sowie verschlüsselte CHX-Zeilen; siehe Kapitel 3, PC-Software STM7000.

Abweichende Firmwareversionen, kundenspezifische Bauformen oder angepasste Steckersysteme entwickeln wir gerne auf Anfrage. Der uMC500 kann in kundenspezifischen Versionen auch als reiner Stromregler für z.B. Magnetpulverbremser oder als intelligentes I/O-Modul mit Leistungsendstufe u.s.w. konfiguriert werden.

Eine kurze Übersicht der Eigenschaften:

- | | |
|--|---|
| - Maximaler Positionierweg: | 32 Bit (-2147483648...2147483647) |
| - Dynamik Geschwindigkeitsregelung: | 1:+-16384 |
| - Niedrigste Geschwindigkeit: | 10 Inkremente/s |
| - Höchste Geschwindigkeit: | 163840 Inkremente/s |
| - Geschwindigkeitstoleranz: | max. 2% statisch, 0% bei Laständerung |
| - Abtastrate Lageregler: | 2,560KHz |
| - Dynamik Beschleunigungs-Bremsrampen: | 1:255 in 2 Bereichen |
| - Rampenprofile: | insgesamt 3: linear, quadratisch, keine |
| - Index- resp. Nullpunktsuche: | 5 verschiedene Methoden |
| - Statusanzeige: | 2 LED |
| - Motorstromregelung/Begrenzung: | 50mA...800mA in Stufen von 0.683mA |
| - Mehrstufiger Endstufen- und Motorenschutz: | Kurzschlußschutz Motorausgänge gegeneinander sowie gegen Masse und Speisung, Schleppfehler- und Übersteuerungsüberwachung, 4 Stromregelmodi |
| - Serielle Schnittstelle: | RS485 voll- und halbduplex, UART (3,3 / 5V) 19200 oder 115200Bd, 8Data, 1Stop, No parity sowie I ² C |
| - Modul- resp. Deviceadressen am RS485-bus: | max. 9 (device adressen 1...9) |
| - Speisung Inkrementalgeber: | integriert, 5V / max. 50mA belastbar |

2. Kommandostruktur und Rückmeldungen auf RS485 und UART

Die Kommandi bestehen ausschliesslich aus schreibbaren (printable) ASCII-Zeichen, die auf jeder Tastatur zu finden sind. Alle Kommandi können auch mittels Terminal-Programm auf PC oder Hostrechner gesendet werden; die auf dem Terminalfenster ersichtlichen Rückmeldungen erscheinen in Form lesbaren Textes.

2.1 Kommandostruktur zum Schreiben der Parameter

nks=xxxxxxxx<LF>

<LF>=0x0a, wobei <CR>=0x0d ignoriert resp. verworfen wird.

Beispiel: 4G0=200<LF> schreibt den Sollgeschwindigkeitswert mit 200 ($200 \cdot 10 = 2000$ Inkremente/s)
4X1=109471<LF> schreibt eine Position relativ um +109471 Inkremente zur momentanen Position

n: Device- resp. Moduladresse, ASCII 1...9

k: Kommandogruppe, ASCII - Grossbuchstaben A...Z sowie a..z

s: Subindex der Kommandogruppe, ASCII - Dezimalzahl 0..9 sowie Grossbuchstaben A...Z

x: Parameterwert, ASCII - Dezimalzahlenwert mit bis zu 10 Stellen plus Vorzeichen

Auf jedes gültige Kommando mit gültiger Adresse erfolgt eine unmittelbare Rückmeldung, die wie folgt strukturiert ist:

na<LF>

Beispiel: 4<LF>*

n: Device- resp. Moduladresse (ASCII 1...9) des quittierenden Moduls

a: "*" oder "!" oder "#", wobei * = Kommando akzeptiert, # = fehlerhafter Parameter (ausserhalb des zulässigen Wertebereiches) und ! = Kommando kann zu diesem Zeitpunkt nicht angenommen werden, z.B. wiederholtes Neudefinieren der Rampe oder des Rampenbereiches mit mehr als einer Verschachtelungsebene. Ist kein Antriebsregler mit der angegebenen Adresse am Netzwerk angeschlossen oder ist das gesendete Kommando ungültig, erfolgt keine Rückmeldung.

2.2 Kommandostruktur zum Lesen der Parameter

nks?<LF>

<LF>=0x0a, wobei <CR>=0x0d ignoriert resp. verworfen wird.

*Beispiel: 4G0?<LF> liest den Sollgeschwindigkeitswert und sendet als Rückantwort z.B. 4:200<LF>.
4Q0?<LF> liest den Status 0 und sendet als Rückantwort z.B. 4:10010110<LF>*

Diese Beispiele sind repräsentativ für das Lesen fast jedes in diesem Manual beschriebenen Parameters, d.h. nahezu alle in diesem Manual beschriebenen Kommandi können mittels Fragezeichen "?" anstelle des Gleichheitszeichens "=" auch wieder ausgelesen werden.

Sequenzierung

Die einzelnen Kommandi werden nicht in einem Ringpuffer akkumuliert oder sequenziell abgearbeitet. Dies bedeutet, dass auf jedes gültige Kommando eine sofortige Rückmeldung (acknowledge) in oben beschriebener Form folgt. Erst nach dem Empfang dieser Rückmeldung seitens Hostrechner kann ein neues Kommando an jeweils einen Antriebsregler im RS485-Netzwerk abgesetzt werden.

Parameter der seriellen Schnittstelle:

- 19200 oder 115200Bd (via PC- Software STM7000 parametrisierbar)
- 8 datenbits
- 1 Startbit
- 1 Stopbit
- keine Parität
- kein XON/XOFF

3. PC-Software STM7000

Die für unsere Antriebsregler zur Verfügung stehende PC-Software STM7000 ermöglicht dem Anwender einen komfortablen Zugang zu allen Parametern, Statuswerten und I/O-Signalen. Ferner können frei definierbare Zielfahrten ausgeführt, ein Update der Firmware, Parametersätze gesichert oder mehrere an der RS485-Schnittstelle angeschlossene Antriebsregler identifiziert und einzeln angesprochen werden.

Alle Kommandi können auch mittels Terminal-Programm auf PC oder Hostrechner gesendet werden; die auf dem Terminalfenster ersichtlichen Rückmeldungen erscheinen ebenfalls in Form lesbaren Textes.

STM7000 ist unter folgenden System - Voraussetzungen lauffähig:

- Windows 2000, XP, Vista, und 7.
- 10MB freier Platz auf HD
- freie Schnittstelle COM1....COM8 mit RS485 voll- oder halbduplex - (4 - oder 2- Draht) Interface resp. USB-Anschluss

RS485-Interface

Die RS485-Schnittstelle kann in Form eines USB zu RS485 - Adapters oder einem im Hostrechner installierten RS485- Interface vorliegen. Da im µMC500 ein Vollduplex - Interface mit Halbduplex - Fähigkeit (Datenrichtungsumschaltung) zur Verfügung steht, kann auch ein einfacher RS232 zu RS485 - Konverter ohne Datenrichtungsumschaltung angeschlossen werden; hierin werden lediglich die ca. +-10V- TxD/RxD - Signale der RS232-Schnittstelle in differentielle 5V- A/B und Y/Z - Signale der RS485-Schnittstelle gewandelt.

USB zu RS485 - sowie RS232 zu RS485 - Adapter sind von vielen Herstellern erhältlich und bei fast allen Distributoren verfügbar.

Hersteller: ADLINK, Spectra, Exsys, Trigress Security, Moxa, Athen, Maxxtro, IPC-DAS u.s.w.

Zur Integration einer RS-485 - Schnittstelle in einem Hostrechner stehen eine grosse Auswahl an RS-485- transceiver verschiedener Hersteller zur Verfügung: Fairchild Semiconductor, TI, ON Semiconductor, Linear Technology, Analog Devices, Maxim u.s.w. um nur eine kleine Auswahl zu nennen.

4. Rampenprofile und Betriebsmodi

Mit insgesamt 9 verschiedenen Betriebsmodi im µMC500 kann zwischen 3 Rampenprofilen für Positionieraufgaben, kontinuierliche Geschwindigkeitsregelung z.B. zur Indexsuche, Schrittmotoremulation, Impulszugsteuerung via Index1, kontinuierlicher Geschwindigkeit in beide Richtungen mittels Steuersignalen an Index0 und Index1, "0-Modus" sowie IxR-kompensierter Drehzahlregelung (nur mit Firmware Version -A) gewählt werden.

	Kommando	Bereich x	Antwort
- Rampenprofil / Betriebsmodus schreiben:	nP0=x<LF>	0...7	n*<LF>
- Rampenprofil / Betriebsmodus lesen:	nP0?<LF>		n:x<LF>

Beispiel: 4P0=1<LF> für Positionieren mit linearer Rampe oder 4P0=4<LF> für kontinuierliche Drehzahl- resp. Geschwindigkeitsregelung ohne Zielposition, z.B zur Indexsuche (alle Beispiele für Device mit Adresse 4)

Zum Lesen der Parameter wird das Gleichheitszeichen "=" im Kommandostring durch ein Fragezeichen "?" ersetzt.

- x=0: 0-Modus (Position wird gehalten, auch wenn Drehzahl und neue Position schon vorgegeben sind).
- x=1: Positionieren mit linearer Beschleunigungs- und Bremsrampe
- x=2: Positionieren ohne Rampe
- x=3: Positionieren mit quadratischer Rampe.
- x=4: kontinuierliche Bewegung mit linearer Beschleunigungs- und Bremsrampe zur Indexsuche
- x=5: Schrittmotormodus, Schrittsignal an Index0, Richtungssignal an Index1
- x=6: (Nur in Firmware Version -A: Tachogenerator geregelte Drehzahl; nicht für µMC500 verfügbar, da keine Analogeingänge sowie Referenzspannungsausgänge vorhanden)
- x=7: Nur in Firmware Version -A: IxR - kompensierte Drehzahlregelung

Die weiteren Betriebsmodi, d.h. Impulszugsteuerung via Index1 sowie kontinuierliche Geschwindigkeit in beide Richtungen mittels Steuersignalen an Index0 und Index1 werden nicht über die Kommandi für das Rampenprofil (nP0=xxx<LF>) initiiert. Die Beschreibung dieser Konfigurationen wird im Kapitel 5, Signalgesteuerter Betrieb erläutert.

