

Technisches Handbuch MA4Q / MD4Q

Für Software Release 1.11A

Handbuch Version 0638 vom 20.09.2006

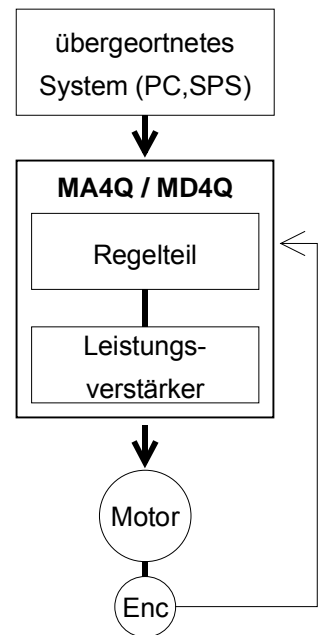
1 Einleitung und Inhaltsverzeichnis

Das MA4Q und das MD4Q sind Motorcontroller für AC (EC, resp. BL)- und DC - Servo - Motoren.

Unter Motorcontroller verstehen wir digitale Regelgeräte, welche die Position, die Geschwindigkeit und das Drehmoment eines Antriebes regeln.

Unsere Controller bestehen aus einem Regelteil und einer Endstufe (Leistungsverstärker)

- Der **Regelteil** ist zuständig für die Regelkreise (Drehmoment, Drehzahl und Position). Er ist die Schnittstelle zwischen Ihrem übergeordneten System und der Endstufe. Es spielt für den Regler grundsätzlich keine Rolle welcher Motortyp verwendet wird.
Das Positionssignal des Antriebes wird mittels optischem Impulsgeber (Encoder) erfasst.
- Die **Endstufe** (auch als Leistungsteil oder **Leistungsverstärker** bezeichnet) ist die Schnittstelle zwischen Regler und Motor. Sie wandelt den Stellwert des Reglers in Spannungen und Ströme für den anzutreibenden Motor um.
ACHTUNG: Die Endstufe muss zum angeschlossenen Motor passen (Strom, Spannung, Motorinduktivität, Kommutierungstyp, etc).



Achtung:

Je nach der verwendeten Anschlussspannung, kann bei Berührung irgendeines Geräteteiles **LEBENSGEFAHR** bestehen!

Technische Änderungen vorbehalten

Wir bemühen uns unsere Unterlagen aktuell zu halten. Dennoch kann es vorkommen, dass durch die Weiterentwicklung Angaben nicht mehr aktuell sind.

Haftungsausschluss

Wir lehnen jegliche Schadenersatzansprüche aufgrund von unsachgemäßem oder falschem Einsatz unserer Geräte ab. Beachten Sie unsere allgemeinen Geschäftsbedingungen.

1.1 Copyright

Das Technische Handbuch, die Kurzdokumentation und die Datenblätter sind unser geistiges Eigentum. Sie sind nur zum Gebrauch in Zusammenhang mit den beschriebenen elektronischen Geräten bestimmt.

1.2 Inhaltsverzeichnis

1	Einleitung und Inhaltsverzeichnis	1-2
1.1	Copyright	1-2
1.2	Inhaltsverzeichnis	1-3
1.3	Verfügbare Dokumentation	1-4
2	Systembeschreibung	2-1
2.1	Varianten	2-1
2.2	Der Leistungsteil	2-1
2.3	Der Bewegungscontroller.....	2-2
2.4	Peripherie	2-4
2.5	Bestellschlüssel	2-5
2.6	Technische Daten (Ergänzung zum Datenblatt)	2-5
3	Funktionen und wichtige Größen	3-1
3.1	Die drei Funktionsbetriebsarten	3-1
3.2	Positions- oder Drehzahlregelung	3-3
3.3	Positionsinformation, Encoder	3-5
3.4	Referenzierung und Massstab	3-6
3.5	Die Software - Lagegrenzen, Endschalter, externer Stopp	3-7
3.6	Handbetrieb, manuelle Drehzahlregelung.....	3-8
3.7	Drehzahl, Maximaldrehzahl	3-9
3.8	Beschleunigung	3-10
3.9	Regelparameter	3-11
3.10	Grenzwerte für Schlepp- und Lagefehler	3-12
3.11	Grenzwert der Ausgangsspannung, Stromgrenze	3-12
3.12	Auslösen einer Bewegung	3-13
3.13	Endlos - Sprünge, Massstabsverschiebung	3-14
3.14	Triggerpunkt, Auslösen gespeicherter Bewegungen.....	3-15
3.15	Handrad, manuelle Positionsregelung, elektronisches Getriebe.....	3-16
4	Kommunikation	4-1
4.1	Allgemeines zur Kommunikation.....	4-1
4.2	Übertragungsprotokoll	4-1
4.3	Das Telegramm	4-5
4.4	RS232-Schnittstelle	4-6
4.5	RS422-Schnittstelle	4-7
4.6	RS485-Schnittstelle	4-9
4.7	parallele Schnittstelle.....	4-11
4.8	Unterschiede zwischen serieller- und paralleler Schnittstelle.....	4-14
4.9	Adresse im Netzwerk.....	4-14
5	Befehle	5-1
5.1	Aufbau	5-1
5.2	Zusammenfassung der Befehle	5-2
5.3	Detailbeschreibung der Befehle.....	5-3
6	Parameter	6-1
6.1	Die Beschleunigung.....	6-1
6.2	Die Inkremente entlang des Bewegungsweges	6-2
6.3	Die Regelparameter.....	6-2
6.4	Der Kopplungsfaktor.....	6-3
6.5	Die Drehzahl.....	6-3
6.6	Der Ankerstrom.....	6-4
7	Bedienelemente und Leiterplattenanschlüsse	7-1
7.1	Bedienelemente (Potentiometer, DIP - Switches und LED).....	7-1
7.2	Detailbeschreibung der Printanschlüsse.....	7-3
7.3	Digitaleingänge im PC-Betrieb, Funktionen	7-9
8	Externe Beschaltung	8-1
8.1	Eingangsschaltungen und Eingangspegel.....	8-1
8.2	Ausgangsschaltungen und Ausgangspegel.....	8-2
8.3	Externe Belastung der Speisung	8-2
9	Inbetriebnahme	9-1
9.1	Einrichten.....	9-1
9.2	Erste Bewegungen	9-1
10	Anwendungsbeispiele	10-1

10.1	Einachsige Vorschubeinheit.....	10-1
11	Tipps und Tricks.....	11-1
11.1	Häufig gestellte Fragen.....	11-1
12	Anhang.....	12-1
12.1	Was ist "hexadezimal"?	12-1
12.2	Hexadezimale Zahlengerade, negative Hexadezimalzahlen.....	12-2
12.3	Parameter mit verstecktem Dezimalpunkt	12-2
12.4	Grafik: V- und A - Wert, Rampenzeit, Drehzahl und Encodertyp	12-4
12.5	Berechnung der Prüfsumme	12-5
12.6	ASCII-Tabelle und Hexadezimale Ziffern	12-6
12.7	Geschichte bisheriger Versionen	12-7
13	Schlussbemerkungen.....	13-1

1.3 *Verfügbare Dokumentation*

1.3.1 **Datenblatt**

Das Datenblatt ist ein doppelseitiges aufklappbares A4 - Blatt, das alle wichtigen Informationen zum entsprechenden Antriebsregler (MA4Q, MD4Q etc) in komprimierter Form enthält. Das Datenblatt ist auch in elektronischer Form (pdf - Datei) erhältlich.

1.3.2 **Technisches Handbuch**

Das Technische Handbuch beinhaltet die komplette Beschreibung des digitalen Positionsreglers. Mit Hilfe dieses Technischen Handbuches kann der Anwender das MA4Q oder das MD4Q verstehen und einsetzen. Sollten dennoch Unklarheiten oder weitergehende Fragen auftreten, kann unsere Applikationsabteilung angefragt werden. Das technische Handbuch ist in elektronischer Form (pdf - Datei) erhältlich.

1.3.3 **Demo-Software**

Um den Anwender rasch in die Kommunikationstechnik einzuführen - und damit nicht jeder das Rad frisch erfinden muss - steht eine CD-R mit dokumentierten Programmen für PCs (DOS) zur Verfügung. Die Programme werden in Pascal – Quellencode abgegeben und sind modular aufgebaut.

1.3.4 **Homepage**

Unter <http://www.hardmeier-control.ch> finden Sie weitere Informationen zu unseren Produkten.

2 Systembeschreibung

2.1 Varianten

Die Antriebsregler (auch Motor-Regelgeräte oder Motorcontroller genannt) bestehen aus einem Regelungs- und einem Leistungsteil (Endstufe).

Es folgt eine kurze Beschreibung. Ausführlichere Informationen siehe Kap 2.2 (Leistungsteil) und Kap 2.3 (Controller).

Es gibt verschiedene Leistungsteile:

A) für elektrisch kommutierte (EC-, BL-) und B) für bürstenbehaftete (DC-) Motoren.

Je nach Leistungsklasse kann aus verschiedenen Spannungs- und Stromvarianten gewählt werden. Der Antriebsregler sollte immer mindestens so stark wie die Nennleistung des Motors gewählt werden. Die Bauform des Antriebsreglers ist von der Nennleistung abhängig.

Die Controller - Karte bietet hardwaremässig einige Optionen:

- Schnittstellentyp (RS232 oder RS422/RS485)
- Digitalausgangslogik (source oder sink)
- Zweiter Encoder - Eingang (Indexer, Handrad)

Software - Versionen und Erweiterungen:

Softwaremässig befasst sich diese Dokumentation ausschliesslich mit der sog. "Standardsoftware". Diese bietet ebenfalls einige Varianten, die speziell im Kapitel über die Kommunikation (Kap 4) beschrieben werden. Es gibt die Möglichkeit, die Standardsoftware kundenspezifisch zu erweitern.

2.2 Der Leistungsteil

Das Herzstück des Leistungsteils sind die Leistungsschalter (Transistoren), welche die Motorphasen wechselseitig an die positive oder negative Betriebsspannung legen und durch den Mittelwert den Motorstrom und die Motorspannung bestimmen. Aus Strom und Spannung ergibt sich das Drehmoment und die Drehzahl des Antriebes.

Der Leistungsteil wird mit einem Ankerstrom - Sollwert gesteuert, welche vom Controller errechnet wird.

Auf dem Leistungsteil befindet sich zudem die Logik zur Freigabe der Endstufe, die Erfassung des Motorstromes für den Controller und die Energieversorgung für die Controllerkarte.

Je nach Typ sind diverse Überwachungsfunktionen integriert (Geräte - Übertemperatur, Kurzschluss der Motorphasen, Erdschluss der Motorphasen, Überspannung des Zwischenkreises etc).

Die Tabelle auf der ersten Seite des Datenblattes gibt über die verfügbaren Typen und deren Dauerstrom Auskunft. Je nach Endstufentyp kann für wenige Sekunden zusätzlich ein dynamischer "Überstrom" abgegeben werden. Der Code der eingesetzten Endstufe kann abgefragt werden (siehe J-Befehl).

Anschlüsse

- UDC+ Positive Betriebsspannung.
- UDC- Negative Betriebsspannung, ist verbunden mit 0 Volt
- AY, AX Positiver und negativer Anschluss des Motorankers beim MD4Q
- U, V, W Die drei Phasen des Motorankers beim MA4Q
- E (PE) Gehäusemasse, muss extern mit Schutzterde verbunden werden.

Positions-Signale DC-Motor

Bei Geräten für DC - Motoren (MD4Q) werden die Encodersignale direkt bei der Reglerkarte eingespeist.

Positions-Signale EC-Motor

Die Antriebsregler für die elektrisch kommutierten Motoren benötigen zu den Encoder - Signalen für die Positionierung zusätzlich absolute Rotorlage - Signale (sog. RLG- oder HAL - Signale) für die korrekte Kommutierung.