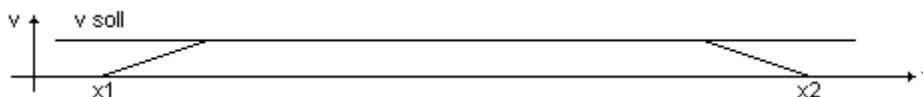


Bild 1: Positionieren mit linearer Rampe (mode 1): Sollgeschwindigkeit erreicht

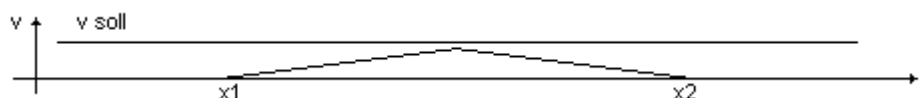


Bild 2: Positionieren mit linearer Rampe (mode 1): Sollgeschwindigkeit nicht erreicht

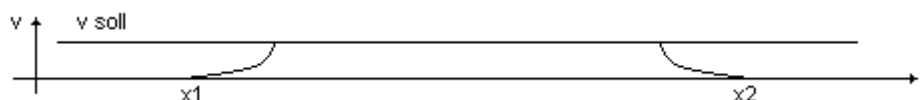


Bild 3: Positionieren mit quadratischer Rampe (mode 3)

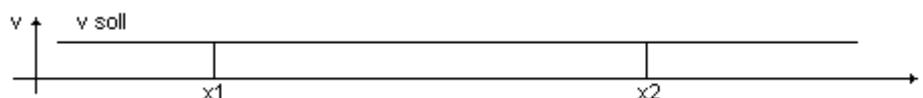


Bild 4: Positionieren ohne Rampe (mode 2)

Ein laufender Positioniervorgang kann jederzeit mit vorgegebener Bremsrampe unterbrochen werden. Das Kommando zur Zielfahrtunterbrechung kann in allen 3 Betriebsmodi für laufende Positioniervorgänge ohne Einschränkung angewendet werden. Zu beachten ist, das eine Zielfahrtunterbrechung den Antrieb immer mit linearer Rampe abbremst, unabhängig ob der momentan gesetzte Positioniervorgang eine lineare, quadratische oder abrupte (keine) Rampe besitzt. Nach einer Unterbrechung kann der Positioniervorgang durch Neusetzen der Zielposition fortgesetzt werden.

Bei einem laufenden Positioniervorgang kann 1 neue Zielposition vorgegeben werden: der Antrieb bremst dann mit linearer Rampe auf Drehzahl 0, d.h. mit Geschwindigkeitsnulldurchgang, wenn die neue Zielposition in entgegengesetzter Richtung liegt und beschleunigt dann wieder mit gesetzter Rampe in die neue Zielrichtung. Es kann maximal 1 neue Zielposition während eines laufenden Positioniervorganges vorgegeben werden. Weitere Zielpositionsvorgaben werden nur nach erfolgtem Geschwindigkeitsnulldurchgang, ausgelöst durch eine vorhergehende Zielneuvorgabe akzeptiert (eine Verschachtelungsebene maximal).

4.1 Kontinuierliche Geschwindigkeitsregelung mit linearer Rampe

Der Betriebsmodus 4 ermöglicht ein beliebiges Geschwindigkeitsprofil mit linearer Beschleunigungs- und Bremsrampe zu fahren. Bei Änderung der Sollgeschwindigkeit beschleunigt oder bremst der Antrieb mit linearer Rampe auf den neu vorgegebene Wert. Negative Geschwindigkeitssollwerte müssen vorzeichenbehaftet (-) gesendet werden, positive Werte können ohne Vorzeichen gesendet werden.

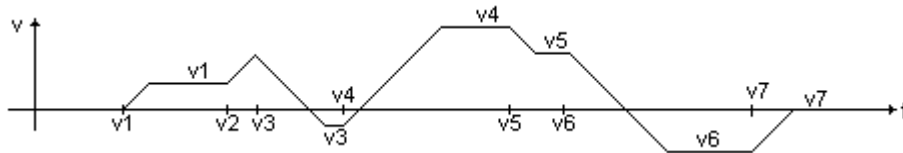


Bild 5: Geschwindigkeitsprofil im Modus 4 mit wechselnden Sollwerten

5. Signalgesteuerter Betrieb

Der Antriebsregler µMC500 kann auch zur Auswertung statischer oder dynamischer Signale an den Eingängen Index0 und Index1 konfiguriert werden. Somit sind Zielfahrten auf maximal 8 Positionen, Schrittmotoremulation oder kontinuierliche Geschwindigkeitsregelung ohne weiter Kommandierung via serieller Schnittstelle im autonomen Betrieb möglich; die erforderlichen Parameter können - wie alle anderen auch - im Parameterspeicher dauerhaft gesichert werden, so dass diese nach einem erneuten power up wieder zur Verfügung stehen (siehe Kapitel 19, Sichern und Laden der Parameter, Defaultwerte).

5.1 Schrittmotoremulation via Index0 und 1

Dieser Betriebsmodus wertet den Eingang Index0 für die Schrittpulse sowie Index1 als Richtungssignal aus. Der Eingang Index0 reagiert hierbei auf die fallende Flanke des Schrittsignals und löst am Antrieb jeweils eine Positionsänderung um ein Inkrement des Gebers aus. Die maximal zulässige Frequenz am Eingang Index0 beträgt 10KHz. Die Vorgabe von Sollgeschwindigkeitswerten ist in diesem Betriebsmodus ohne Bedeutung.

Zur Beachtung: im autonomen Schrittmotorbetrieb ohne Anbindung der seriellen Schnittstelle an einen Hostrechner sind allfällig auftretende Fehlerstati (z.B. Verlust der Position aufgrund aktiver Strombegrenzung u.s.w.) nur über die Status - LED's ersichtlich, da im µMC500V10 keine weiteren I/O - Signale zur Fehler-Rückmeldung verfügbar sind; digitale Statussignale zur Fehlermeldung sind nur ab Version µMC500V20 resp. V12 und höher verfügbar.

Der Betriebsmodus 5 zur Schrittmotoremulation wird wie folgt geschrieben:

	Kommando	Bereich x	Antwort
- Rampenprofil / Betriebsmodus schreiben:	nP0=5<LF>	-	n*<LF>

5.2 Kontinuierliche Geschwindigkeitsregelung via Index0 und 1

Diese Konfiguration ermöglicht den autonomen Betrieb des µMC500 in Form von statischen Steuersignalen an den Eingängen Index0 und 1. Prinzipiell wird der Antriebsregler in dieser Konfiguration intern mit Betriebsmodus 4 d.h. mit kontinuierlicher Geschwindigkeitsregelung und linearer Beschleunigungs- und Bremsrampe betrieben, so dass die geschwindigkeits- und beschleunigungsrelevanten Parameter, die für den Modus 4 gültig sind, verarbeitet werden. Die insgesamt 4 möglichen logischen Zustände der Eingänge Index0 und 1 regeln und steuern den Antrieb wie folgt:

Index0	Index1	Aktion
0	0	Antrieb deaktiviert, d.h. frei beweglich (stop); alle internen Fehlerstati, die in den aktiven Regelungszuständen eintraten, werden gelöscht.
0	1	Bewegung/Drehung nach links, abhängig von Polarität Antrieb/Inkrementalgebersignale sowie Vorzeichen des Sollgeschwindigkeitswertes
1	0	Bewegung/Drehung nach rechts, abhängig von Polarität Antrieb/Inkrementalgebersignale sowie Vorzeichen der Sollgeschwindigkeitswertes
1	1	Antrieb wird in stehender Lageregelung gehalten (hold)

Damit diese Funktion in Form eines autonomen Betriebs nutzbar wird, müssen die Parameter Sollgeschwindigkeit und Rampensteilheit, ein der Applikation angepasster Stromregelmodus mitsamt Grenzstromwerten sowie die Aktivierung dieser Funktion geschrieben und im Parameterspeicher gesichert werden. Die hierfür notwendigen Parameter sind folgend aufgeführt:

	Kommando	Bereich y	Antwort	siehe Kap.
- Sollgeschwindigkeit schreiben:	nG0=yyyyy<LF>	-16384...16383	n*<LF>	6
- Rampensteilheit schreiben:	nJ0=yyy<LF>	0...255	n*<LF>	12
- Bereichseinstellung schreiben:	nJ1=y<LF>	0, 1	n*<LF>	12
- Stromregelmodus schreiben:	nB3=y<LF>	0...3	n*<LF>	17
- Konstantstrom I _{oc} schreiben:	nB0=yyy<LF>	0...1023 [mA]	n*<LF>	17

Fortsetzung:	Kommando	Bereich y	Antwort	siehe Kap.
- Kont. Geschw. via Index0/1 aktivieren:	nxS=y<LF>	1, 0	n*<LF>	5
- Sichern der Parameter im nichtflüchtigen Speicher:	nE1=1<LF>	-	n*<LF>	19

Die Referenzierung der einzelnen Parameter auf entsprechende Kapitel (siehe Kap.), die diese genauer beschreiben, wird nur hier einmalig zu Beginn dieses User's Manuals aufgeführt, um einen leichteren Einstieg in die Materie rund um die Antriebsregler von IMG zu ermöglichen. In den nachfolgenden Kapiteln wird nur noch die betreffende Funktion resp. der Parameter ohne Bezug auf weitere Zusammenhänge wie z.B. Sichern der Parameter u.s.w. beschrieben .

Konfigurationsbeispiel: 4G0=123<LF> (Sollgeschwindigkeit 1230 inc/s); 4J0=5<LF> (Rampensteilheitswert 5); 4J1=1<LF> (Bereich für Rampensteilheit hoch); 4B3=1<LF> (Strombegrenzung rastend mit Wiederanlauf); 4B0=100<LF> (Grenzstrom 100mA); 4xS=1<LF> (Aktivierung der Steuerung für kont. Bewegung via Index0/1); 4E1=1<LF> (Sichert alle Parameter für einen autonomen Betrieb im nichtflüchtigen Speicher).

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

Danach ist das µMC500 für den autonomen Betrieb dieser Funktion ohne Anbindung an die serielle Schnittstelle bereit und reagiert wie eingangs beschrieben auf Signale an Index0 und Index1. Das obige Beispiel gilt für Devices mit Adresse 4. Die PC-Software STM7000 ermöglicht eine komfortable Parametrierung dieser Funktion ohne Kenntnisse der Kommandostruktur: im Folder "Configure I/O control" können diese auf die Erfordernisse der Applikation angepasst und gesichert werden.

Zur Beachtung: im autonomen Geschwindigkeitsregelbetrieb ohne Anbindung der seriellen Schnittstelle an einen Hostrechner sind allfällig auftretende Fehlerstati (z.B. Verlust der Position aufgrund aktiver Strombegrenzung u.s.w.) nur über die Status - LED's ersichtlich, da im µMC500V10 keine weiteren I/O - Signale zur Fehler-Rückmeldung verfügbar sind; digitale Statussignale zur Fehlermeldung sind nur ab Version µMC500V20 resp. V12 und höher verfügbar.

Impulszuggesteuerte Zielfahrt via Index0 und 1

Der µMC500 kann für impulszuggesteuerte Zielpositionsfahrten mit maximal 8 frei definierbaren Positionen im autonomen Betrieb konfiguriert werden. Hierbei werden die fallenden Flanken des Impulszugs am Eingang Index1 gezählt und nach Ablauf der (frei konfigurierbaren) Impulsfilterzeit als Zielpositionsfahrt ausgeführt. Bei Empfang von 9 Impulsen resp. fallenden Flanken wird eine Index- resp. Nullpositionssuchfahrt ausgelöst, die auf die fallende Flanke am Eingang Index0 reagiert: der Antrieb stoppt abrupt, die internen Positionswerte werden auf null gestellt (bezüglich Indexsuche siehe Kap. 8).