Beim MA4Q erfolgt der Anschluss für die RLG - Signale mittels C32 Stecker auf der Geräterückseite, Die Encodersignale werden (wie beim MD4Q) auf dem C64 Stecker der Reglerkarte angeschlossen.

Beim MA4QC werden alle Signale (Encoder und RLG) mittels 25 pol. Sub-D Stecker auf der Gerätevorderseite angeschlossen.

2.3 Der Bewegungscontroller

Der Bewegungscontroller (Regler) wird mit diversen Informationen versorgt und bedient seinerseits sein Umfeld mit diversen Informationen. (siehe Blockschema im Datenblatt).

Der zentrale Teil des Controllers (Motion - Controller, MC2) ist zweifelsohne der Mikroprozessor mit dem Programm- und dem Datenspeicher. Ein Teil des Datenspeichers wird standardmässig netzausfallsicher in einem NOVRAM (Non Volatile RAM) gespeichert. Darin sind u. a. die Parameter (die Inhalte der Controllerregister) angelegt. Dieser Speicher wird auch bei Ausfall der Speisespannung nicht gelöscht. Der Hersteller dieses Speichers garantiert eine Speicherdauer von zehn Jahren. Optional kann statt dem NOVRAM auch ein normales RAM bestückt werden (Die Parametrierung geht damit bei Speisespannungsunterbruch verloren und muss nach dem einschalten der Speisung neu erfolgen).

Der Quarz: Ist der zentrale Zeitgeber. Daher sind alle zeitabhängigen Grössen wie Motordrehzahl und Fahrzeit quarzgenau.

Drei Potentiometer: Es kann jeweils eine beliebige Grösse eingestellt werden. Dabei kann es sich je nach Software um eine Zeit, eine Motordrehzahl oder ein Drehmoment handeln. Die Standardsoftware ordnet im Handbetrieb Drehzahlen zu (vgl. Kap 3.6).

Der DA - Wandler: Dient dazu, den vom Mikroprozessor ausgerechneten Stellwert (12-Bit) in eine Spannung [+/-10 Volt] umzuwandeln. Es handelt sich hier um die Bildung des Ankerstrom - Sollwertes (Rated I).

Ankerstrom - Istwert: Der Leistungsteil informiert den Controller über den momentanen Istwert des Motorstromes. (Diese Information wird u.a. für die Berechnung der Überstrom - Zeitfläche benötigt).

LED H1 bis H8: Vom Mikroprozessor gesteuerte Leuchtdioden. Sie zeigen in der Standardsoftware den Zustand einiger wichtigen Grössen des Regelsystems an (vgl. Kap. 4.1).

SEPI: Das ist der Schaltkreis (Shaft - Encoder - Peripheral - Interface), welcher die Positionsinformation vom Encoder an den Mikroprozessor weiterleitet.

DIP - Switches S1 bis S8: Eingabeschalter, die vom Mikroprozessor gelesen werden. Die Standardsoftware benutzt einige der DIP - Switches um den Motion - Controller zu konfigurieren (vgl. Kap. 7.1).

RS232 / RS485: Die serielle Schnittstelle zum Motion - Controller. Man kann hardwaremässig zwischen RS232 (Standard) und RS422 (RS485) wählen (vgl. Kap. 4).

Zwei Analogeingänge: Dienen dazu, eine beliebige Grösse (als Spannung) dem Regler einzuspeisen (ähnlich den drei Potentiometer). Der Spannungsbereich ist standardmässig auf 0 bis 5 Volt begrenzt. Die Standardsoftware benutzt diese Analogeingänge (vgl. Kap. 7.2)

Vier potentialfreie, galvanisch getrennte Optokopplereingänge: Es sind 4 Digitaleingänge, die speziell schnell ausgewertet werden. In der Standardsoftware werden damit die Signale der Endschalter, der Vorreferenz und eine allfällige Stoppinformation eingespeist (vgl. Kap. 7.2).

8 NICHT - potentialfreie Digitaleingänge (DIO0 bis DIO7): Die Standardsoftware verwendet diese Digitaleingänge je nach Betriebsart unterschiedlich (vgl. Kap. 3.1, 7.2 und 7.3).

8 NICHT - potentialfreie Digitalausgänge (A00 bis AO7): Die Standardsoftware verwendet diese Ausgänge, um einige Zustände zu signalisieren (vgl. Kap. 7.2).

Wichtig ist, dass diese Ausgänge in positiver und negativer Logik ausgeführt werden können (siehe Abschnitt "Varianten ...").

Reset HWR: Eingang, um den Controller hardwaremässig zurückzustellen (open Collector Eingang). Dieser Pin kann auch als Ausgang zur Signalisierung eines internen Controller erzeugten Reset Signals verwendet werden.

Achtung: Ein "low" - Pegel an diesem Pin führt unmittelbar einen Reset des ganzen Controllers durch, was bedeutet, dass alle aktuellen Informationen wie Istposition, Referenzpunkt, etc verloren gehen. Verwenden Sie diesen Eingang höchstens im Ausnahmefall.

Rated I: Dieser Pin führt das gleiche Stromsollwert - Signal, welches auch der Controller zum Leistungsverstärker (Endstufe) sendet.

Speiseleitungen

0V	Signalbezugspotential 0 Volt.
F0V	"Floating" 0 Volt. Ein 0 Volt Bezugspotential für alle Ein- und Ausgänge.
+5V	5 Volt Ausgang. Für die externen Potentiometer, welche die zwei Analogeingänge speisen.
+15V	Speisung für 15 Volt Encoder, für die Digitaleingänge etc. Mit einer hardwaremässigen Umrüstung kann hier der Controller ab einer externen 24 Volt - Quelle gespeist werden.

Achtung: Aus störungstechnischen Gründen sind die geräteinternen, stabilisierten Spannungen nicht für die Speisung der Encoder geeignet!

Varianten

Die Digitalausgänge sind in zwei Varianten verfügbar:

- **Source-Driver** (PNP - Transistorausgang). Sie ziehen im aktiven Zustand nach "+AO" und sind im passiven Zustand hochohmig. Die Schaltspannung, die im aktiven Zustand an die Digitalausgänge gelegt wird, muss über den Anschluss +AO eingespeist werden. Sie kann gewählt werden: z.B. 15 Volt vom Controller selber, oder 24 Volt von einer externen SPS.
→ Diese Variante ist **Standardbestückung!**
- **Sink-Driver** (NPN – Open – Collector - Transistorausgang). Der Ausgang ist im aktiven Zustand 0 Volt, im passiven Zustand ist er hochohmig.
→ Diese Bestückung muss speziell verlangt werden (vgl. Kap. 2.5)!
Achtung: die Funktion der Digitalausgänge ist in Bezug auf die Spannung gegenüber der Standardbestückung invertiert!

Controller - Funktion

Die prinzipielle Regelfunktion des Reglers besteht darin, über den Ankerstrom den Motor so zu steuern, dass dieser einer bestimmten Position (Lage als Funktion der Zeit) folgt. Die Kennwerte der Position, z.B.: die anzustrebende Zielposition oder die anzustrebende Motordrehzahl, werden in einer geeigneten Art dem Controller übermittelt. Er berechnet daraus ein sog. Fahrprofil, d.h. für alle Millisekunden den Ort, wo der Motor stehen sollte, und steuert nun den Leistungsverstärker so, dass der Motor diesem Profil nachfährt. Da der Controller laufend über die aktuelle Ist - Position informiert ist, kann er aufgrund der Abweichung zur Sollposition entsprechend eingreifen. Es entsteht ein geschlossener Regelkreis, der, richtig eingestellt, ein stabiles und kontrolliertes Verhalten der Motorwelle garantiert.

2.4 Peripherie

Es wird hier kurz auf die weiteren wesentlichen Bausteine des ganzen Antriebs - Systems hingewiesen. Für Ihre speziellen Probleme steht Ihnen unsere Verkaufsabteilung gerne zur Verfügung.

DC Motoren: Es können grundsätzlich alle permanent- und fremderregten Gleichstrommotoren eingesetzt werden. Der Typ richtet sich nach der erforderlichen Leistung und der gewünschten Betriebsspannung. Das Datenblatt und der Bestellschlüssel (Kap. 2.4) gibt über die lieferbaren Typen Auskunft. Die minimal erforderliche Induktivität finden Sie im jeweiligen Datenblatt.

EC Motoren: Normalerweise handelt es sich um block- (oder trapez-) kommutierte bürstenlose DC-Motoren. Der Antriebsregler-Typ richtet sich nach der erforderlichen Leistung und der gewünschten Betriebsspannung. Das Datenblatt und der Bestellschlüssel (Kap. 2.4) gibt über die lieferbaren Antriebsregler Auskunft.

Encoder (Inkrementalgeber, Impulsgeber): Als Rückmeldung der Positionsänderung resp. der Drehzahl kann ein handelsüblicher AB - Encoder verwendet werden (vgl. Kap. 3.3 und 3.4). Er muss zwei um 90 Grad verschobene rechteckige Signale (A und B), die sich pro Umdrehung der Encoderachse eine bestimmte Anzahl mal wiederholen (beispielsweise 1000 Mal), sowie ein Referenzsignal (Z), das pro Umdrehung genau einmal auftritt, aufweisen.

Wir empfehlen als Encoder einen 15-Volt-Line-Driver-Typ. Es kann aber auch ein 5Volt- oder ein 10...30-Volt-Typ eingesetzt werden. Aus EMV - Gründen sollte der Encoder nicht mit der 5V der Reglerkarte versorgt werden.

Die Regelqualität steigt, je harter der Encoder an die Motorachse gekoppelt ist. Üblicherweise sitzt der Encoder deshalb direkt auf der Motorachse und ist mit dieser starr verbunden (ohne Kupplung, ohne Dämpfungselemente etc).

In Ausnahmefällen kann der Encoder irgendwo im System eingebaut werden, um die Position des Antriebs zu melden. Dabei muss beachtet werden, dass alle Getriebefaktoren in die Berechnung des Systems einbezogen werden müssen und dass aufgrund von GetriebeSpielen und Elastizitäten (Kupplungen etc.) regeltechnische Probleme auftreten können.

Tachodynamo: Ein Tachometer ist nicht nötig, da der Controller die Drehzahl aufgrund der Positionsänderung errechnet.

SPS: Als Leitreechner von einem oder mehreren Controllern sind alle handelsüblichen SPS geeignet. Die Kommunikation läuft über die serielle Schnittstelle (vgl. Kap. 4.7). Zu beachten ist der Umstand, dass alle Parameter hexadezimal an den Controller übergeben werden müssen (vgl. Kap. 2.6 und 12.1).

PC: Als Leitreechner von einem oder mehreren Controllern kann statt einer SPS auch ein beliebiger PC dienen, der eine freie RS232 Schnittstelle aufweist. Die Distanz zwischen PC und Controller sollte 15 Meter nicht überschreiten. Um die Übertragungsdistanz auf hunderte von Metern zu vergrößern, kann der PC und der Controller mit einer RS422/RS485 ausgerüstet werden. Mit dieser Option können auch mehrere Controller an einen einzigen PC-Ausgang angeschlossen werden (vgl. Kap. 4.5 und 4.6). Für DOS-PCs steht ein Demoprogramm zur Verfügung (vgl. Kap. 1.3).

Potentiometer: Im Handbetrieb (vgl. Kap. 3.6) können zwei Analogeingänge dazu dienen, die gewünschte Drehzahl vorzugeben. Diese Führungsspannungen können mit einem Potentiometer erzeugt werden. Alle handelsüblichen Potentiometer mit einem Wert von 10 k Ω können dazu eingesetzt werden. Aus EMV - Gründen empfehlen wir, die externen Potentiometer an den 5 Volt des Controllers anzuschliessen (vgl. Kap. 7.1, 7.2 und 8.1).

2.5 Bestellschlüssel

Das Datenblatt gibt in der Tabelle auf der ersten Seite in der Zeile "Typ" die gängigsten Varianten wieder. Die Bezeichnung setzt sich wie folgt zusammen:

MD4Q u / i x

MD4Q ← Gerätetyp (Motorcontroller für **DC**-Motoren mit **4 Q**uadranten betrieb)
 u ← Betriebsnennspannung (Richtwert)(vgl. Kap. 2.2)
 i ← maximaler Dauer - Motorstrom (vgl. Kap. 2.2)
 x ← Hardwarevariante (vgl. Kap 2.3)
 "P" oder nichts : Standardbestückung Digitalausgänge: Source - Driver
 "N" : Spezialbestückung Digitalausgänge: Sink - Driver

Bei Geräten, welche von einem externen Lüfter gekühlt werden müssen (Fremdbelüftung), steht am Schluss der Bezeichnung zusätzlich ein "F" (für "**F**an").

Spezialgeräte (OEM-Lösungen) sind mit einem "Z" bezeichnet.

2.6 Technische Daten (Ergänzung zum Datenblatt)

Regelzyklus: Der Regelzyklus beträgt für den P- und I- Anteil 1024 Mikrosekunden, auf dieser Zeit basieren **alle zeitbezogenen Parameter!**

Ausnahme ist der für die Stabilität ausserordentlich wichtige D - Anteil. Dieser hat einen Rechenzyklus von nur 256 Mikrosekunden. Es kann deshalb von einer Regler - Reaktionszeit von 256 Mikrosekunden gesprochen werden.

Parameter: Die Parameter, welche die Zahlenwerte von Befehlen darstellen, werden in hexadezimal übertragen. Da die Übertragung mit ASCII-Zeichen geschieht, werden die ASCII-Zeichen "0".."9" und "A".."F" benötigt, um die Parameter zu übertragen (vgl. Kap. 6 und 12.1 bis 12.3).

Übertragungsdistanz: Zwischen Leitrechner und Controller können unter Einhaltung der RS232-Spezifikationen 15 Meter überbrückt werden. Entsprechend können es mit RS422 / RS485 bis 1000 Meter sein (vgl. Kap. 4.4.1, 4.5.1, 4.6.1 und 4.7.1).

Stromaufnahme des Reglers: Der Controller (ohne externe Geräte wie Encoder usw.) benötigt (je nach Betriebsfall) ungefähr 50mA von der +15V und ca. 10mA von der -15V Speisung.

3 Funktionen und wichtige Grössen

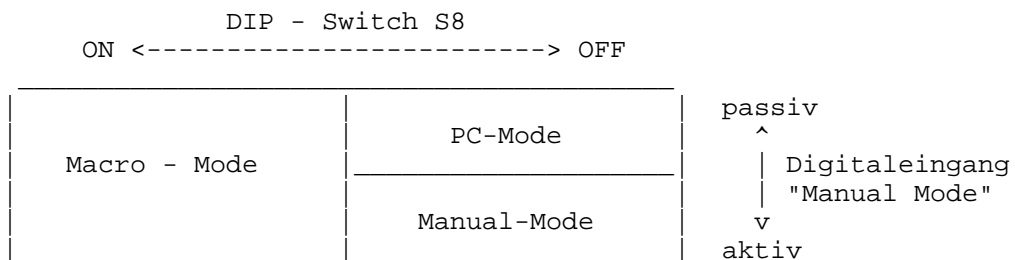
Dieses Kapitel beschreibt die Funktion des Controllers und die damit verbundenen Grössen welche beim Systementwurf beachtet werden müssen.

3.1 Die drei Funktionsbetriebsarten

1. **PC-Mode:** Betrieb über die serielle Schnittstelle, auch **PC-Betrieb** genannt.
Als serielle Schnittstelle steht standardmässig eine RS232, optional eine RS422 / RS485 Schnittstelle zur Verfügung (vgl. Kap. 4.4, 4.5, 4.6 und 4.8).
2. **Manual Mode (Handbetrieb):** Der Antrieb wird über einen analogen Sollwert, Drehzahl geregelt betrieben. Der Sollwert kann intern oder extern vorgegeben werden (siehe Kap. 3.6).
3. **Macro-Mode:** Betrieb mit zusätzlicher (kundenspezifischer) Software (sog. Makros).
In dieser Betriebsart ist eine Zusatzsoftware aktiv, welche den Controller parametrieren und ansteuern kann. Beispielsweise kann man mit dem sog. **SPS-Makro** die 8 digitalen Eingänge als **parallele Schnittstelle** nutzen (siehe Kap. 4.7 und 4.8).

Aktivierung der Funktionsbetriebsarten:

- Indem der DIP - Switch S8 auf OFF (gegen den Leiterplattenrand) geschaltet wird, betreibt man den Controller im **PC-Betrieb** (Manual-Mode Eingang passiv). Die Digitaleingänge haben als Funktion "Run" , "Manual Mode" etc. - vgl. Kap. 7.3)
- Aktivieren des Digitaleinganges "Manual Mode" (DIP - Switch 8 OFF) schaltet in den **Handbetrieb**.
- Indem der DIP - Switch S8 auf ON (Richtung Leiterplattenmitte) geschaltet wird, betreibt man den Controller im **Makromodus (Aktivierung Kundensoftware)**.
ACHTUNG: Die Einstellung des DIP - Switch 8 muss vor dem Einschalten geschehen, da der Zustand des Switches nur beim Initialisieren des Reglers abgefragt wird!
Beachten Sie bitte, dass der Digitaleingang "Manual Mode" im Makromodus eine geänderte Bedeutung haben kann und deshalb anders oder gar nicht funktioniert.

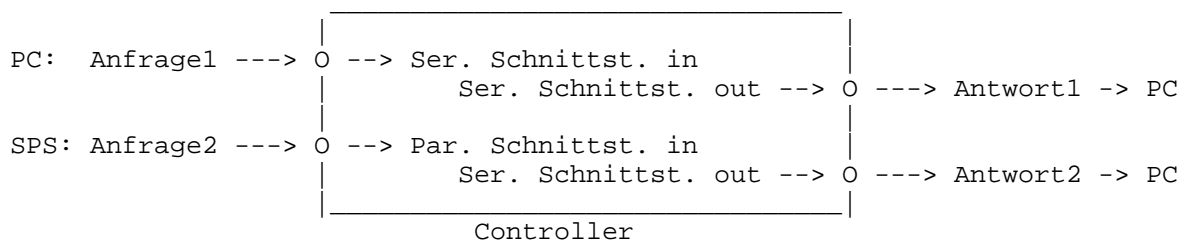


Bemerkungen zum "SPS-Makro" resp. der "parallelen Schnittstelle"

Mit installiertem und aktiviertem "SPS-Makro" kann neben der seriellen Schnittstelle zusätzlich (mittels den 8 digitalen Eingängen) die parallele Schnittstelle zur Übertragung von Befehlen benutzt werden. Weil unter Umständen (auch im Makromodus) die digitalen Ausgänge fest zugeordnete Funktionen haben (um beispielsweise einer SPS den Zustand des Controllers zu signalisieren), werden alle Befehle (von der parallelen- und der seriellen Schnittstelle), die eine Rückmeldung im Telegrammstil erzeugen, über die serielle Schnittstelle beantwortet.

Allgemein wird der Einsatz der seriellen- und der parallelen Schnittstelle während des Betriebs nicht gemischt. Dennoch ist er möglich und kann, speziell zu Testzwecken resp. bei der Inbetriebnahme, wertvolle Dienste leisten.

Ein ganzes Telegramm muss von ein und derselben Quelle kommen, also entweder über die serielle- oder über die parallele Schnittstelle.



ACHTUNG: Im SPS - Makro werden die Digitaleingänge zu "Datenleitungen" resp. "Taktleitung" der synchronen parallelen Schnittstelle. Ihre Datenfunktion ist während des PC-Betriebes nicht aktiv. Aktivieren Sie bitte den Makromodus (DIP - Switch 8 auf ON) im spannungslosen Zustand des Controllers, damit dieser durch das Einschalten richtig initialisiert wird.

Im **PC- und Manual-Mode** stehen folgende Funktionen an den Digitaleingängen zur Verfügung:

- **Run:** Eine passiv - aktiv Flanke ermöglicht erst Drehmoment am Motor - eine aktiv - passiv Flanke deaktiviert ein allfälliges Drehmoment am Motor wieder. Solange Run "aktiv" ist kann mittels M-Befehl (M1, M2 oder M3) eine Reglerbetriebsart aktiviert werden. ACHTUNG: Mit aktivem SPS - Makro wird das Drehmoment nur mit dem Kommando "M" ein- und wieder ausgeschaltet!
- **Manual Mode:** Schaltet zwischen PC-Betrieb und Manual-Mode um (siehe Kap. 3.6 und 7.3)
- **Forward:** Legt im Manual-Mode die Drehrichtung "Uhrzeigersinn" fest. (Vgl. Kap. 3.6 und 7.3)
- **Reverse:** Legt im Manual-Mode die Drehrichtung "Gegenuhrzeigersinn" fest. (Vgl. Kap. 3.6 und 7.3)
- **Slow / Fast:** Wählt im Manualmode den schnelleren oder den langsameren Drehzahlbereich. (Vgl. Kap. 3.6 und 7.3)
- **Home:** Im PC-Modus aktiviert eine passiv - aktiv Flanke eine Bewegung an die Position 0 (Referenzposition). Falls diese Position noch nicht definiert wurde, wird sie mit einem "Z5"-Befehl gesucht. (siehe Kap. 7.3)
Im Manual Mode wird mit diesem Signal die Quelle des Sollwertes (intern oder extern) festgelegt.
- **Go:** Startet eine parametrisierte Bewegung und hat die gleiche Wirkung wie ein "G" - Kommando. Ein Ziel (resp. eine Sprungweite), eine Maximaldrehzahl und eine Rampensteilheit müssen vorher definiert worden sein.
- **Capture:** Eine 0-1-Flanke speichert die aktuelle Position im Register B. Dieses Register kann über die Schnittstelle abgefragt werden. Somit kann man diese Position zur Weiterverarbeitung benutzen.

3.2 Positions- oder Drehzahlregelung

Der Controller kennt verschiedene Regelbetriebsarten:

- Motor ohne Moment (Leerlauf) (M0)
- Positionsbetrieb (M1)
- Drehzahlbetrieb (M2)
- Positionsbetrieb mit retriggerbarem Jump, sog. Fly - Modus (M3)

Die Regelbetriebsart wird durch das Kommando "M" festgelegt und kann jederzeit geändert werden (siehe Kapitel 5.3).

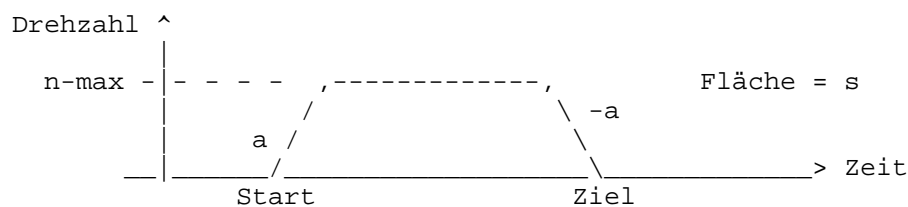
3.2.1 Positionsbetrieb (M1)

Die Lage des Motors oder die Position des Antriebes wird im sog. "Punkt - zu - Punkt Betrieb" geändert. Das typische Fahrprofil (Drehzahl als Funktion der Zeit) ist das Trapez oder als Spezialfall das Dreieck.

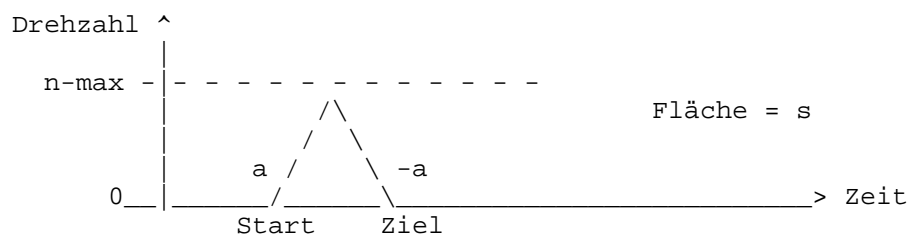
Die Kennwerte einer Bewegung sind:

- Der Weg (s), als Differenz der momentanen Position zur Zielposition.
- Die Maximaldrehzahl ("n max")
- Die Beschleunigung (a) oder Rampensteilheit

Der Motor beschleunigt mit dem Wert "a" bis zur Drehzahl "n max". "n max" wird bis zu einem Punkt beibehalten, wo der Motor mit der negativen Beschleunigung abbremst, um exakt an die Zielposition zu gelangen.