Damit diese Funktion in Form eines autonomen Betriebs nutzbar wird, müssen die Parameter Sollgeschwindigkeit und Rampensteilheit, ein der Applikation angepasster Stromregelmodus mitsamt Grenzstromwerten, die Positionswerte für 1 bis 8 Impulse, die Impulsfilterzeit (Impulszug-Anerkennungszeit) sowie die Aktivierung dieser Funktion geschrieben und im Parameterspeicher gesichert werden. Die hierfür notwendigen Parameter sind folgend aufgeführt:

	Kommando	Bereich y	Antwort
- Position 0 (für 1 Impuls) schreiben:	nx0=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Position 1 (für 2 Impulse) schreiben:	nx1=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Position 2 (für 3 Impulse) schreiben: u.s.w. ...	nx2=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Position 7 (für 8 Impulse) schreiben:	nx7=yyyyyyy<LF>	-2147483648...2147483647	n*<LF>
- Impulsfilter- resp. Anerkennungszeit schreiben (10ms-Einheiten):	nxT=yyyy<LF>	0...1000 [10ms]	n*<LF>
- Impulszuggesteuerte Zielfahrt schreiben (aktivieren/deaktivieren):	nxP=y<LF>	1, 0	n*<LF>

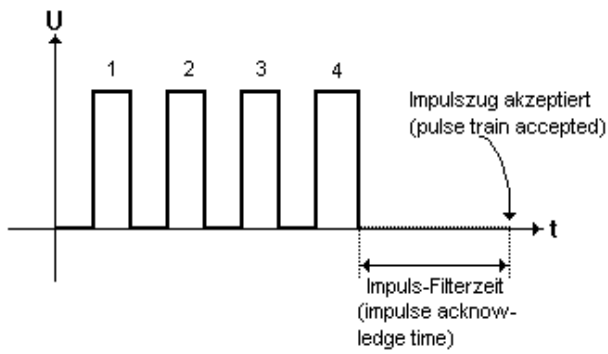


Bild x: Impulszug - Steuersignal an Eingang Index1 für autonomen Betrieb

Zur Beachtung: im autonomen Impulszug-Steuerungsbetrieb ohne Anbindung der seriellen Schnittstelle an einen Hostrechner sind allfällig auftretende Fehlerstati (z.B. Verlust der Position aufgrund aktiver Strombegrenzung u.s.w.) nur über die Status - LED's ersichtlich, da im µMC500V10 keine weiteren I/O - Signale zur Fehler-Rückmeldung verfügbar sind; digitale Statussignale zur Fehlermeldung sind nur ab Version µMC500V20 und höher verfügbar.

6. Sollgeschwindigkeit

Die Sollgeschwindigkeit bezeichnet den Drehzahl - resp. die Geschwindigkeitswert, der während einer Zielfahrt nach vollendeter Beschleunigungsphase so lange eingehalten wird, bis der Bremsvorgang eingeleitet wird. Im mode 4, d.h. mit kontinuierlicher Geschwindigkeitsregelung ohne Zielposition ist dies die Endgeschwindigkeit, die nach vollendeter Beschleunigungs- resp. Bremsrampe erreicht wird.

- Für die CPM motion control engine wird dieser Wert in Form von Inkrementen *10 pro Sekunde [10* l/s] vorzeichenbehaftet vorgegeben.
- Im Modus für kontinuierliche Geschwindigkeit ohne Zielposition (Modus 4) ist dies die Endgeschwindigkeit, die nach vollendeter Beschleunigung resp. Verzögerung erreicht wird.
- Die minimale Geschwindigkeit beträgt 10 Inc./s, die maximale 163840 Inc./s. Eine vorgegebene Sollgeschwindigkeit von z.B. 379 erzeugt am Antrieb im geschlossenen Regelkreis eine Inkrementalgeberfrequenz von genau 3790Hz. Die statische Abweichung wird mit maximal 2% spezifiziert.
- Die Bewegungsrichtung kann im mode 4 anstelle von vorzeichenbehafteten Geschwindigkeitswerten mit dem separierten Kommando nG2=x<LF> vorgegeben werden.
- Bei Positionierzielfahrten in den Modi 1, 2 und 3 hat das Vorzeichen der Sollgeschwindigkeit oder die explizite Vorgabe durch Kommando nG2=x<LF> keinen Einfluss auf die Bewegungsrichtung; diese wird ausschliesslich durch den kürzesten Weg zur Zielposition bestimmt.

Zur Beachtung: da die Sollgeschwindigkeit in der CPM motion control engine in Inkrementen pro Sekunde verarbeitet wird, ist die Geschwindigkeit umgekehrt proportional zur Auflösung des Inkrementalgebers:

- > je geringer die Auflösung des Inkrementalgebers ist, desto höher wird die Geschwindigkeit resp. Drehzahl bei gegebenem Sollwert
- > Je höher die Auflösung des Inkrementalgebers ist, desto geringer wird die Geschwindigkeit resp. Drehzahl bei gegebenem Sollwert

Beispiele für einen gegebenem Geschwindigkeitssollwert 100 d.h. 1000 Inkrementen pro Sekunde, Kommando nG0=100<LF>:

- Bei einem kleinen Servoantrieb mit einem Inkrementalgeber mit nur 10 Inkrementen pro Umdrehung generiert dieser Sollwert eine Drehzahl von 100 U/s
- Bei einem Servoantrieb mit einem Inkrementalgeber mit 2000 Inkrementen pro Umdrehung generiert dieser Sollwert eine Drehzahl von 0.5 U/s, d.h. eine halbe Umdrehung pro Sekunde.

	Kommando	Bereich x,y	Antwort
- Sollgeschwindigkeit schreiben:	nG0=xxxxx<LF>	-16384...16383	n*<LF>
- Sollgeschwindigkeit lesen:	nG0?<LF>	-16384...16383	n:yyyyy<LF>
- Bewegungsrichtung schreiben:	nG2=x<LF>	0, 1	n*<LF>

Alle Sollgeschwindigkeitswerte in [Inc./s *10]

Beispiel: 4G0=123<LF> erzeugt eine Endgeschwindigkeit von 1230 Inkrementen pro Sekunde (Beispiel für Device mit Adresse 4)

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

7. Zielpositionen

7.1 Zielposition absolut

Dieser Wert gibt die Zielposition in Inkrementen vor. Der Antrieb wird unmittelbar nach Empfang dieses Kommandos mit vorgegebener Rampe und Geschwindigkeit auf diese Position fahren.

Bei laufenden Positioniervorgängen kann auch eine neue absolute Zielposition vorgegeben werden; der Antrieb bremst dann mit linearer Rampe auf Geschwindigkeit 0 (mit Drehzahlnulldurchgang, wenn die neue Zielposition in entgegengesetzter Richtung liegt) und beschleunigt dann wieder mit vorgegebenem Rampenmodus in die neue Zielrichtung. Es kann maximal eine neue Zielposition während eines laufenden Positioniervorganges vorgegeben werden. Eine erneute Zielvorgabe kann nur nach erfolgtem Drehzahlnulldurchgang (ausgelöst durch vorhergehende Zielneuvorgabe) vorgegeben werden: nur 1 Verschachtelungsebene kann verarbeitet werden. Das repetitive Senden von immer gleichen Zielpositionswerten während laufender Zielfahrt ist jederzeit möglich, ohne dass der Antrieb durch Geschwindigkeitsnulldurchgänge geführt wird.

	Kommando	Bereich x,y	Antwort
- Zielposition absolut schreiben:	nX0=xxxxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Zielposition absolut lesen:	nX0?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>
- Sollposition lesen:	nX3?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>
- Istposition lesen:	nX2?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>

Beispiel: 4X0=67219<LF> setzt die Zielposition auf 67219 Inkremente (Beispiel für Device mit Adresse 4)

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

Erklärungen zu den Begriffen Zielposition, Sollposition, Istposition:

- Zielposition: der Positionswert, der angefahren werden soll.
- Sollposition: der momentane Sollwert für den Lageregler, der sich bei laufender Zielpositionsfahrt kontinuierlich ändert.
- Istposition: der momentane Istwert des Lagegebers, der die effektive resp. physikalische Momentanposition des Antriebs widerspiegelt.

7.2 Zielposition relativ

Das Setzen dieses Wertes ermöglicht eine Zielposition relativ zu der momentan gehaltenen Position anzufahren. Dies hat den Vorteil, dass der Hostrechner diese Zielposition nicht selbst errechnen muss, z.B. wenn von einem virtuellen Nullpunkt aus neu positioniert werden soll.

Der Antrieb wird unmittelbar nach Empfang dieses Kommandos mit vorgegebener Rampe und Geschwindigkeit auf diese Position fahren.

	Kommando	Bereich x,y	Antwort
- Zielposition relativ schreiben:	nX1=xxxxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Zielposition relativ lesen:	nicht möglich		
- Sollposition lesen:	nX3?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>
- Istposition lesen:	nX2?<LF>	-2147483648...2147483647	n:yyyyyyy<LF>

Beispiel: 4X1=5421<LF> setzt die neue Zielposition auf Momentanposition + 5421 Inkremente (Beispiel für Device mit Adresse 4)

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

Erklärungen zu den Begriffen Zielposition, Sollposition, Istposition:

- Zielposition: der Positionswert, der angefahren werden soll.
- Sollposition: der momentane Sollwert für den Lageregler, der sich bei laufender Zielpositionsfahrt kontinuierlich verändert.

- Istposition: der momentane Istwert des Lagegebers, der die effektive/physikalische Momentanposition des Antriebs widerspiegelt.

7.3 Position ohne Zielfahrt setzen

Das Schreiben dieses Kommandos ermöglicht es, dem Antrieb in seiner momentanen Position einen neuen numerischen Wert zuzuweisen, ohne dass sich der Antrieb an diese Position bewegen muss.

	<u>Kommando</u>	<u>Bereich x,y</u>	<u>Antwort</u>
- Pos. ohne Zielfahrt setzen:	nX5=xxxxxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Sollposition lesen:	nX3?<LF>	-2147483648...2147483647	n:yyyyyyyy*<LF>
- Istposition lesen:	nX2?<LF>	-2147483648...2147483647	n:yyyyyyyy*<LF>

Beispiel: 4X5=7890<LF> setzt die neue Position ohne Zielfahrt auf 7890 Inkremente (Beispiel für Device mit Adresse 4)

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

Erklärungen zu den Begriffen Zielposition, Sollposition, Istposition:

- Zielposition: der Positionswert, der angefahren werden soll.
- Sollposition: der momentane Sollwert für den Lageregler, der sich bei laufender Zielpositionsfahrt kontinuierlich ändert.
- Istposition: der momentane Istwert des Lagegebers, der die effektive/physikalische Momentanposition des Antriebs widerspiegelt.

8. Index- und Nullpunktdetektion

In den meisten zu positionierenden Systemen dient ein Referenz- Index- oder Nullpunkt als Bezugsposition für alle nachfolgend angefahrenen Zielpositionen. Dieser wird während der Initialisierungsphase des Systems aufgesucht und legt den internen, numerischen Nullpunkt fest.