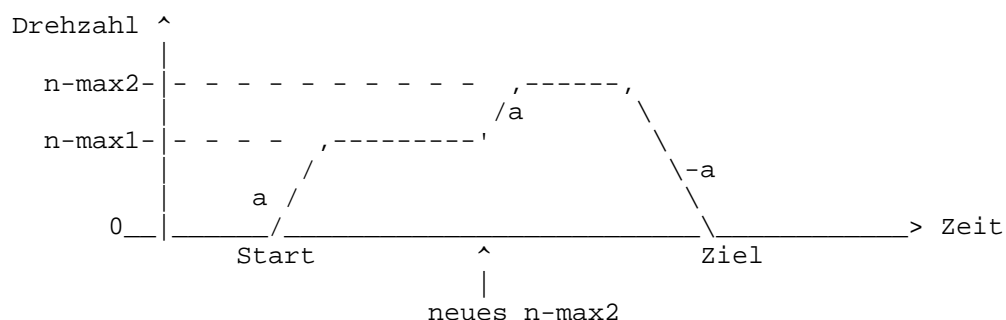
Ist die Fahrstrecke "zu kurz" um die maximale Drehzahl zu erreichen entsteht ein Dreieckprofil:



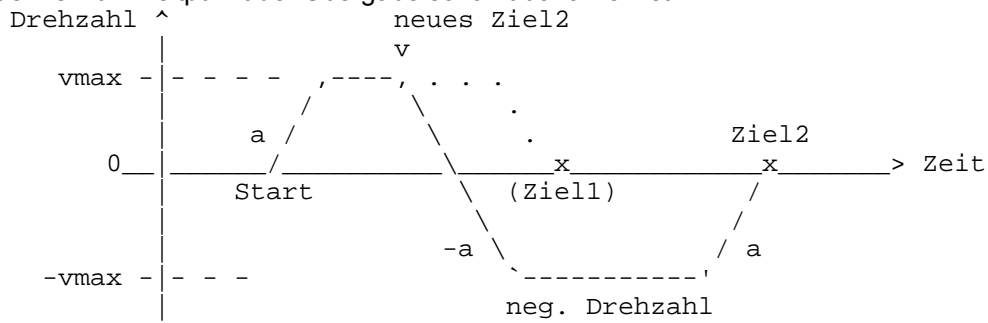
Es ist möglich, nicht nur im Stillstand (also in der Zielposition), sondern auch während der Fahrt, eine neue Zielposition oder eine andere Maximaldrehzahl (oder beides zusammen) vorzugeben.

Die Beschleunigung kann zwar jederzeit verändert werden, der neue Wert wird aber erst nach dem Erreichen der Zielposition übernommen.

Es kann folgendes Drehzahlprofil entstehen:



Falls das neue Ziel zum Zeitpunkt der Übergabe schon überfahren ist:



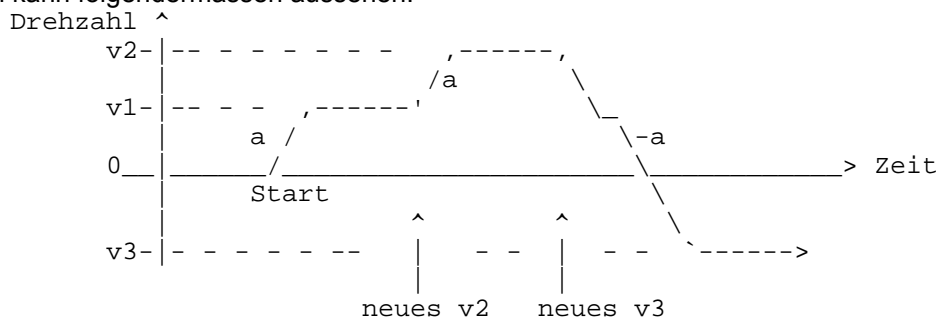
3.2.2 Positionsbetrieb mit retriggerbarem Jump, "Fly - Modus" (M3)

Dieser Modus erlaubt Jumps (Sprünge) während der Bewegung zu verlängern (retriggerbare Jumps). Dürfen Sprünge während der Fahrt nicht verlängert werden, muss der "normale" Positions - Modus M1 gewählt werden: Mit M1 wird ein weiterer Jump erst akzeptiert, wenn die vorangegangene Bewegung abgeschlossen, und das System in Position ist.

3.2.3 Drehzahlbetrieb (M2)

- In diesem Modus wird die Position zwar weiterhin nachgeführt, das Hauptmerkmal ist aber die Drehzahlkontrolle. Eine kommandierte Drehzahl (mit Vorzeichen, im Gegensatz zur Maximaldrehzahl im Positionierungs -Modus) wird sofort entsprechend dem Beschleunigungswert angestrebt und solange beibehalten bis ein neuer Wert kommandiert wird.

Ein Profil kann folgendermassen aussehen:

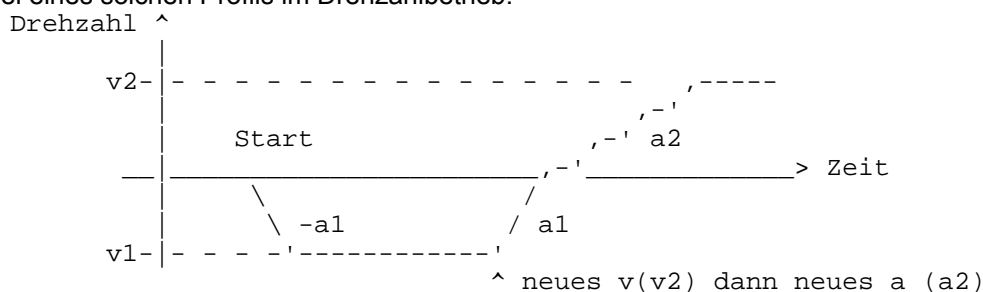


Die Beschleunigung kann im Drehzahlbetrieb jederzeit geändert werden.

Mit einer SW-Version **vor 1.09f** wird der neue Beschleunigungswert **nur im Stillstand** übernommen (d.h. wenn der Antrieb in der Zielposition steht) oder während der Fahrt, wenn die Drehrichtung ändert.

Seit Version 1.09f wird eine neue Beschleunigung aktiviert, sobald eine bereits aktive Beschleunigung abgeschlossen und damit die kommandierte Drehzahl erreicht ist. Damit ist es möglich mit unterschiedlichen Werten zu beschleunigen und zu verzögern.

Ein Beispiel eines solchen Profils im Drehzahlbetrieb:



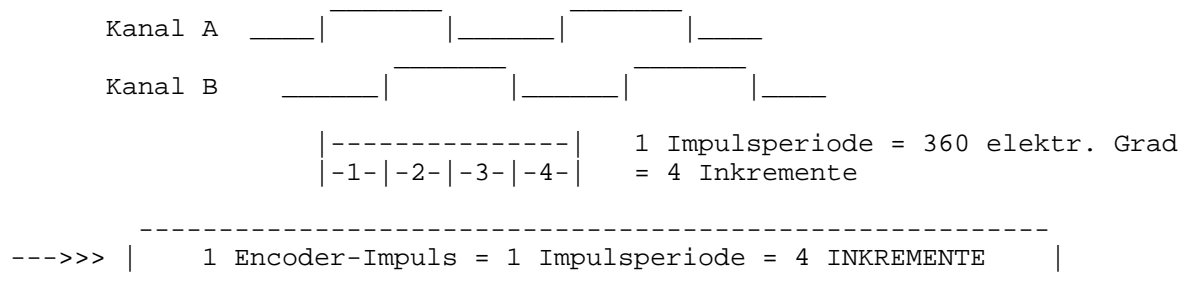
Je nach Anwendung kann ein gemischter Betrieb (verschiedene M - Betriebsarten) sinnvoll sein. Eine programmierte Zielposition geht durch das Umschalten in den Drehzahlbetrieb nicht verloren. Die Zielposition wird im Drehzahlmodus nicht weiter beachtet und kann - zurück im Positionsbetrieb - wieder aktiviert werden.

3.3 Positionsinformation, Encoder

Als Positionsinformation verwendet der Controller die aufsummierten Inkremente vom Drehgeber.

Achtung: Der Controller verliert die Information über die absolute Position, sobald die Speisespannung unterbrochen wird oder ein Reset durchgeführt wird!

Die kleinste Positionsinformation ist das Inkrement. Es ist von den Impulsen des Encoders abgeleitet. Da der Encoder zwei Kanäle mit um 90 Grad versetzten Impulszügen besitzt, kann eine Impulsperiode in 4 Positionsinformationen zerlegt werden:



Davon ausgehend, dass der Controller im Idealfall +/- 1 Inkrement benötigt, um seine Lage zu halten, kann man als Faustregel sagen, dass die maximale Lagegenauigkeit etwa ein Encoder - Impuls beträgt. Bei einem 1000-impulsigen Encoder sind das etwa 0,3 Grad an der Encoder - Achse. Die Positionsgenauigkeit (Anzahl Inkremente) ist von den Regelparametern abhängig.

Es kann vorkommen, dass der Drehsinn des Encoders umgekehrt werden muss. Dies ist z.B. dann der Fall, wenn der Encoder auf der anderen Seite der Motorwelle montiert werden muss. Der Drehsinn kann einfach durch das Tauschen der Kanäle A und B gekehrt werden.

Achtung: Ggf. müssen bei einem DC Motor auch die beiden Ankerleitungen vertauscht werden.

Achtung: Bei einem EC Motor ist das Wechseln des Encoders zum anderen Endes der Antriebswelle meist nicht ohne weiteres möglich, wenn zusätzlich zu den AB-Signalen auch die Rotorlage erfasst wird. Die drei Signale der Rotorlage können nicht ohne weiteres im Drehsinn gekehrt werden, sondern es muss der Drehsinn in der Kommutierungslogik in der Endstufe geändert werden. In einem solchen Fall kann Ihnen unsere Verkaufsabteilung weiterhelfen.

Für die Referenzierung können spezielle Anforderungen an die Signale der Kanäle A, B und Z (resp. A, B und 0) bestehen (vgl. dazu Kap. 3.4).

3.4 Referenzierung und Massstab

Als Positionsinformation verwendet der Controller die aufsummierten Inkremente vom Drehgeber.

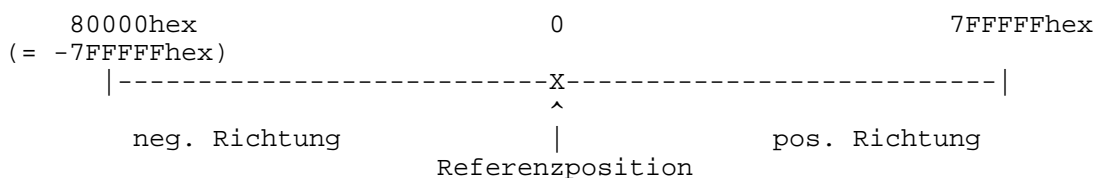
Achtung: Der Controller verliert die Information über die absolute Position und den Referenzpunkt, sobald die Speisespannung unterbrochen wird!