Folgende Indexsuchprozeduren wurden im µMC500 integriert:

- Indexdetektion ohne Vorindex an Eingang Index0 wahlweise mit fallender oder steigender Signalfanke
- Indexdetektion mit Vorindex an Eingang Index0 wahlweise mit fallender oder steigender Signalfanke
- Indexdetektion mit Vorindex and Eingang Index1 und Nullpunkt an Eingang Index0, beide wahlweise mit fallender oder steigender Signalfanke
- Motorstromindex: Indexsuche auf Basis von mechanischen Endanschlägen

Die Nullpunktsuche kann mit 2 verschiedenen Indexsuchgeschwindigkeiten in unterschiedlichen Richtungen erheblich beschleunigt werden.

8.1 Nullpunkt- Detektion an Eingang Index0 ohne Vorindexschalter

Verursacht einen abrupten Stop des Antriebs nach der ersten fallenden oder steigenden Flanke an Eingang Index0. Die numerische Position, der Rampen- resp. Betriebsmodus, sowie die Solldrehzahl werden nach der Indexerkennung auf 0 gesetzt.

Damit diese Funktion genutzt werden kann, muss zusätzlich der Modus 4 (nP0=4<LF> für kontinuierliche Geschwindigkeitsregelung) sowie eine angemessene Geschwindigkeit und Bewegungsrichtung (nG0=xxx<LF> resp. nG0=-xxx<LF>) vorgegeben werden.

	<u>Kommando</u>	<u>Bereich x</u>	<u>Antwort</u>
- Nullpunktdetektion ohne Vorindex über Eingangssignal an Index0:	nY0=x<LF>	0 oder 1	n*<LF>
- Betriebsmodus f. Indexsuche schreiben:	nP0=x<LF>	4	n*<LF>
- Sollgeschwindigkeit schreiben:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

Der Kommandoparameter x (1 oder 0, z.B. 4Y0=1<LF>) dient zum Festlegen der Signalfanke, auf die eine Indexerkennung ausgelöst werden soll:

x=1: Eingang Index0 reagiert auf die positive, ansteigende Signalfanke.

x=0: Eingang Index0 reagiert auf die negative, fallende Signalfanke.

Wenn kein Parameter angegeben wird (z.B. 4Y0=<LF>), triggert Index0 auf negative Flanken.

8.2 Nullpunkt- Detektion an Eingang Index 0 mit Vorindexschalter an Eingang Index0

Diese Nullpunkt - Detektion verursacht beim ersten Auftreten einer fallenden oder steigenden Signalfanke am Eingang Index0 ein Abbremsen des Antriebs mit linearer Rampe von Sollgeschwindigkeit auf die (evtl. gegenläufige) Indexsuchgeschwindigkeit. Bei Auftreten der zweiten Nullpunktsignalfanke stoppt der Motor abrupt; alle positionierrelevanten Parameter sowie die Solldrehzahl werden auf 0 gestellt.

Folgende Konfigurationen für Vorindex resp. Nullpunktschalter können mit diesem Verfahren ausgewertet werden:

- 2 parallel verdrahtete Schalter für Vorindex und Nullpunkt so installiert, dass beim Über- und Auffahren 2 separate Signalfanken entstehen.
- 1 Indexschalter und Vorgabe einer negativen Indexsuchdrehzahl: mit nur einem einzigen Nullpunktschalter wird die Vorindex- und Nullpunkt- Signalfanke erzeugt.

Für beide Konfigurationen gilt: der Antrieb läuft mit hoher Geschwindigkeit auf den Vorindex- resp. Nullpunktschalter auf und überfährt diesen; der Vorindex- resp. Nullpunktschalter liefert eine erste Signalfanke und veranlasst ein Abbremsen des Antriebs auf Indexsuchgeschwindigkeit resp. Geschwindigkeit 0 und Wechsel auf Indexsuchgeschwindigkeit im Falle einer negativen Indexsuchgeschwindigkeit mit Richtungsumkehr.

Danach wird mit sehr viel kleinerer Indexsuchgeschwindigkeit in Richtung Nullpunktschalter (weiter- oder zurück-) gefahren, bis dieser bei Auffahrt die zweite Signalfanke erzeugt; der Antrieb stoppt abrupt. Nach dem Auslösen einer Nullpunkterkennung - gleichgültig ob mit oder ohne Indexschalter - wird die numerische Position, die Sollgeschwindigkeit sowie der Rampen- resp. Betriebsmodus auf 0 gestellt.

Damit diese Funktion genutzt werden kann, muss der Modus 4 (nP0=4<LF> für kontinuierliche Geschwindigkeitsregelung) sowie eine angemessene Geschwindigkeit und Bewegungsrichtung in Richtung

Nullpunktschalter (nG0=xxx<LF> resp. nG0=-xxx<LF>) sowie eine entsprechende, meist sehr viel kleinere Indexsuchgeschwindigkeit (nG1=xxx<LF> resp. nG1=-xxx<LF>) vorgegeben werden.

	Kommando	Bereich x,y	Antwort
- Indexsuche mit Vorindex über Eingangssignal an Index0	nY1=xy<LF>	0 oder 1	n*<LF>
- Betriebsmodus f. Indexsuche schreiben:	nP0=x<LF>	4	n*<LF>
- Indexsuchgeschwindigkeit schreiben:	nG1=xxxxx<LF>	-16384...16383	n*<LF>
- Sollgeschwindigkeit schreiben:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

Die Kommandoparameter x und y (1 oder 0, z.B. 4Y1=01<LF>) dienen zum Festlegen der Signalfanke, bei der eine Vorindex- resp. Indexerkennung ausgelöst werden soll:

x=1: Eingang Index0 reagiert auf positive, ansteigende Flanke bei erstem Vorindex-Signal

x=0: Eingang Index0 reagiert auf negative, fallende Flanke bei erstem Vorindex-Signal

y=1: Eingang Index0 reagiert auf positive, ansteigende Flanke bei zweitem Index-Signal

y=0: Eingang Index0 reagiert auf negative, fallende Flanke bei zweitem Index-Signal

Wenn kein Parameter angegeben wird (z.B. 4Y1=<LF>), triggert der Eingang Index0 in beiden Fällen auf fallende Flanken.

8.3 Nullpunkt - Detektion mit Vorindex an Eingang Index1 und Nullpunkt an Eingang Index0

Dieser Modus ermöglicht eine kombinierte Nutzung von Inkrementalgeber-Index (ein Impuls pro Umdrehung) in Zusammenhang mit einem Vorindex- Schalter. Hierzu wird das Inkrementalgeber-Indexsignal am Eingang Index0 und der Vorindexschalter am Eingang Index1 des µMC500 angeschlossen. Bei einer Indexsuche werden die Indexsignale des Inkrementalgebers an Eingang Index0 solange ignoriert, bis der am Eingang Index1 angeschlossene Vorindexschalter eine Signalfanke erzeugt. Danach wird auf Indexsuchgeschwindigkeit umgeschaltet und der Eingang Index0 für den Empfang einer Signalfanke aktiviert. Eine jetzt folgende Signalfanke des Inkrementalgeber-Indexes erzeugt einen abrupten Stop.

Nach dem Auslösen einer Nullpunkterkennung - gleichgültig ob mit oder ohne Indexschalter - wird die numerische Position, die Sollgeschwindigkeit sowie der Rampen- resp. Betriebsmodus auf 0 gestellt.

Damit diese Funktion genutzt werden kann, muss der Modus 4 (nP0=4<LF> für kontinuierliche Geschwindigkeitsregelung) sowie eine angemessene Geschwindigkeit und Bewegungsrichtung in Richtung Nullpunktschalter (nG0=xxx<LF> resp. nG0=-xxx<LF>) sowie eine entsprechende, meist sehr viel kleinere Indexsuchgeschwindigkeit (nG1=xxx<LF> resp. nG1=-xxx<LF>) vorgegeben werden.

	Kommando	Bereich x,y	Antwort
- Eingang Index0 für Nullpunkt-Detektion sowie Index1 für Vorindexdetektion schreiben:	nY2=xy<LF>	0 oder 1	n*<LF>
- Betriebsmodus f. Indexsuche schreiben:	nP0=x<LF>	4	n*<LF>
- Indexsuchgeschwindigkeit schreiben:	nG1=xxxxx<LF>	-16384...16383	n*<LF>
- Sollgeschwindigkeit schreiben:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

Die Kommandoparameter x und y (1 oder 0, z.B. 4Y2=01<LF>) dienen zum Festlegen der Flanke bei der eine Vorindex- resp. Indexerkennung ausgelöst werden soll:

x=1: Index1 reagiert auf positive, ansteigende Signalfanke

x=0: Index1 reagiert auf negative, fallende Signalfanke

y=1: Index0 reagiert auf positive, ansteigende Signalfanke

y=0: Index0 reagiert auf negative, fallende Signalfanke

Wenn kein Parameter angegeben wird (z.B. 4Y2=<LF>), triggern Index0 und Index1 auf negative Flanken.

8.4 Indexdetektion mittels Motorstrom

Dieses Verfahren ermöglicht den Verzicht auf den Indexschalter zur Nullpunkt- Detektion, falls aus konstruktiven oder kostenmässigen Gründen kein solcher integriert werden kann. Die Nullposition wird dann an einem mechanischen Endanschlag definiert, bei dessen Anfahrt der momentane Istwert des Motorstroms über eine definierte Schwelle ansteigt. Steigt nach dem Aktivieren dieses Betriebsmodus der Istwert über den Sollwert, wird ein Reset aller positionierrelevanten Parameter ausgeführt: der Antrieb stoppt abrupt, die numerische Position, die Sollgeschwindigkeit sowie der Rampen- resp. Betriebsmodus werden auf 0 gestellt.

Damit diese Funktion genutzt werden kann, muss der Modus 4 (nP0=4<LF> für kontinuierliche Geschwindigkeitsregelung), eine angemessene Geschwindigkeit und Bewegungsrichtung in Richtung Endanschlag (nG0=xxx<LF> resp. nG0=-xxx<LF>) sowie ein angepasster Soll- resp. Maximalstrom zur Indexdetektion (nB0=xxx<LF> in [mA]) vorgegeben werden.

	<u>Kommando</u>	<u>Bereich x</u>	<u>Antwort</u>
- Indexdetektion via Antriebsstrom schreiben:	nY3=1<LF>	-	n*<LF>
- Maximal- Soll- Detektionsstrom schreiben:	nB0=xxx<LF>	0...800 [mA]	n*<LF>
- Betriebsmodus f. Indexsuche schreiben:	nP0=x<LF>	4	n*<LF>
- Sollgeschwindigkeit schreiben:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

9. Abtasten der Momentanposition

Diese Funktion ermöglicht ein Abtasten und Sichern der momentanen Istposition bei fallender oder steigender Flanke am Eingang Index1. Nach Abfangen dieses Positionswertes wird diese Funktion automatisch deaktiviert. Die Initialisierung des Einganges Index1 sowie das Auslesen der abgetasteten, zum Zeitpunkt der Signalfanke gesicherten Momentanposition wird durch folgende Kommandi ermöglicht:

	<u>Kommando</u>	<u>Bereich x,y</u>	<u>Antwort</u>
- Eingang Index1 aktivieren:	nY5=x<LF>	0 oder 1	n*<LF>
- Abgefangene Position lesen:	nX4?<LF>	-2147483648...2147483647	n:yyyyyyyy<LF>

Beispiel: 4Y5=1<LF> initialisiert den Eingang Index1 für das Abfangen der Momentanposition bei steigender Flanke, 4Y5=0<LF> bei fallender Flanke. (Beispiel für Device mit Adresse 4)

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

10. Signalgetriggerte Zielposition setzen

Mit der Vorgabe dieses Positionswertes kann eine Zielfahrt aufgrund einer Signalfanke an Eingang Index1 ausgelöst werden. Hierzu muss die anzufahrende Zielposition sowie die Aktivierung dieser Funktion geschrieben werden, ausserdem der gewünschte Rampen- resp. Betriebsmodus für Zielfahrten (z.B. nP0=1<LF> für lineare Rampe) sowie eine angemessene Geschwindigkeit (nG0=xxx<LF>).

	<u>Kommando</u>	<u>Bereich x, y</u>	<u>Antwort</u>
- Signalgetriggerte Zielpos. schreiben:	nX6=xxxxxxx<LF>	-2147483648...2147483647	n*<LF>
- Eingang Index1 für Zielfahrt aktivieren:	nY4=x<LF>	1 oder 0	n*<LF>
- Betriebsmodus f. Indexsuche schreiben:	nP0=x<LF>	1..3	n*<LF>
- Sollgeschwindigkeit schreiben:	nG0=xxxxx<LF>	-16384...16383	n*<LF>

Der Parameter x (1 oder 0, z.B. 4Y4=1<LF>) dient zum definieren der Signalfanke bei der die Zielfahrt ausgelöst werden soll:

x=1: Eingang Index1 reagiert auf positive, ansteigende Signalfanke

x=0: Eingang Index1 reagiert auf negative, fallende Signalfanke

Wenn kein Parameter angegeben wird (z.B. 4Y4=<LF>), triggert der Eingang auf negative Signalfanken.

11. Unterbrechen einer laufenden Zielfahrt

Ein laufender Positioniervorgang kann jederzeit mit vorgegebener Bremsrampe unterbrochen werden. Das Kommando zur Positionierunterbrechung kann in allen 3 Rampen- resp. Betriebsmodi angewendet werden. Zu beachten ist, das eine Zielfahrtunterbrechung den Antrieb mit linearer Rampe abbremst. Nach einer Unterbrechung kann der Positioniervorgang durch Neusetzen der Zielposition wieder fortgesetzt werden.

	<u>Kommando</u>	<u>Bereich x,y</u>	<u>Antwort</u>
- Positioniervorgang unterbrechen:	nG3=1<LF>	-	n*<LF>

12. Steilheit der Beschleunigungs- und Bremsrampe

Im μMC500 besitzen mit Ausnahme des Rampenmodus 2 (Positionieren ohne Rampe) alle Modi eine Rampenfunktion, deren Steilheit in 2 Bereichen in jeweils 255 Stufen einstellbar ist.

Zu beachten ist, dass Steilheit und Bereichseinstellung der Rampe in Modi mit linearer Rampe (Positionieren mit linearer Rampe sowie kontinuierlicher Geschwindigkeitsregelung) nur im Stillstand des Antriebs geändert werden können. Wird z.B. versucht, die Rampe während eines Positioniervorganges im mode 1 zu verändern, erscheint eine mit einem Ausrufezeichen "!" versehene Rückmeldung. Bei Positioniervorgängen mit quadratischer (wegabhängiger) Rampe kann die Steilheit und die Bereichseinstellung jederzeit erfolgen; in mode 2 haben diese Einstellungen keinen Effekt.

	Kommando	Bereich x,y	Antwort
- Rampensteilheit schreiben:	nJ0=xxx<LF>	0...255	n* <LF>
- Bereichseinstellung schreiben:	nJ1=x<LF>	0, 1	n* <LF>

Beispiel: 4J0=10<LF> setzt die Rampensteilheit auf den Wert 10; : 4J1=1<LF> setzt die Bereichseinstellung auf 1 (Beispiel für Device mit Adresse 4)

Zum Lesen der Parameter werden die Gleichheitszeichen "=" im Kommandostring durch Fragezeichen "?" ersetzt.

Die Zeit, die der Antrieb zur Beschleunigung bis auf Endgeschwindigkeit resp. zum Abbremsen auf 0 benötigt, wird wie folgt errechnet (gilt für Positionieren mit linearer Rampe in mode 1 d.h. nP0=1<LF> sowie kontinuierlicher Bewegung ohne Zielposition in mode 4 d.h. nP0=4<LF>):

$$t = V \cdot \text{range} / T_{\text{sample}} \cdot \text{ramp}$$

t: Beschleunigungszeit [s] bis auf Geschwindigkeitssollwert
 V: Geschwindigkeitssollwert (Kommando nG0=xxxx<LF>)
 ramp: Rampensteilheit (Kommando nJ0=xxx<LF>)
 range: Bereichseinstellung der Rampe (Kommando nJ1=x<LF>) wobei gilt:
 range=1 bei nJ1=1<LF> resp. range=256 bei nJ1=0<LF>
 Tsample: Abtastrate (2560Hz)

Beispiel mit nG0=2560<LF>, nJ0=1<LF>, nJ1=1<LF>:
 Beschleunigungszeit $t = 2560 \cdot 1 / 2560 \cdot 1 = 1 \text{ s}$

Beispiel mit nG0=1000<LF>, nJ0=5<LF>, nJ1=1<LF>:
 Beschleunigungszeit $t = 1000 \cdot 1 / 2560 \cdot 5 = 0.078 \text{ s}$

Beispiel mit nG0=5000<LF>, nJ0=2<LF>, nJ1=1<LF>:
 Beschleunigungszeit $t = 5000 \cdot 1 / 2560 \cdot 2 = 0.97 \text{ s}$

Beispiel mit nG0=5000<LF>, nJ0=5<LF>, nJ1=1<LF>:
 Beschleunigungszeit $t = 5000 \cdot 1 / 2560 \cdot 5 = 0.97 \text{ s}$

Beispiel mit nG0=5000<LF>, nJ0=1<LF>, nJ1=1<LF>:
 Beschleunigungszeit $t = 5000 \cdot 1 / 2560 \cdot 1 = 1.95 \text{ s}$

Beispiel mit nG0=400<LF>, nJ0=10<LF>, nJ1=0<LF>:
 Beschleunigungszeit $t = 400 \cdot 256 / 2560 \cdot 10 = 4 \text{ s}$

13. Status 0, 1 und 3 lesen

13.1 Status 0

Der **Status 0** gibt Auskunft, welche Aktionen der Lage- und Geschwindigkeitsregler gerade ausführt, ob gerade eine Beschleunigungs- oder Bremsrampe gefahren wird, die vorgegebene Zielposition erreicht ist und einiges mehr. Die Information wird in Form eines ASCII-Strings mit insgesamt 10 Zeichen vermittelt, wobei das erste Zeichen das rückmeldende Modul / Device, das zweite Zeichen ein Doppelpunkt (:) und die restlichen 8 Bytes die Statusinformation bilden, z.B 9:00011010<LF>

	Kommando	Bereich x,y	Antwort
- Interner Status lesen: Reihenfolge der bytes:	nQ0?<LF>	00000000...11111111	n:yyyyyyyy<LF> 1.....8

Die einzelne Bytes (y) haben folgende Bedeutung wenn 1:

Byte	Bedeutung
1	Position 0 erreicht
2	Motorstrombegrenzung / Regelung war resp. ist aktiv
3	Tracking error; Soll- und Istposition weichen um x Inkremente voneinander ab (parametrisierbar)
4	Antrieb hat Sollgeschwindigkeit erreicht
5	Antrieb im Stillstand
6	Via Eingang Index1 initiierte Position erreicht
7	Drehrichtungsstellwert
8	vorgegebene Zielposition erreicht

13.2 Status 1

Der **Status 1** gibt Auskunft, ob die elektronische Überlastsicherung, der Kurzschluss- oder Motorschutz aktiv oder die Endstufentemperatur zu hoch ist u.s.w.. Die Information wird in Form eines ASCII-Strings mit insgesamt 10 Zeichen vermittelt, wobei das erste Zeichen das rückmeldende Modul, das zweite Zeichen ein Doppelpunkt (:) und die restlichen 8 Bytes die Statusinformation bilden, z.B 9:10010010<LF>

Der **Status 1** kann - mit Ausnahme der Übertemperatur - und Unterspannungsabschaltung - mit dem Kommando nZ1<LF> zurückgesetzt werden.

	Kommando	Bereich x,y	Antwort
- Statuswort 1 lesen: Reihenfolge der bytes:	nQ1?<LF>	00000000...11111111	n:yyyyyyyy<LF> 1.....8
- Status 1 löschen:	nZ1=1<LF>	-	n*<LF>

Die einzelne Bytes (y) haben folgende Bedeutung wenn 1:

Byte	Bedeutung
1	Übertemperatur Endstufe (>110C)
2	Betriebsspannung <8V +-10% (Unterspannung)
3	Elektronische Überlastabsicherung aktiv; IoM>0.8A @ t > 1s
4	Elektronischer Kurzschlusschutz aktiv; IoM>3A @ t > 10us
5	Istposition ungleich Sollposition aufgrund aktiver Strombegrenzung resp. Motorschutz
6	Übersteuerungsüberwachung; PWM-Stellwert >=zulässiger Wert (parametrisierbar)
7	Lageregler / Endstufe deaktiviert aufgrund Übersteuerung/Kurzschluss/Motorschutz oder beabsichtigter Deaktivierung
8	Überspannung, Betriebsspannung >32V

Beispiel Rückantwort von Adresse 4: 4:00101000<LF>

13.3 Status 3

Der **Status 3** ist die Summe aller Fehler im Status 1 und beträgt 0, wenn keine Fehlerzustände vorliegen.

Die Information wird in Form eines ASCII-Strings mit insgesamt 3 Zeichen vermittelt, wobei das erste Zeichen das rückmeldende Modul, das zweite Zeichen ein Doppelpunkt (:) und das letzte Bytes die Statusinformation bildet, z.B 9:3<LF>

	<u>Kommando</u>	<u>Bereich x,y</u>	<u>Antwort</u>
- Status 3 lesen:	nQ3?<LF>	0...7	n:y<LF>

x=0: keine Fehlerzustände

x>0: Fehlerzustände detektiert, Maximalwert 7

14. Reset, Deaktivieren, Reaktivieren

Mit dem Kommando nZ0=1<LF> werden alle positionierrelevanten Parameter d.h. Sollposition, Istposition, Zielposition, Rampen- resp. Betriebsmodi, Sollgeschwindigkeit, Übersteuerungs- Motorschutz- und Strombegrenzungsstati sowie alle relevanten internen Register auf 0 zurückgesetzt. Falls keine permanenten Fehlerzustände vorliegen, werden hiermit auch alle fehlerrelevanten Statusbits im Status 0 und 1 zurückgesetzt, so dass der Antrieb daraufhin in stehender Lageregelung in mode 0 verharrt. Alle erforderlichen Parameter wie Sollgeschwindigkeit, Rampenprofil u.s.w. müssen danach wieder neu geschrieben resp. aus dem nichtflüchtigen Parametersatz geladen werden.