Um einem frisch eingeschalteten Controller mitzuteilen wo sich die Bezugsmarke (Referenzpunkt) befindet, muss ein spezielles Prozedere, die sog. Referenzierung, durchgeführt werden: Das System sucht dabei nach gewissen Merkpunkten und setzt dann die Position auf 0. Üblicherweise wird dies mit dem Z - Signal des Encoders gemacht. Da dieser Impuls aber bei jeder Umdrehung der Motorwelle einmal auftritt und somit meistens mehrdeutig ist, wird zusätzlich mit einer Vorreferenz gearbeitet. Diese Vorreferenz ist ein Schalter irgendwo im Fahrbereich des Systems. Referenziert wird nun auf denjenigen Z - Impuls des Encoders, der (in einer definierten Richtung) am nächsten bei dieser Vorreferenz liegt. Dieser Punkt ist eindeutig und kann immer wieder exakt aufgesucht werden. Beachten Sie, dass der Mindestabstand zwischen Vorreferenz und Z-Impuls eingehalten wird (siehe Z-Befehl im Kapitel 5).

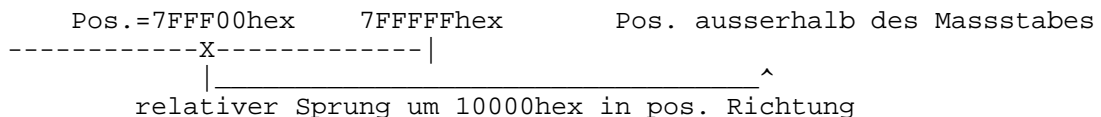
Mittels Kommando "Zz" kann eine Referenzierung jederzeit via Schnittstelle ausgelöst werden. Es sind verschiedene Arten der Referenzierung möglich. (Vgl. Kommando Zz)

Durch den Digitaleingang "Home" kann im Handbetrieb eine Referenzierung ausgelöst werden. (Vgl. Kap. 7.2)

Nach jeder Referenzierung wird der Massstab neu gesetzt. Der Massstab teilt jeder mechanischen Lage im Bezug auf die Referenzposition einen Zahlenwert zu.



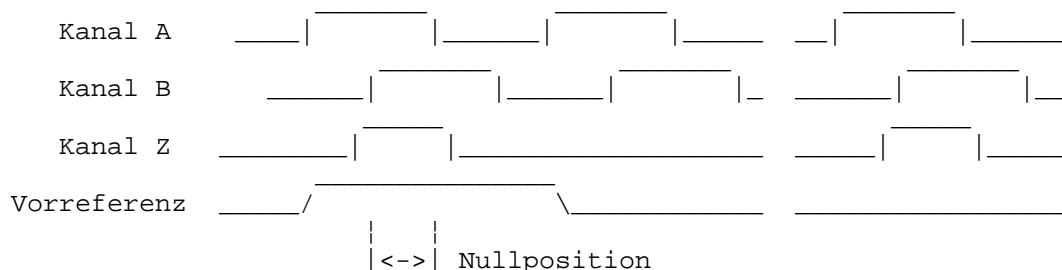
Mit absoluten Zielangaben kann dieser Massstab nicht verlassen werden, hingegen mit relativen Zielangaben und im Drehzahlbetrieb:



Das Flag "REFALT" im Statusbyte gibt Auskunft, ob der Massstab gültig ist oder nicht.

Was bei einer Massstabsübertretung passiert und wie das genutzt werden kann gibt Kap. 3.13 Auskunft.

Damit die Referenzierung zuverlässig durchführen kann, ist es nötig, eine ganz bestimmte Situation auf den drei verwendeten Kanälen des Encoders zu erzeugen. Bei der Standardkonfiguration ist es wichtig, dass einmal pro Geberumdrehung **ALLE SIGNALE GLEICHZEITIG** den logischen Pegel 1 annehmen (*1):



Achtung: Da dies bei gewissen Gebertypen nicht garantiert ist, ist bei der Wahl des Encoders diesbezüglich Vorsicht geboten!

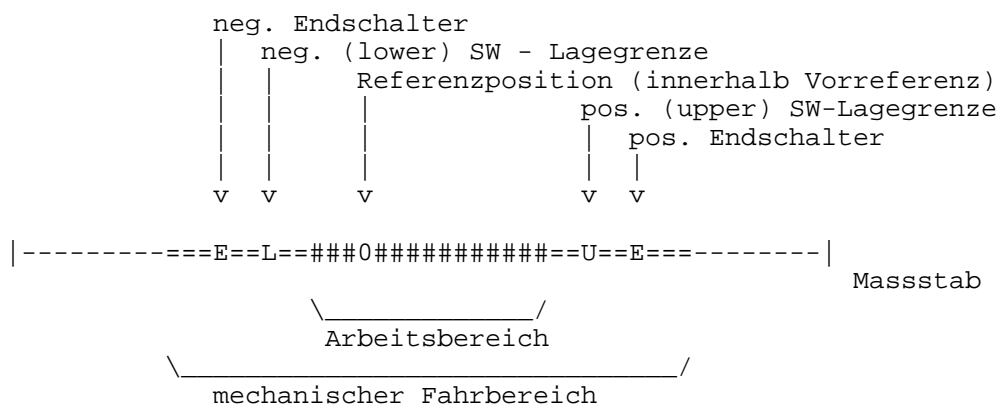
(*1) Die neuen Reglerkarten sind mit einem Encoderimpuls - Zählerbaustein ausgestattet, der die Bedingung "alle drei Signale gleichzeitig logisch 1" nicht mehr stellt. Man erkennt diese Reglerkarten daran, indem auf dem Etikett des Speicherbausteines (EPROM) hinter der Softwarebezeichnung ein "HP" folgt. Beispiel: "mc2 1.09I HP".

3.5 Die Software - Lagegrenzen, Endschalter, externer Stopp

Entlang des Massstabes können zwei Punkte (als Positionen mit dem Kommando "L" und "U" für "lower" und "upper") definiert werden - einen unteren (negativeren) und einen oberen (positiveren) - hinter denen kein Ziel angefahren werden darf bzw. kann. Der Controller überprüft alle Zielpositionen die eingegeben werden und wird im Falle, dass sich eine dieser Zielpositionen "hinter" den Grenzen befindet, ein Fehler - Flag "LIMERR" setzen und den Befehl NICHT ausführen. Diese Grenzen werden Software - Lagegrenzen genannt.

Zusätzlich zu den Softwarelagegrenzen können zwei Endschalter montiert werden. Die Endschalter sind zwei mechanische Schalter die entlang des Fahrbereiches des Systems installiert werden. Der Controller blockiert sofort die entsprechende Richtung falls ein Endschalter überfahren wird und lässt keine weitere Bewegung in diese Richtung zu bis diese Richtungsblockierung mit einem Befehl aufgehoben wird (vgl. Kommando "F" in Kap. 5).

Es wird empfohlen, die Software - Lagegrenzen näher zur Referenzposition hin als die mechanischen Endschalter zu legen, so dass eine doppelte Sicherheit gegen Verlassen des mechanischen Fahrbereiches entsteht:



Wird die untere Lagegrenze "L" positiver als die obere "U" gewählt, ist das System zwischen den beiden Grenzen blockiert.

Es ist möglich, beide Lagegrenzen auf der gleichen Seite der Referenzposition zu definieren. Fahren Sie dazu nach dem Referenzieren in den gültigen Bereich und aktivieren dann die Software-Grenzen mit dem "SWLiEn" - Flag. (siehe Kapitel 5.3 F - Kommando).

Der externe Stopp kann benutzt werden um hardwaremässig das System anzuhalten. Der Stopp kann

- möglichst schnell,
- entlang der Beschleunigungsrampe oder
- durch Auslaufen lassen

geschehen. Die Flag "StMode0" und "StMode1" bestimmen das Verhalten. (Siehe Kommando F im Kap. 5). Weil der Stopp-Eingang auch benutzt werden kann um von extern ein Ziel zu simulieren ("Stopp, hier ist das Ziel"), wird bei aktivem "Stopp" automatisch der Zustand "Ziel erreicht" ausgegeben! Sobald der Stoppeingang deaktiviert und das Drehmoment wiederhergestellt ist kann aus dieser Position - wie aus einer normalen Zielposition - weitergefahren werden.

3.6 Handbetrieb, manuelle Drehzahlregelung

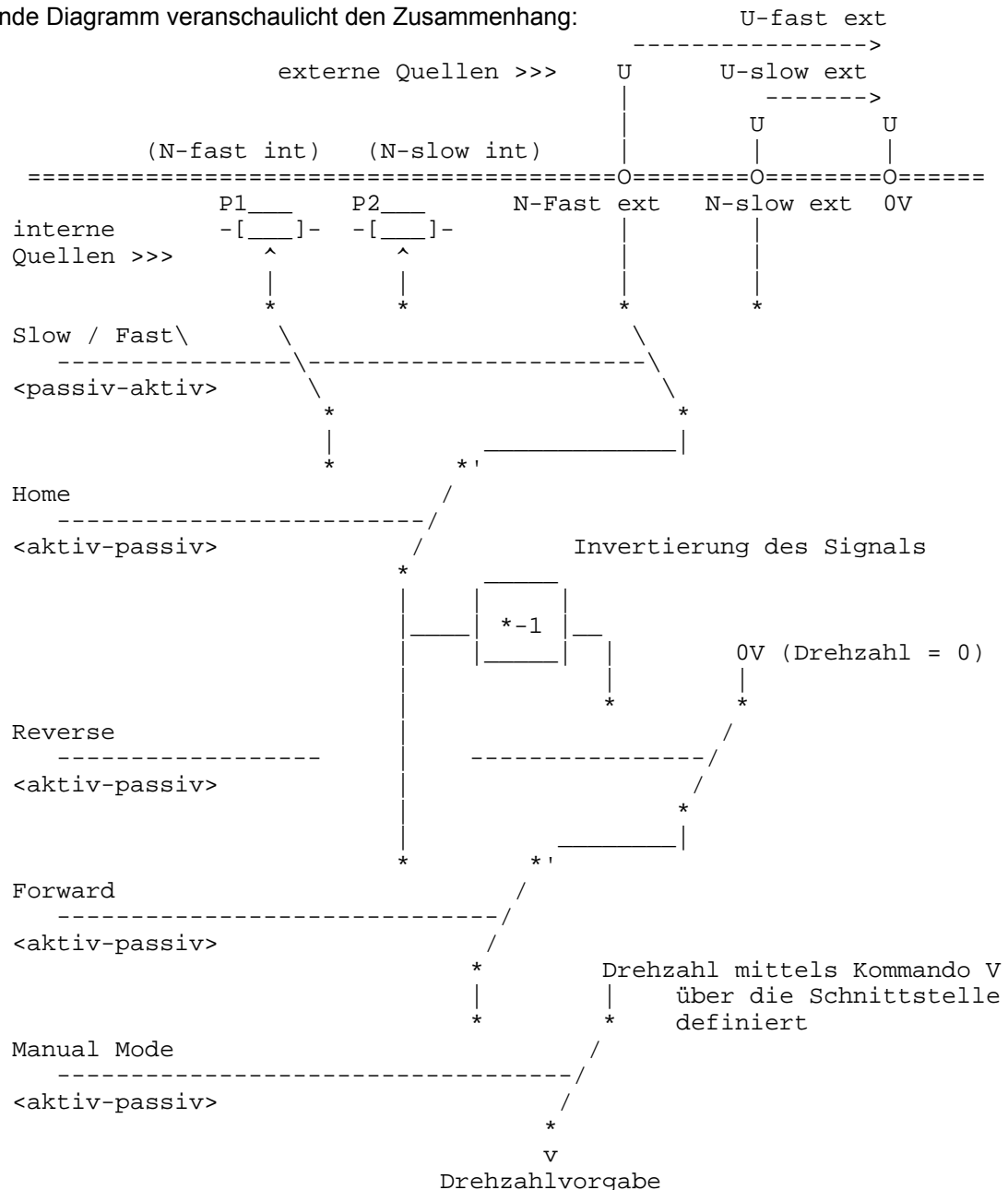
Um den Handbetrieb zu aktivieren muss der Eingang "Manual Mode" aktiv sein (DIP - Switch 8 off).