Das Kommando nZ1=1<LF> verursacht einen Reset aller relevanten Antriebsschutzflags, ausgelöst durch zu grossen Stellwert (PWM-Wert) oder zu grossen tracking error (Abweichung Soll - Istposition). Diese Flags werden nur aktiv, wenn die Werte für die Strombegrenzung resp. Regelung für den Antrieb in Zusammenhang mit der Betriebsspannung zu hoch eingestellt wurden, d.h. der Stromregler nicht begrenzend eingreifen kann; siehe Kapitel Motorstromregelung und Begrenzung. Das Rücksetzen der Motorschutzflags belässt alle anderen Parameter auf ihren eingestellten Werten. Der Lageregler wird nach dem Rücksetzen dieser Flags reaktiviert, so dass unmittelbar danach wieder neue Zielpositionswerte angefahren werden können, sofern keine weiteren oder neuen Fehlerzustände vorliegen.

Das Kommando nZ2=1<LF> führt zu einer Deaktivierung des Antriebs, so dass dieser durch äussere Kräfte frei beweglich wird; die Istposition wird hierbei laufend nachgeführt. Eine nachfolgend initiierte Reaktivierung mittels Kommando nZ3=1<LF> lädt die Soll- und Zielposition mit dem laufend nachgeführten Ist-Positionswert und reaktiviert daraufhin die Lageregelung des Antriebs.

	<u>Kommando</u>	<u>Bereich x,y</u>	<u>Antwort</u>
- vollständigen Reset auslösen:	nZ0=1<LF>	-	n*<LF>
- Reset Antriebsschutzflags:	nZ1=1<LF>	-	n*<LF>
- Lageregelung deaktivieren:	nZ2=1<LF>	-	n*<LF>
- Lageregelung reaktivieren:	nZ3=1<LF>	-	n*<LF>

15. Eingangswerte von Index0 und 1, Betriebsspannung, Ist - Strom und Temperatur lesen

Die Eingangswerte können mit folgenden Kommandi gelesen werden:

	Kommando	Bereich x	Antwort
- Index0 lesen:	nI0?<LF>	0, 1	n:x<LF>
- Index1 lesen:	nI1?<LF>	0, 1	n:xLF>
- Betriebsspannung lesen:	nI5?<LF>	5,00..32,00	n:xx.xxV<LF> [V]
- Ist-Strom des Antriebs lesen:	nI6?<LF>	0.05..0.80	n:x.xxA<LF> [A]
- Temperatur lesen:	nI9?<LF>	0..100	n:xxC<LF> [°C]
- Analogspannung 0 lesen: (nur μMC500V20 resp. V12)	nIA?<LF>	0..1023	n:xxxx<LF>

Bemerkung bezüglich nI0<LF>: "I" ist das grosse "i" und steht für "Input"

15.1 Ausgangswerte schreiben

Die Ausgangswerte können mit folgenden Kommandi geschrieben werden:

	Kommando	Bereich x	Antwort
- Ausgang 0 schreiben: (nur μMC500V20 resp. V12)	nO0=x<LF>	0, 1	n:x<LF>

Zum Lesen des Zustandes des Ausgangs wird das Gleichheitszeichen "=" im Kommandostring durch ein Fragezeichen "?" ersetzt.

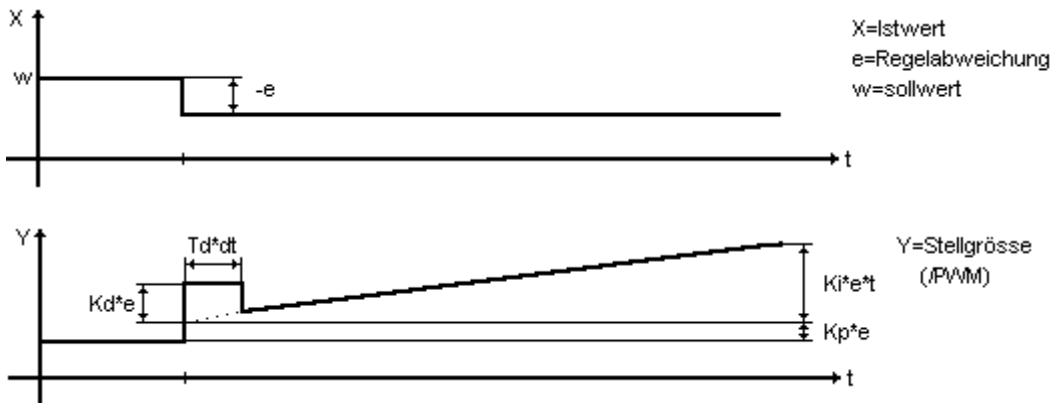
Achtung: nach erfolgtem Schreiben des Ausgangs 0 steht dieser nicht mehr als Statusausgang für Fehlermeldungen zur Verfügung!

16. Übertragungsfunktionen des Lagereglers

Damit der Lage/Drehzahlregler für jede Applikation optimale Ergebnisse bezüglich Einschwingverhalten und Regelabweichung gewährleisten kann, sind im µMC500 alle 6 Parameter der Übertragungsfunktion frei definierbar gestaltet: Proportionalanteil K_{pp} , Differentialanteil K_{dp} , Integralanteil K_{ip} , Zeitkonstante des Differentialanteils T_{dp} sowie die Begrenzung des Integralanteils R_{ip} können in weiten Bereichen an das Regelschleifenverhalten angepasst werden. Die Abtastrate ist fest mit 2.560KHz definiert.

Die im µMC500 implementierte CPM motion control engine erzeugt ihre eigene space vector - Kommutation mittels kontinuierlicher Lageregelung des Antriebs, so dass dieses System ohne separierten Geschwindigkeits-Regler auskommt. Bei Positioniervorgängen wird auch kein Geschwindigkeits - Estimator für die Beschleunigungs- / Bremsrampen benötigt; die erforderliche Momentangeschwindigkeit bei Zielfahrten wird direkt aus der Momentanposition abgeleitet. Dies vereinfacht die Parametrisierung des Reglers erheblich und verleiht dem System intrinsische Stabilität (Patentiertes Verfahren).

Die Grafik veranschaulicht das Regelverhalten eines PID-Reglers mittels einer Sprungfunktion des Istwertes:



Die folgenden Kommandi erlauben den Zugriff auf die Regelparameter der CPM motion control engine:

	Kommando	Bereich x	Antwort	Default
- Proportionalanteil K_{pp} schreiben [1]:	nA0=xxx<LF>	0...255	n*<LF>	5
- Integralanteil K_{ip} schreiben [1]:	nA3=xxx<LF>	0...255	n*<LF>	10
- Differentialanteil K_{dp} schreiben [1]:	nA1=xxx<LF>	0...255	n*<LF>	10
- Zeitkonst. Differential -Anteil T_{dp} schreiben [400us]:	nA2=xxx<LF>	1...255	n*<LF>	2
- Begrenzung R_{ip} schreiben [1]:	nA4=xxx<LF>	1...127	n*<LF>	3
- Abtastrate [Hz]:	-	-	-	2.560KHz

Alle 6 Parameter können jederzeit ohne Rücksicht auf den momentanen Betriebszustand geändert werden. Die in dieser Tabelle angegebenen default-Werte sind die Werte, die von Werk aus im Parameter-Speicher abgelegt und bei power-up geladen werden. Diese sind so gewählt, dass ein grosses Spektrum an Antriebs- und Inkrementalgeberkombinationen bei erstem power-up prinzipiell stabil sein sollte (unkritische "low - performance - Einstellung").

Zum Lesen der Parameter wird das Gleichheitszeichen "=" im Kommandostring durch ein Fragezeichen "?" ersetzt.

17. Stromregler, Strombegrenzung, Übersteuerungsüberwachung, Schleppfehler, Motorschutz

im µMC500 sind insgesamt 4 in weiten Bereichen parametrisierbare Strombegrenzungs- und Stromregelmodi integriert, die einen umfassenden Antriebs- Mechanik- und Endstufenschutz gewährleisten. Hinzu kommen der separierte Kurzschluss- und Übertemperaturschutz der Leistungsendstufe, die globale Übertemperaturüberwachung, der Übersteuerungsschutz sowie die Tracking - Überwachung.

Zusammenfassend sind dies folgende Stromregel- und Schutzmechanismen:

- Stromregelmodus 0: Übersteig der Antriebsstrom den Sollwert (Parameter Konstantstrom I_{oc}), wird der Antrieb angehalten, d.h. z.B. eine laufende Zielfahrt wird abrupt unterbrochen. Die Auslösezeit beträgt $(I_{st} - I_{oc}) \times 400 \mu s / 1A$. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Stromregelmodus 1: Übersteig der Antriebsstrom den Sollwert (Parameter Konstantstrom I_{oc}), wird der Antrieb rastend Strombegrenzt, ähnlich dem Übersteigen des momentanen Drehmoments an einem Schrittmotor bei niedriger Drehzahl. Während eines laufenden Positioniervorganges wird ein repetitiver Wiederanlaufversuch mit einer Frequenz zwischen 2...20Hz je nach Momentangeschwindigkeit und Flexibilität des mechanischen Systems durchgeführt. Die wiederholende Auslösezeit beträgt $(I_{st} - I_{oc}) \times 400 \mu s / 1A$. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Stromregelmodus 2: Konstantstromregelung mit frei definierbaren Regelparametern für den Proportional- und Integralanteil sowie den Konstantstrom I_{oc} . Dieser Stromregler lässt durch sein integrierendes Verhalten kurzzeitig auch hohe Spitzenströme bei Beschleunigungs- und Bremsrampen zu, ohne dass dieser eingreifend wirkt. Dieser Zustand wird, solange die Stromregelung aktiv ist im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Stromregelmodus 3: $I^2 \cdot t$ - Stromregelung mit parametrisierbarem Grenzstrom I_{om} , Konstantstrom I_{oc} sowie Zeitkonstante $\tau_{I_{oc}}$. Die Zeitkonstante $\tau_{I_{oc}}$ kann in einem Bereich von 10ms...100s eingestellt werden. Bei Überschreiten des Konstantstromes I_{oc} wird der Strom im Antrieb primär auf den Grenzstrom I_{om} begrenzt, der in der Regel ein Mehrfaches des Konstantstromes I_{oc} beträgt. Darauf folgend wird die Differenz des Antriebsstromes zum Konstantstrom I_{oc} quadriert und einem numerischen Integrator für die Zeitberechnung übergeben, der die zeitlich abhängige Reduktion des Sollstromes bis hinab zum Konstantstrom I_{oc} berechnet. Nach vollständigem Ablauf der $I^2 \cdot t$ - Zeitkonstante $\tau_{I_{oc}}$ fließt im Antrieb nur noch der (geringere) Konstantstrom I_{oc} . Dies ist der Antriebsstrom, der unter allen temperaturmässigen Bedingungen kontinuierlich, d.h. zur Einhaltung der thermischen Stabilität dauerhaft fließen darf. Dieser Stromregler ermöglicht durch sein integrierendes Verhalten kurzzeitig auch hohe Spitzenströme bei Beschleunigungs- und Bremsrampen, ohne dass dieser eingreifend wirkt. Dieser Zustand wird, solange die Stromregelung aktiv ist im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Übersteuerungs- resp. Stellwertüberwachung: Der Stellwert (das PWM-Signal) sowie die zugehörige Filterzeit können in einem grossen Bereich an die Anforderungen angepasst werden. Bei Überschreiten des Stellwertes und nach Ablauf der Filterzeit wird die Endstufe deaktiviert. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Tracking error resp. Schleppfehlerüberwachung: Überwacht die Abweichung zwischen Soll- und Istposition in Zusammenhang mit einer Filterzeit; beide Parameter können in einem grossen Bereich an die Anforderungen angepasst werden. Bei Überschreiten des Schleppfehlers und nach Ablauf der Filterzeit wird die Endstufe deaktiviert. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Unter- und Überspannungsüberwachung: Betriebsspannungen ausserhalb des zulässigen Bereiches deaktivieren die Endstufe. Bei Überspannungsereignissen wird der Antrieb innerhalb $< 10 \mu s$ kurzgeschlossen. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Globale Übertemperaturüberwachung: schützt das gesamte Modul vor Betriebstemperaturen $> 100^\circ C$; bei Temperaturen oberhalb dieser Grenze wird die Endstufe deaktiviert. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.
- Kurzschluss - Schutz: schützt die Antriebsendstufe auf Kurzschlüsse gegeneinander, gegen Masse und gegen Speisung. Die Endstufe verfügt ausserdem über einen autonomen Übertemperatur- und Überlastschutz. Dieser Zustand wird im Status1 hinterlegt und auf der LED-Statusanzeige angezeigt.