Handbetrieb ist immer Drehzahlregelung. Der Drehzahlsollwert kann aus vier Quellen ausgewählt werden:

intern:	schnelle Drehzahl:	Potentiometer P1
	langsame Drehzahl:	Potentiometer P2
extern:	schnelle Drehzahl:	Spannung am Eingang N-Fast-ext. (17a)
	langsame Drehzahl:	Spannung am Eingang N-slow-ext. (18a)

Mit dem Digitaleingang "Home" wird der Sollwert zwischen interner (Potentiometer) und externer (Analogeingang) Quelle umgeschaltet. Mit dem Digitaleingang "Slow / Fast" wird entweder die langsame Drehzahl (aktiv) oder die schnelle Drehzahl (passiv) ausgewählt. Die Digitaleingänge "Forward" und "Reverse" bestimmen die Drehrichtung: Beide passiv bedeutet Stillstand, beide aktiv bedeutet vorwärts.

Das folgende Diagramm veranschaulicht den Zusammenhang:



Alle Bewegungsänderungen unterliegen der parametrisierten Beschleunigung (Register A).

Achtung: Verlassen des Handbetriebs während der Fahrt erzeugt einen Schnellstopp (ohne Rampe). Der Antrieb verbleibt dann in Lageregelung (vgl. auch Kap. 7.1, 7.2 und 8.1).

Der Drehzahlbereich beträgt im Falle der schnellen Drehzahl 100%, im Falle der langsamen Drehzahl 6% (1/16) der maximal programmierbaren Drehzahl (vgl. Kap 3.7).

Sobald der "Handbetrieb" durch Deaktivierung des Digitaleinganges "Manual Mode" verlassen wird, ist der Controller im Zustand "Positionsbetrieb".

3.7 Drehzahl, Maximaldrehzahl

Je nach Betriebsart muss eine vorzeichenbehaftete Drehzahl (Drehzahlbetrieb) oder eine vorzeichenlose Maximaldrehzahl (Positionsbetrieb) mitgeteilt werden. Diese Drehzahl wird generell in Inkrementen pro Regelzyklus (Inkr/1024us) angegeben. (Vgl. Kommando "V" in Kap. 5). Die programmierte Drehzahl kann maximal FFhex (=255) Inkremente pro 1024 Mikrosekunden betragen. Das sind ziemlich genau 62 Encoderimpulse pro Millisekunde oder 249'023 Inkremente pro Sekunde (vgl. Kap. 3.3). Die maximale Geschwindigkeit die der Controller verarbeiten kann ist doppelt so hoch.

Je nach Impulszahl des Gebers entstehen entsprechende Maximaldrehzahlen (die maximale Drehzahl ist durch die Frequenz der Encodersignale A resp. B gegeben). Die Werte in der Tabelle sind theoretische Grenzwerte, die effektiven Werte sollten etwas tiefer liegen.

		max. parametrierbare Drehzahl (Kommando V)	max. verarbeitbare Drehzahl (Zählen)
500er Encoder	(2000Ink/U)	7470 U/min	14940 U/min
1000er Encoder	(4000Ink/U)	3735 U/min	7470 U/min
2500er Encoder	(10000Ink/U)	1494 U/min	2988 U/min
5000er Encoder	(20000Ink/U)	747 U/min	1494 U/min

Achtung: Falls die Impulsfrequenz vom Encoder zu hoch ist, ist der Controller nicht mehr imstande die Encodersignale richtig zu verarbeiten! Er ist zudem nicht in der Lage diese Fehlersituation zu erkennen.

- Wenn ein System aufgrund seiner Encoder - Auflösung und seiner maximalen Motor - Betriebsspannung über die maximal verarbeitbare Encoder - Impulsfrequenz gelangt, wird unvorhersehbares Verhalten auftreten! Diese Situation, kann vom Motion - Controller NICHT überwacht und kontrolliert werden!
- Vorzugsweise sollte die max. Frequenz (d.h. Impulse pro Sekunde; nicht Inkremente pro Sekunde) auf max. 100kHz begrenzt werden! Dies kann durch geeignete Wahl von Encoder, Motor und Betriebsspannung erreicht werden.

Der höchste, vom Anwender parametrierte Drehzahlwert (Register V), darf nicht höher als es die Möglichkeiten des Systems zulassen gewählt werden:

- mechanische Maximaldrehzahl des Motors, des Getriebes, der Last
- el. Maximaldrehzahl aufgrund der Versorgungsspannung und der Drehzahlkonstante des Motors.

(Beachten Sie zudem die Spannungsabfälle von Innenwiderstand Motor und von der Endstufe).

Kann die Motordrehzahl dem Fahrprofil nicht folgen wird eine "Schleppfehler" - Meldung "PosErr" (siehe Kommando E) erzeugt (vgl. zudem Kap. 3.10).

Im Normalfall entspricht die maximale Drehzahl höchstens dem parametrierten Wert. In Spezialfällen können aber höhere Drehzahlen auftreten:

- Wenn der Motor von seiner Zielposition genügend weit weggezogen- und losgelassen wird, kehrt er mit maximal möglicher Drehzahl zur Zielposition zurück.
- Wenn der Massstab bei aktiver Endstufe verschoben wird: Endstufe aktiv (Drehmoment ein) und aktives FixPos - Flag (siehe Kommando F) und dann ein Z&zzzzz - Befehl (Vgl. Kap. 3.13), rast der Motor entsprechend der Verschiebung hinterher.

Diese Drehzahl ist von der Drehzahlkonstante des Motors und der Betriebsspannung (sowie dem Drehmoment und dem Innenwiderstand des Motors) abhängig. Sollte einer diese Fälle möglich sein und diese Drehzahl unzulässig hoch werden, muss entweder die Speisespannung verringert, oder eine andere Motorwicklung gewählt werden).

3.8 Beschleunigung

Die Steilheit der Rampe des Fahrprofils (vgl. Kap. 3.2) wird in Inkrementen pro Regelzyklus im Quadrat [$\text{Inkr} / (1024\mu\text{ms})^2$] angegeben (siehe Kommando "A"). Dieser Beschleunigungswert entspricht der Drehzahländerung pro Regelzyklus. Der Beschleunigungswert muss an die Beschleunigungsmöglichkeit des Systems angepasst sein (maximales Drehmoment, wirksames Trägheitsmoment, maximaler Motorstrom etc). Generell darf der Beschleunigungswert nicht höher sein als das System zum Beschleunigen und zum Abbremsen imstande ist, da sonst ein, vom System während der Beschleunigungsphase nicht aufholbarer Positionsfehler erzeugt wird, der am Ende der Beschleunigungsphase zu Überschwingen und Instabilität führen kann.

Probleme kann es beim Abbremsen grosser Trägheitsmomente geben. Die Rotationsenergie die der Mechanik entzogen wird, wird in elektrische Energie umgewandelt (Motor im Generatorbetrieb) und lädt die Zwischenkreiskondensatoren auf. Sind diese Kondensatoren zu klein dimensioniert (Kapazität) steigt die Spannung soweit an, bis die Endstufe wegen drohender Überspannung den Motor in Leerlauf schaltet.

In einem solchen Fall muss entweder:

- die Zwischenkreiskapazität höher gemacht
- die Zwischenkreisspannung mittels Brems - Chopper begrenzt
- die Energie des Zwischenkreises ins Netz zurückgespeist
- der Motor mechanisch abgebremst werden.

Achtung: Weil ein Beschleunigungswert von "0" keinen Sinn macht, kann 0 nicht parametrierbar werden und es wird ein Command Error "CmdErr" erzeugt.

Die in einem System maximal erzielbare Beschleunigung ist vom Lastmoment, vom Trägheitsmoment und vom maximalen Strom abhängig.

Achtung: Beachten Sie bitte, dass besonders bei höheren Drehzahlen der zulässige Strom u. U. nicht erreicht werden kann: Der Strom kann nicht grösser werden als die Spannungsdifferenz zwischen Speise- und Motorspannung (Drehzahlkonstante bei entspr. Drehzahl) dividiert durch den Widerstand.

Im Grenzfall gilt:

$$\text{Speisespannung} > \frac{\text{Innere Spannung des Motors} + \text{Spannungsabfall der Endstufe}}{\frac{\text{Widerstand des Motors}}{\text{maximaler Motorstrom}}}$$

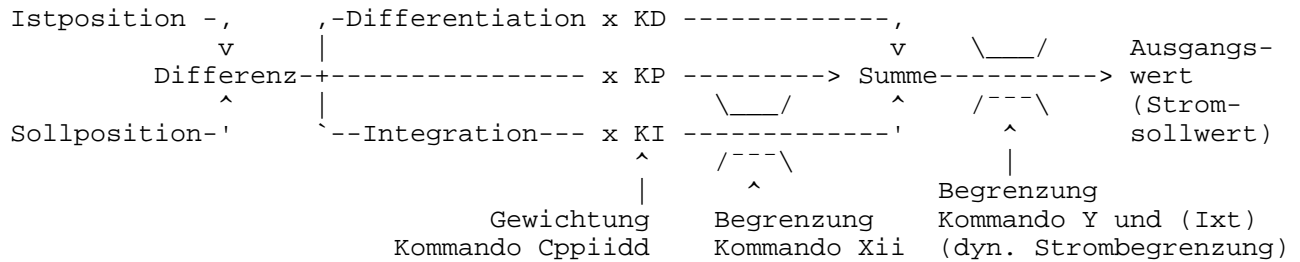
↑ proportional zur Drehzahl
 ↑ je nach Typ: ca. 4V
 ↑ Temperaturabhängig

3.9 Regelparameter

Die Ausgangsgrösse des Controllers (Rated I , Stromsollwert) ist abhängig

- vom momentanen Positionsfehler (Istposition minus Sollposition)
- der Integration (Aufsummierung) dieses Positionsfehlers
- der Ableitung dieses Positionsfehlers.

Jeder dieser drei Summanden wird mit einem dazugehörenden Faktor gewichtet, diese Faktoren werden Regelparameter genannt. Die drei gewichteten Teilsummanden ergeben die Ausgangsgrösse des Controllers:



Der Gewichtungsfaktor zum Positionsfehler wird Proportionalitätsfaktor (KP), derjenige zum integrierten Positionsfehler, Integrationsfaktor (KI) und derjenige zum abgeleiteten Positionsfehler, Differentialfaktor (KD) genannt.

Für jedes System muss der Anwender die optimalen Werte für KP, KI und KD ermitteln. Es gibt neben der empirischen Methode, einige theoretische Verfahren, die aber sehr gute Kenntnis des Antriebssystems voraussetzen. Im Kap. 6 wird etwas näher auf die empirische Methode eingegangen.

Die 3 Werte KP, KI und KD werden mit dem Kommando "C" definiert. Weil diese Werte auch während des Betriebes geändert werden können, ist es denkbar einen adaptiven Regler einzurichten, der je nach Betriebssituation (z.B.: Laständerung) andere Regelparameter verwendet.

Die Integration des Lagefehlers wird begrenzt. Die Grenze kann vom Anwender bestimmt werden (Kommando "X"). Dieser Grenzwert hat ebenfalls eine Auswirkung auf das Regelverhalten des Systems.

Der Controller berechnet die Summanden mit KP und KI alle 1024 Mikrosekunden. Der Summand mit KD wird alle 256 Mikrosekunden berechnet, weshalb von einer Regler - Reaktionszeit von 256 Mikrosekunden gesprochen werden kann.

3.10 Grenzwerte für Schlepp- und Lagefehler

Es existieren zwei unterschiedliche Grenzwerte zur Überwachung des Positionsfehlers:

- **Schleppfehler:** Wird die Differenz zwischen Soll- und Istposition während der Fahrt zu gross, wird entspr. einem Grenzwert (siehe Kommando Xdd) das Fehlerflag "PosErr" gesetzt (siehe Kommando E). Im Drehzahlbetrieb signalisiert PosErr eine Differenz zwischen Solldrehzahl und Istzahl, also einen Drehzahlfehler.
- **Lagefehler:** Ist die Differenz zwischen Soll- und Istposition im Ziel (Fahrt abgeschlossen) innerhalb des entspr. Grenzwertes (siehe Kommando Xss), wird das Zielflag "Target" (siehe Kommando S) gesetzt. Ist die Fahrt abgeschlossen, das Target-Flag bleibt aber nicht gesetzt, muss von einem Lagefehler ausgegangen werden. Ursachen können sein: Zu kleines Drehmoment (z.B. weil Ixt abgelaufen), zu kleiner P- und I-Anteil der Regelparameter (Kommando C) oder zu kleines Zielfenster (Xss). Im Drehzahlbetrieb kann kein Lagefehler auftreten.