Die Kommandi, die den Zugriff auf diese Parameter ermöglichen, sind nachfolgend aufgelistet:

	<u>Kommando</u>	<u>Bereich x</u>	<u>Antwort</u>	<u>Default</u>
- Stromregelmodus:	nB3=x<LF>	0...3	n*<LF>	1
- Konstantstrom I _{oc} :	nB0=xxx<LF>	0...1023 [mA]	n*<LF>	200mA
- Grenzstrom I _{om} für I ² *t :	nB1=xxx<LF>	0...1023 [mA]	n*<LF>	800mA
- Zeitkonstante τ _{loq} für I ² *t :	nB2=xxx<LF>	0...10000 [10ms]	n*<LF>	10s
- Max. Stellwert für Übersteuerungsüberw.:	nZ4=xxx<LF>	0...255 [PWM]	n*<LF>	250
- Filterzeit für Übersteuerungsüberwachung:	nZ5=xxx<LF>	0...10000 [10ms]	n*<LF>	5s
- Max. tracking error / Schleppfehler:	nZ6=xxx<LF>	0...2147483647[!]	n*<LF>	63Inc.
- Filterzeit für tracking error / Schleppfehler:	nZ7=xxx<LF>	0...10000 [10ms]	n*<LF>	5s
- Proportionalanteil K _{pc} :	nB4=xx<LF>	0...32	n*<LF>	1
- Integralanteil K _{ic}	nB5=xx<LF>	0...32	n*<LF>	10

Die in dieser Tabelle angegebenen default-Werte sind die Werte, die von Werk aus im Parameter-Speicher abgelegt und bei power-up geladen werden.

Zum Lesen der Parameter wird das Gleichheitszeichen "=" im Kommandostring durch ein Fragezeichen "?" ersetzt.

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

Die Integration von CRC - Summen ermöglichen eine laufende Verifizierung aller auf der seriellen Leitung transferierten Daten, wenn diese z.B. über Schleifringe, lange Leitungen (>>10m) und/oder durch stark elektromagnetisch belastete Umgebung sicher übertragen werden müssen.

Werkseitig ist die CRC - Auswertung deaktiviert; sie kann aber mittels des unten beschriebenen Kommandos aktiviert und nichtflüchtig im Parametersatz gesichert werden. Nach erfolgter Aktivierung müssen alle zum Regler gesendeten Kommandi zwingend mit dieser CRC - Summe verknüpft werden, damit diese als gültig anerkannt und rückbeantwortet werden. Verfälschte resp. fehlende Checksummen oder Kommandi haben zur Folge, dass die empfangenen Daten weder akzeptiert noch rückgemeldet werden (inexistente Kommunikation).

Es existiert kein Kommando zum Deaktivieren der CRC- Prüfung! Sie kann nach dem Sichern im Parametersatz nur durch Laden der Defaultwerte - natürlich nur mit korrekter CRC - wieder deaktiviert werden.

Bei aktivierter CRC müssen alle in diesem Dokument beschriebenen Kommandi folgende Struktur besitzen:

nks=ppppppp&yyyyyy<LF> zum Schreiben resp. nks?&yyyyyy<LF> zum Lesen

Das Zeichen "&" markiert den Beginn der Checksumme. Der nach dem Zeichen & folgende Zahlenwert yyyyyy ist die CRC - Checksumme; diese muss in Form eines ASCII-Zahlenwertes mit maximal 5 Stellen + Vorzeichen, d.h. in einem Bereich von -32768 bis 32767) vorliegen.

	<u>Kommando</u>	<u>Bereich y</u>	<u>Antwort</u>
- CRC aktivieren:	n&<LF>	-	n*&yyyyyy<LF>

Wobei n*&yyyyyy<LF> die Rückantwort der aktivierten CRC ist; aufgrund der Tatsache, dass die CRC über die gesamte Zeichenkette inkl. "&" gerechnet wird, ist diese auch von der Device- resp. Moduladresse abhängig.

Beispiele für Kommandi ohne und mit CRC:

<u>Kommando ohne CRC</u>	<u>Kommando mit CRC</u>	<u>Rückmeldung</u>
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>

Die in diesen Beispielen angegebenen CRC - Werte sind effektiv zutreffende, d.h. stimmende Werte

19. Sichern und Laden der Parameter, Defaultwerte

Alle Parameter können mithilfe des weiter unten beschriebenen Kommandos im nichtflüchtigen Speicherbereich gesichert; diese werden nach power-up automatisch wieder geladen. Diese Funktion ist insbesondere zur Initiierung autonomer Betriebsmodi, in denen nach vollzogener Konfiguration keine weitere Kommunikation über die serielle Schnittstelle stattfindet, unverzichtbar. Gesichert werden alle nachfolgend summarisch aufgelisteten Parameter:

- Parameter des Lagereglers
- Parameter des Stromreglers
- Parameter des Übersteuerungs- Motor- und Mechanikschutzes
- Rampen- resp. Betriebsmodi
- Parameter und Modi für autonomen Betrieb
- Sollgeschwindigkeit und Rampenwerte
- Die aktuelle Ist-Position
- Die Device- resp. Moduladresse
- Die Baudrate der seriellen Schnittstelle, 19200Bd oder 115200Bd

Die gesicherten Parameter können auch explizit im laufenden Betrieb geladen werden; die evtl. bereits über die serielle Schnittstelle geschriebenen Werte werden daraufhin mit denen aus dem nichtflüchtigen Parametersatz überschrieben.

Im Falle fehlerhafter Parametrisierung können auch fest codierte Defaultwerte (Werksvorgaben) geladen werden, die einen prinzipiell stabilen Betrieb sowie einen hinreichenden Antriebsschutz für typische Anwendungen gewährleisten sollen. Die Werte der Werksvorgaben (Default) sind in der zusammenfassenden Tabelle aller Kommandi aufgelistet, siehe Tabelle 1, Seite30. Die Werksvorgaben können bei Bedarf auch an kundenspezifische Vorgaben angepasst werden. Achtung: der Ladevorgang für die Werks- resp. Defaultvorgaben hat immer auch einen Sicherungsvorgang dieser Werksvorgabe im nichtflüchtigen Speicher zur Folge.

Zum Sichern und Laden der Werks- und Anwenderspezifischen Parameter dienen folgende Kommandi:

	<u>Kommando</u>	<u>Antwort</u>
- Parametersatz sichern:	nE1=1<LF>	n*<LF>
- Parametersatz laden:	nE2=1<LF>	n*<LF>
- Werks- (default-) Werte laden:	nE3=1<LF>	n*<LF>
- Seriennummer lesen:	nE4?<LF>	n:S/N jj-nnnnn<LF> z.B. S/N 10-02537

Die Seriennummer kann durch den Anwender nicht geändert werden (Passwortgeschützter Wert).

Beispiel: 4E1=1<LF> sichert alle aktuellen Werte im Parametersatz 1 (das µMC500 besitzt nur einen einzigen Satz)

Die Lebensdauer des nichtflüchtigen Parameterspeichers beträgt typisch 250'000, minimal 20'000 Lösch/Schreibzyklen für 10 Jahre Datenerhalt. Obwohl diese Lebensdauer mehr als ausreichend für eine typische Anwendung ist, sollten Parameter - Sicherungsvorgänge auf die minimal notwendige Anzahl reduziert bleiben; es wird daher empfohlen, zyklische Sicherungsvorgänge mit hoher Repetitionsrate zu vermeiden.

20. Adresse des Reglers schreiben und sichern

Die individuelle Adresse "n" jedes Antriebsreglers wird über die serielle Schnittstelle vorgegeben, d.h. es stehen im µMC500 keine Schalter in Form von DIP-switch o.ä. zur Vorgabe dieses Adressenwertes zur Verfügung.

Besonders zu beachten ist, dass das Kommando zur Adressenvergabe ein broadcast-Kommando ist, dass von jedem Antriebsregler mit jeder beliebigen Moduladresse akzeptiert wird. Dies hat zur Konsequenz, dass dieses Kommando im RS485-Netzwerkverbund mit mehreren angeschlossenen Antriebsreglern nicht gesendet werden darf, da alle Antriebsregler die gleiche Moduladresse annehmen würden; eine individuelle Kommunikation wäre als Folge davon nicht mehr möglich.

Die Adressenvergabe wird daher mit nur einem, isoliert an einem RS485-Bus angeschlossenen Exemplar durchgeführt, bevor dieses in einem RS485-Netzwerkverbund mit mehreren Antriebsreglern integriert und individuell über seine Adresse ansprechbar wird.

Zu beachten ist ausserdem, dass dem Schreibvorgang für die Moduladresse ein Sicherungsvorgang aller Parameter (nE1=1<LF>) folgen muss, damit die individuelle Adresse auch nach erneutem power-up wieder zur Verfügung steht!

Die Werksvorgabe der Moduladresse für alle µMC500 ist 4, siehe Beispiele aller Kommandi in diesem Dokument. Die Device- resp. Moduladresse kann zwischen 1 und 9 eingestellt werden; dies geschieht am einfachsten mithilfe der PC-Software STM7000.

Das Kommando zum festlegen der Modul- resp. Deviceadresse lautet:

	<u>Kommando</u>	<u>Bereich x</u>	<u>Antwort</u>
- Modul- resp. Deviceadresse schreiben:	nAdress=x<LF>	1...9	n*<LF>
- Parametersatz sichern:	nE1=1<LF>	-	n*<LF>

Unmittelbar nach Empfang und Auswertung dieses Kommandos wird der angeschlossene Antriebsregler nur noch über die neu definierte Adresse ansprechbar. Danach: Sichern aller Parameter mit nE1=1<LF> nicht vergessen!