3.11 Grenzwert der Ausgangsspannung, Stromgrenze

Die Ausgangsspannung des Controllers wird von der Endstufe als Motorstromsollwert interpretiert (Rated I). Das Kommando "Y" wird benutzt um diese Spannung - und somit indirekt den Motorstrom - zu begrenzen. Innerhalb des Regelgerätes gibt es mehrere Elemente die den Ankerstrom beeinflussen:

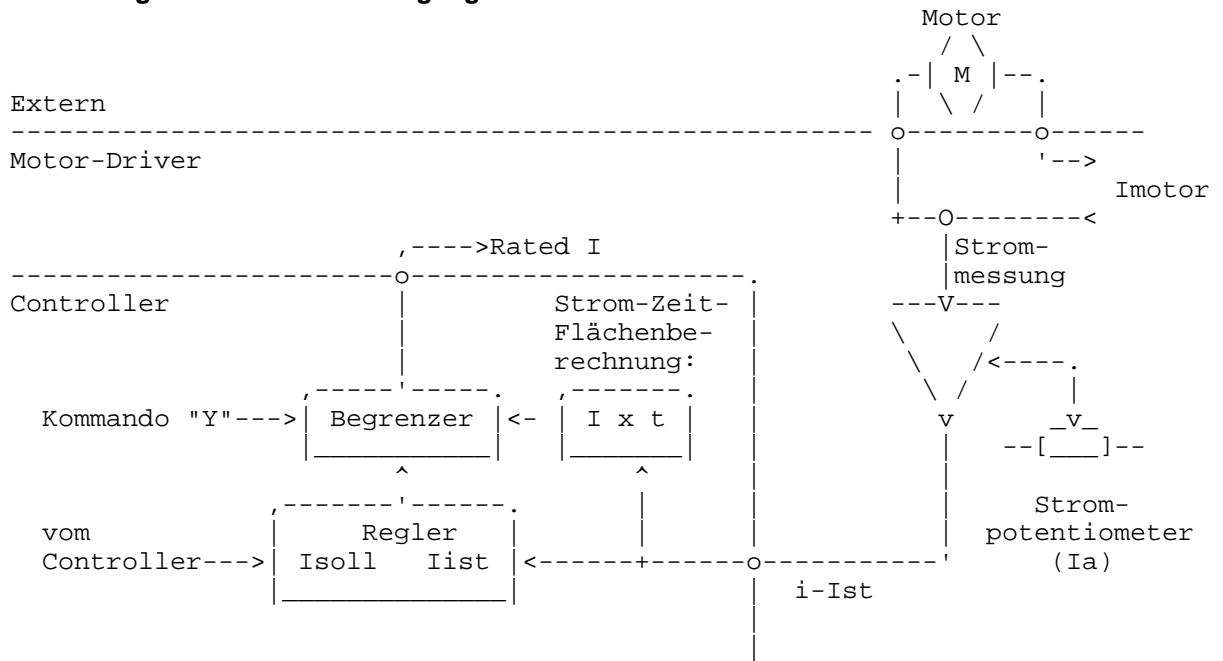
- Die **Endstufe** mit der Strommessung und den Leistungstransistoren bestimmt den maximalen Nennstrom des Gerätes. Dieser Nennstrom schlägt sich in der Gerätebezeichnung nieder (vgl. Kap. 2.2).
- Ein allenfalls vorhandenes Potentiometer auf der Endstufe ermöglicht den effektiven (statischen) Nennstrom abzugleichen (dies wird im Werk gemacht). Für den Controller entspricht dieser Wert einer Strombezugsgrösse von 100%. Ist die Ausgangsspannung des Controllers (Rated I) gleich 5 Volt, versucht die Endstufe diesen statischen Nennstrom einzunehmen. Weil der Controller einen maximalen Ausgangsspannungsbereich von +/-10 Volt aufweist, ist er in der Lage, 200% des statischen Nennstromes zu befehlen - man spricht dann von 100% dynamischem Überstrom. Je nach Endstufentyp darf dieser doppelte Strom nicht beliebig lange beibehalten oder gar nicht kommandiert werden (vgl. Kap 2.2). So haben beispielsweise die MA4Q und MD4Q mit der Bauform Exx eine Zeitüberwachung eingebaut, welche nach einer gewissen Zeit mit Überstrom die Ausgangsspannung (Rated I) auf +/-5 Volt begrenzen. Diese Überstrom - Zeitüberwachung arbeitet nach dem Flächenabtauschverfahren (vgl. Diagramm Flächenabtausch unten).
- Der Maximalwert des Stromes welcher vom Controller kommandiert werden kann wird mit dem Kommando "Y" definiert. Dieser Parameter kann die Ausgangsspannung begrenzen und somit auch den maximalen Motorstrom.

Bei den MA4Q und den MD4Q mit dem Motor - Driver Typ Exx (welche einen Motorstrom von über 100% eine gewisse Zeit abgeben können) kann die Höhe des Stromes begrenzt werden (Befehl "Y"). Die Zeit während der dieser Überstrom zur Verfügung steht wird bei kleineren Strömen entsprechend länger. Die Überstrom - Zeitfläche ist eine Endstufenspezifische Konstante und beträgt bei den Exx - Typen 150% x 1 Sekunde.

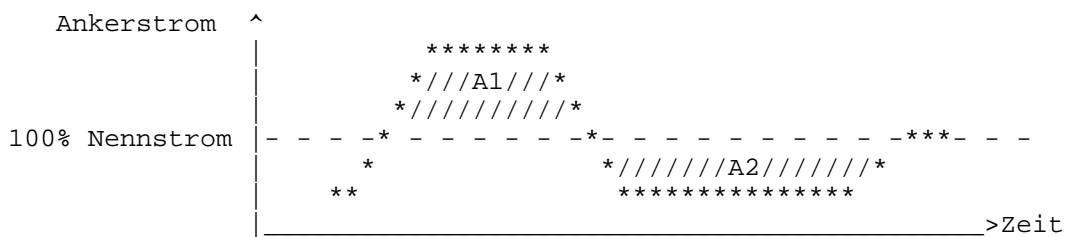
Exx - Typen können also maximal 100% Überstrom während ca. 1,5 Sekunden (Y-Wert = 7Fhex), oder beispielsweise 25% Überstrom während maximal 6 Sekunden (Y-Wert = 4Fhex) abgeben.

Bemerkung: Diese Überstrom - Zeitfläche und das Y - Register werden bei einem Umladen (Initialisieren) des Speichers (ausgelöst durch den Befehl `_ * 3A` oder durch einen Checksummenfehler im Speicher, also bei Einsatz von einem RAM statt einem NOVRAM jedes Mal beim einschalten) mit Defaultwerten initialisiert. Die Initialwerte sind vorgegeben und werden durch den Typ der Endstufe bestimmt. (Siehe Befehl Yyy in Kap. 5.3 und Überstromzeitfläche in Kap. 6.6).

Zusammenhang vom maximalen Ausgangsstrom:



Flächenabtausch:



Die Strom - Zeitfläche oberhalb der 100%-Marke (A1) muss über längere Zeit mit einer Strom – Zeit - Lücke unterhalb (A2) kompensiert werden. Der Mittelwert des Ankerstromes beträgt über längere Zeit maximal 100% Nennstrom.

3.12 Auslösen einer Bewegung

Nachdem eine Fahrt parametrierung wurde (Register A, V, T oder J) und das Drehmoment eingeschaltet ist (M1 oder M3) muss die Bewegung nur noch ausgelöst werden.

Es gibt dazu verschiedene Möglichkeiten:

- Mit dem Befehl "G" (Go)
- Mit einer positiven Flanke am Pin Go. (Pin 22c), (parallele Schnittstelle nicht eingerichtet)
- Beim Überfahren eines aktiven Triggerpunktes (siehe Kap 3.14 und Befehl B)

Bemerkung: Die Kommando "H" und "Z" können auch eine Bewegung auslösen, sind aber spezieller Natur (vgl. Kap. 5.3)

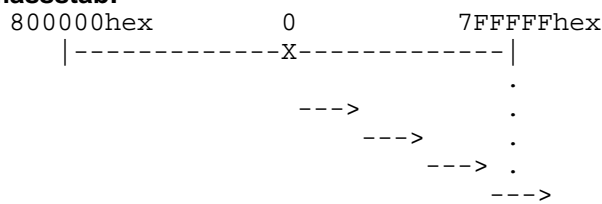
3.13 Endlos - Sprünge, Massstabsverschiebung

In Kap. 3.4 wird beschrieben, dass der Rand des Massstabes mit relativen Zielangaben (Sprünge oder Jumps) überschritten werden kann. In diesem Fall tritt keine Fehlfunktion ein, weil der Controller den Massstab selbständig nachzieht. Endlose Sprünge in gleichbleibender Richtung haben also keine Verluste von Inkrementen zur Folge. Der überlagerte Leitreechner kann die Massstabsverschiebungen mitverfolgen und somit den Wegbereich (von ca. +/- 8 Millionen Inkremente) beliebig erweitern.

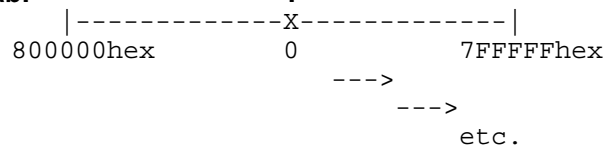
Das Nachziehen des Massstabes geschieht immer um eine halbe Massstabslänge, der überschrittene Grenzpunkt wird neu zur Position Null.

Achtung: Bei einem Ziel unmittelbar vor der Grenze, kann, falls der Antrieb überschwingt, eine Massstabsverschiebung auftreten, obwohl der theoretische Weg die Grenzlage nicht überschritten hat!

Der Originalmassstab:

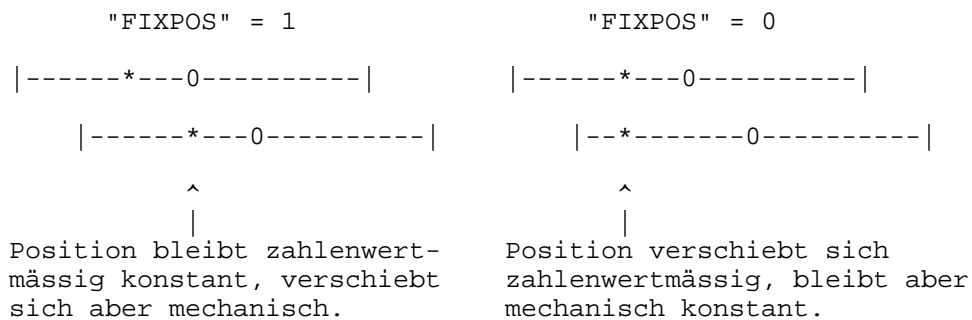


Jetzt der verschobene Massstab:



Das Flag "REFALT" wird gesetzt, sobald der Massstab verschoben wurde und somit die bisherige Referenzposition nicht mehr gilt.

Mit dem Befehl "Z&zzzzz" (vgl. Kap. 5, Kommando "Z") kann der Massstab ebenfalls relativ zur aktuellen Lage verschoben werden. Dabei ist es möglich, mit dem Parameterflag "FixPos" (vgl. Kap. 5, Kommando "F") zu wählen, ob die Position des Systems (Rotor) an den sich verschiebenden Massstab gebunden (Flag=1) oder an die mechanische Position gebunden wird (0). Im ersten Fall führt das System einen Sprung entsprechend der Massstabsverschiebung aus. Im zweiten Fall wird der Zahlenwert der Position verändert.