Beispiel: 4Adress=1<LF> ändert die Adresse des angeschlossenen Moduls mit beliebiger Adresse auf Adresse 1, nE1=1<LF> sichert den Parametersatz inkl. Modul- resp. Deviceadresse.

21. Firmwareupdate

Die Firmware im µMC500 kann ohne weitere Massnahmen auch innerhalb eines RS485-Netzwerks mit mehreren µMC500 individuell aktualisiert werden (in system / in application); der Bootloader akzeptiert adressierte, unverschlüsselte HEX- sowie verschlüsselte CHX-Zeilen.

Die PC-Software STM7000 ermöglicht ein komfortables Update der Firmware auf das jeweils selektierte Zielmodul. Hierzu werden die einzelnen HEX- resp. CHX- Zeilen der Firmware um die Zieladresse erweitert, z.B. 9:04207B0012202F22DE<LF> für eine HEX-Zeile an Zieladresse 9. Bei einzeilig initiiertem Transfer wird der Empfang jeder einzelnen Zeile durch das Zielmodul quittiert (9*<LF>), danach folgt die nächste Zeile mit Quittierung u.s.w.

Bei mehrzeiligem Transfer von bis zu maximal 15 Zeilen wird der Empfang erst nach der 15. Zeile quittiert. Dieses Verfahren erlaubt einen stark beschleunigten Updatevorgang in Systemen mit USB zu RS485-Adapter. Die Ladegeschwindigkeit ist jedoch in hohem Masse abhängig von der jeweiligen Konfiguration sowie der Systeminternen Belastung des Computers. Die PC-Software STM7000 entscheidet aufgrund der Bootloader-Version im Antriebsregler, ob ein ein- oder mehrzeiliger Transfer von HEX- oder CHX- Zeilen an das Zielmodul möglich ist; wann immer möglich, wird der mehrzeilige Transfer bevorzugt. Die Ladezeit beträgt bei mehrzeiligem Transfer typischerweise ca. 5..15s

IMG Antriebstechnik stellt Firmwareupdates sowie kundenspezifische Varianten prinzipiell in Form von verschlüsselten CHX- Dateien zum Download bereit. Die CHX- Dateien besitzen eine Schlüssellänge von 2048 bits und werden erst innerhalb des Bootloaders im Antriebsregler entschlüsselt.

Beispiel einer CHX- Zeile mit Zieladresse 9: 9:°Çi±→• • Ñ»Ñ³-š%òù¹ÂÞ>Í¡°´«Ë§æðó´!íó´Ú-éË...öi<LF>

Informationen bezüglich Bootloader oder Verschlüsselungsprogramm werden prinzipiell nicht veröffentlicht.

Das Ziel von IMG Antriebstechnik ist jedoch, für seine Antriebsregler im Laufe der Zeit eine open - source - community nach dem Vorbild von Linux aufzubauen. Hierzu wird die Hardware bei gegebener Zeit in Form von Blockschaltbildern veröffentlicht, die eine individuelle Erstellung von eigenen Firmwarekonzepten mit unverschlüsseltem Download ermöglicht. Weitere Informationen sowie rechtliche Bedingungen hierfür folgen.

22. Erste Inbetriebnahme, Beispiele

22.1 Kontinuierliche Bewegung mit 1290Hz Inkrementalgeberfrequenz

4P0=4<LF> kontinuierliche Geschwindigkeitsregelung ohne Zielposition, mode 4 schreiben.
4G0=129<LF> Sollgeschwindigkeit auf 129, d.h. 1290 Inkremente pro Sekunde setzen.

Der Antrieb beschleunigt unmittelbar nach dem Schreiben von 4G0=129<LF> auf Sollgeschwindigkeit und läuft solange mit dieser Geschwindigkeit, bis ein anderer Sollgeschwindigkeitswert gesendet wird.

22.2 Zielposition 1456 Inkremente mit Geschwindigkeit 150 und quadratischer Beschleunigungs- / Bremsrampe anfahren:

4P0=3<LF> Quadratische, wegabhängige Rampe, mode 3 schreiben.
4G0=150<LF> Sollgeschwindigkeit auf 150, d.h. 1500 Inc./s setzen.
4J0=3<LF> Rampensteilheit auf 3 festlegen.
4X0=1456<LF> Zielposition 1456 Inkremente schreiben.

Der Positioniervorgang startet unmittelbar nach dem Schreiben der Zielposition.
Nachfolgende Zielfahrten werden durch Senden neuer Zielpositionen ausgelöst

Tabelle 1: Zusammenfassung Kommandi

	Kommando	Schreibbereich ("=")	Lesebereich ("?"	Default
<u>Lageregler</u>				
Proportionalanteil Lageregler [1]:	A0	0..255	0..255	20
Differentialanteil Lageregler [1]:	A1	0..255	0..255	10
Zeitkonst. Diff.- Anteil Td [1/400us]:	A2	1..255	1..255	2
Integralanteil Lageregler [1]:	A3	0...255	0..255	10
Vorbegrenzung I-Anteil Ri [1]:	A4	0...127	0..127	3
<u>Stromregler</u>				
Konstantstrom I _{oc} [mA]	B0	0..1023	0..1023	200mA
Grenzstrom I _{om} für I ² *t [mA]	B1	0..1023	0..1023	800mA
Zeitkonstante τI _{oq} für I ² *t [10ms]	B2	0..10000	0..10000	10s
Strombegrenzungs- resp. Regelmodus:	B3	0..3	0..3	1
Prop.- Anteil Kpc Stromregler [1]:	B4	0..32	0..32	1
Integralanteil Kic Stromregler [1]:	B5	0..32	0..32	10
Abtastezeit Stromregler [1/400us]:	B6	0..255	0..255	1
<u>Geschwindigkeit</u>				
Sollgeschwindigkeit [Inc./s *10]:	G0	-16384..16383	-16384..16383	0
Index- Suchgeschwindigkeit [Inc./s *10]:	G1	-16384..16383	-16384..16383	0
Bewegungsrichtung explizit	G2	0, 1	0, 1	0
Abbruch Zielfahrt	G3	1	-	-
<u>Rampe</u>				
Rampensteilheit [1]:	J0	1..255	1..255	1
Steilheitsbereich:	J1	0, 1	0, 1	1
<u>Indexsuche, Zielfahrt, Momentanposition</u>				
Index an Index0, abrupter Stop:	Y0	0, 1	0, 1	0
Index an Index0 mit Vorindex an Index0:	Y1	00, 01, 10, 11	0, 1	00
Index an Index0 mit Vorindex an Index1:	Y2	00, 01, 10, 11	0, 1	00
Index mittels Antriebsstrom- Detektion:	Y3	1	0, 1	0
Index1- getriggerte Zielfahrt aktivieren:	Y4	0, 1	0, 1	0
Index1 für Abfangen der Momentanpos.:	Y5	0, 1	0, 1	0
<u>Rampen- resp. Betriebsmodus</u>				
Betriebsmodus:	P0	0..7	0..7	0
<u>Status</u>				
Status 0 lesen:	Q0	-	00000000 bis 11111111	
Status 1 lesen:	Q1	-	00000000 bis 11111111	
Status 3 lesen:	Q3	-	0..7	
<u>Ziel- und Momentanpositionen</u>				
Zielposition absolut:	X0	-2147483648 bis 2147483647	-2147483648 bis 2147483647	0
Zielposition relativ:	X1	-2147483648 bis 2147483647	-	0
Istwert Momentanposition:	X2	-	-2147483648 bis 2147483647	0
Sollwert Momentanposition:	X3	-	-2147483648 bis 2147483647	0
Abgefangene Momentanposition:	X4	-	-2147483648 bis 2147483647	0
Zielposition ohne Zielfahrt:	X5	-2147483648 bis 2147483647	-2147483648 bis 2147483647	0
Index1 - getriggerte Zielposition:	X6	-2147483648 bis 2147483647	-2147483648 bis 2147483647	0

	Kommando	Schreibbereich ("=")	Lesebereich ("?")	Default
<u>Signalgesteuerter, autonomer Betrieb</u>				
Zielposition 0 für 1 Impuls an Index1:	x0	-2147483648 bis 2147483647	-2147483648 bis 2147483647	0
Zielposition 1 für 2 Impulse an Index1:	x1	-2147483648 bis 2147483647	-2147483648 bis 2147483647	100
Zielposition 2 für 3 Impuls3 an Index1: u.s.w. u.s.w.	x2	-2147483648 bis 2147483647	-2147483648 bis 2147483647	200
Zielposition 7 für 8 Impulse an Index1:	x7	-2147483648 bis 2147483647	-2147483648 bis 2147483647	700
Filterzeit für Impulszug [10ms]:	xT	0..1000	0..1000	1s
Impulszuggesteuerte Zielfahrt:	xP	0, 1	0, 1	0
Kont. Geschwindigkeit via Index0/1:	xS	0, 1	0, 1	0
<u>Reset, Antriebsschutz, Re/Deaktivieren</u>				
Vollständiger Reset:	Z0	1	-	-
Reset Antriebsschutzflags:	Z1	1	-	-
Lageregelung deaktivieren:	Z2	1	0, 1	0
Lageregelung reaktivieren:	Z3	1	0, 1	-
Stellwert für Übersteuerungsüberw.:	Z4	0..255	0..255	250
Filterzeit für Übersteuerungsüberw.:	Z5	0..10000	0..10000	5s
Tracking error / Schleppfehler:	Z6	0...2147483647	0...2147483647	63Inc.
Filterzeit tracking error / Schleppfehler:	Z7	0..10000	0..10000	5s
CRC auf Kommandostring aktivieren:	Z8	1	-	off
<u>Parametersatz, Defaultwerte, Seriennummer</u>				
Parametersatz sichern:	E1	1	-	-
Parametersatz laden:	E2	1	-	-
Werks- (default-) Werte laden:	E3	1	-	-
Seriennummer lesen:	E4	-	S/N jj-nnnnn jj=Produktionsjahr nnnn=fortlaufende Seriennummer	S/N jj-nnnnn
<u>Eingangswerte lesen</u>				
Index0 lesen:	I0	-	0, 1	-
Index1 lesen:	I1	-	0, 1	-
Analogspannung lesen (nur V20):	IA	-	0..1023	-
Betriebsspannung lesen:	I5	-	xx.xxV	-
Ist-Strom des Antriebs lesen:	I6	-	x.xxA	-
Temperatur lesen:	I9	-	xxC	-
<i>Bemerkung bezüglich nI0<LF>:</i> "I" ist das grosse "I" und steht für "Input"				
<u>Ausgangswerte schreiben</u>				
Ausgang 0 (nur V20 resp. V12):	O0	0, 1	0, 1	-
<u>Modultyp, Version, Datum, Hersteller</u>				
Infostring lesen:	^	-	µMC500 <Version> <Datum> IMG Antriebstechnik Switzerland	