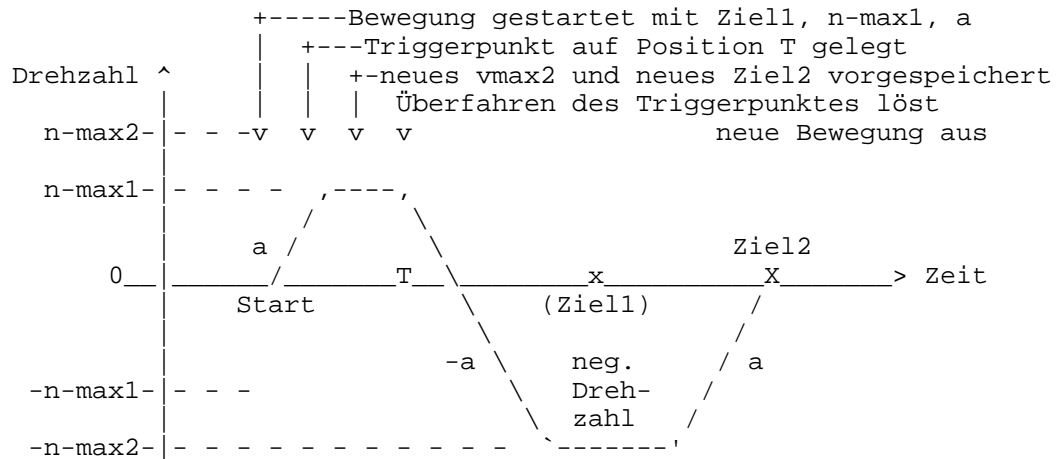


Wann "FixPos" auf 1?

In einer elektrischen Welle kann mit der Verschiebung des Massstabes bei gleichzeitiger mechanischer Verschiebung die Phasenlage zwischen Führungs- und Folgesystem beeinflusst werden.

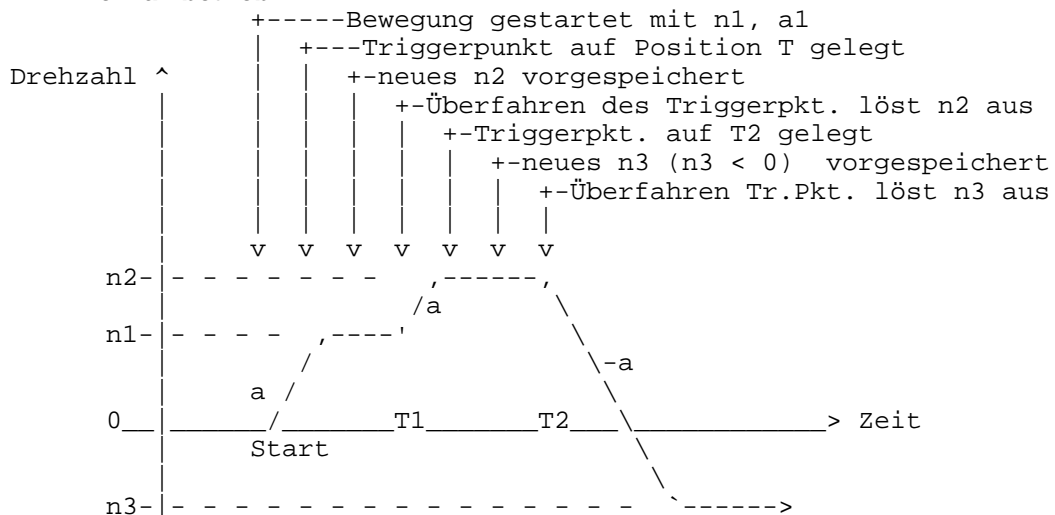
3.14 Triggerpunkt, Auslösen gespeicherter Bewegungen

An einem beliebigen Punkt auf dem Massstab kann ein Triggerpunkt (Brake-Point) gesetzt werden (Befehl B). Sobald der Antrieb diesen Punkt überfährt, wird das Flag "BrPass" (Statusregister S) inaktiv. Wenn der Anwender zuvor das Flag "AXeq" (Register F) gesetzt hat, werden an dieser Stelle die zuvor geladenen Werte für Ziel, Drehzahl (oder max. Drehzahl) und Beschleunigung übernommen, es wird also eine neue Bewegung ausgelöst. Beispiel mit möglichem Fahrprofil:



Der Triggerpunkt kann ebenfalls im Drehzahlbetrieb benutzt werden um eine andere vorzeichenbehaftete Drehzahl und eine andere Beschleunigung zu aktivieren. So ist es möglich, vorausgesetzt die Werte werden genügend rasch nachgeladen, ein fast beliebiges Fahrprofil zu erzeugen, dessen Eckpunkte sehr genau an eine bestimmten Stelle gebunden sind.

Fahrprofil im Drehzahlbetrieb:



(Vgl. Fahrprofil im Drehzahlbetrieb in Kap. 3.2.1 Der Zeitpunkt, wo die Drehzahl ändert, ist dort allerdings von der Geschwindigkeit der Übertragung und ev. auch vom Programm im Leitrechner abhängig!)

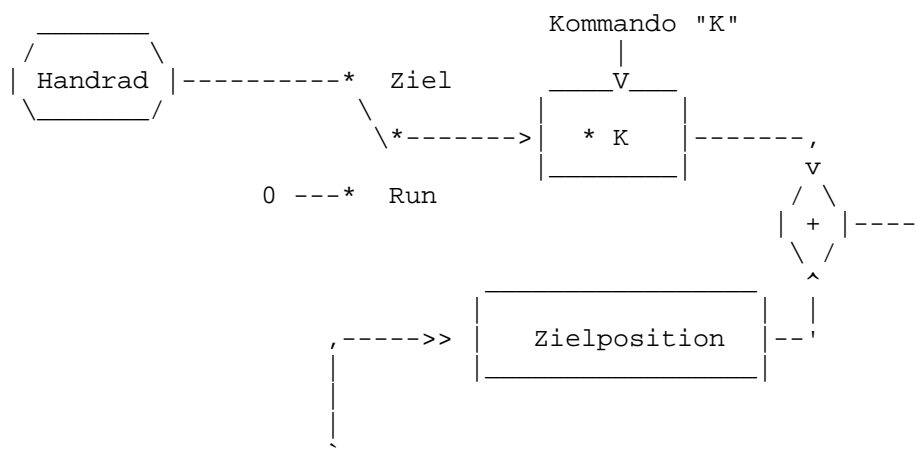
3.15 Handrad, manuelle Positionsregelung, elektronisches Getriebe

Um die Signale des Handrad (Indexer, zweiter Encoder, etc) auswerten zu können muss der Motion - Controller mit der entsprechenden Hardware ausgerüstet sein (Option).

Als Handrad dient ein gewöhnlicher Encoder wie er auch für die Rückmeldung der Drehzahl des Systems verwendet wird. Der Anschluss und die Speisung erfolgen wie beim ersten Encoder.

Im Gegensatz zum Handbetrieb (Manual Mode) ist der Betrieb des Handrades keine eigene Funktionsbetriebsart, denn das Handrad ist immer dann einsetzbar, wenn das System:

- im Positionsbetrieb und in Zielposition ist
- im PC-Betrieb ist
- im SPS - Betrieb ist



Sobald das System in Zielposition ist, wird die Drehinformation am Handrad gelesen, mit einem Faktor multipliziert und die Zielposition um diesen Wert verschoben. Der Faktor kann mit dem Kommando "K" übergeben werden. Wir empfehlen, falls kein Handrad angeschlossen ist, bzw. falls der Controller nicht mit dieser Option ausgerüstet ist, diesen Faktor auf Null zu setzen. Ferner kann der Bereich, in welchem mit dem Handrad verfahren werden kann, mit den Software - Lagegrenzen eingegrenzt werden. Von ausserhalb dieses Bereiches kann zwar hinein gefahren, danach aber dieser Bereich nicht wieder verlassen werden.

Wichtig! Der Indexer ist als Eingang für ein manuell betriebenes Handrad gedacht. Dennoch ist in beschränktem Umfang eine Art elektronisches Getriebe realisierbar. Die maximale Eingangsfrequenz liegt unter 50% der maximal programmierbaren Drehzahl (vgl. Kap. 3.7).

	max. verarbeitbare Indexer - Drehzahl	empfohlene Indexer- Maximal - Drehzahl
250er Encoder	7441 U/min	5580 U/min
500er Encoder	3720 U/min	2790 U/min
1000er Encoder	1860 U/min	1395 U/min
2500er Encoder	744 U/min	558 U/min
5000er Encoder	372 U/min	279 U/min

Es wird empfohlen die maximal verarbeitbare Drehzahl des Indexers höchstens 75% auszunutzen (empfohlene Maximaldrehzahl, rechte Spalte), da beim Überschreiten der zulässigen Maximalfrequenz (124kHz, mittlere Spalte), die Drehrichtung und die Position des Indexers vom Controller falsch ausgewertet werden können!

4 Kommunikation

Dieses Kapitel beschreibt die Kommunikation mit den Controllern. Das Übertragungsprotokoll und der Aufbau des Telegramms werden erläutert. Die verschiedenen Schnittstellen werden beschrieben und es wird auf den Betrieb von mehreren Controllern in einem Netzwerk eingegangen.

4.1 Allgemeines zur Kommunikation

Die Kommunikation dient dem Austausch von Daten zwischen der Steuerung (Leitrechner, PC, SPS, Schaltkontakte etc.) und dem Controller.

Der Begriff "Daten" kann in diesem Zusammenhang folgende grundlegende Arten umfassen:

- Befehle zum Controller (Kommandos, Setzen oder Abfragen von Parametern)
- Antworten vom Controller (als Reaktion auf Abfragen)
- Quittungen (in beiden Richtungen)

Es existieren verschiedene Pfade um Daten auszutauschen:

- Die serielle Schnittstelle vom Typ:
 - * RS232 für eine Einzelverbindung PC <---> Controller oder optional:
 - * RS422 für eine Einzelverbindung PC <---> Controller
 - * RS485 für eine Mehrfachverbindung PC <---> mehrere Controller
- Die parallele Schnittstelle (benötigt das SPS-Makro, eine Zusatzsoftware):
 - * SPS <---> ein oder mehrere Controller
- Die "manuelle Schnittstelle": Digitale Ein- und Ausgänge, um Schalterstellungen einzulesen und Lasten (Lampen, Relais usw.) anzusteuern.

4.2 Übertragungsprotokoll

Die folgenden zwei Kapitel beschreiben die Art und Weise, wie Befehle vom Master zum Controller und Antworten von dort zurück zum Master gelangen. Die Kapitel unterscheiden die Protokolle für seriellen- und parallelen - Betrieb.

4.2.1 serielle Kommunikation

Es herrscht striktes Master - Slave - Prinzip: Jeder Controller (Slave) gibt nur Antwort auf eine Anfrage (des Masters), niemals sendet es von sich aus. Daher ist RS485-Betrieb möglich (Sende- und Empfangsleiterpaar kombiniert). Der Master muss einzig dafür sorgen, dass sein Sender hochohmig wird und auf Empfang geht, nachdem er eine Anfrage an einen Controller gesendet hat. Erst nachdem der Master die Antwort empfangen hat, kann er den nächsten Befehl (an den gleichen oder einen anderen Controller) senden.

Der Master (die Steuerung, der PC etc.) kann jeweils nur mit einem Controller (Slave) kommunizieren. Alle anderen am Netzwerk beteiligten Controller verhalten sich passiv. Nur so ist gewährleistet, dass jeder Controller den Empfang quittieren kann.

Das Timing auf dem Bus sieht wie folgt aus:

Daten zu den Controller: Adr1__Kdo__Kdo__Adr2__Anf_____Kdo__Adr1__Anf_____

Daten vom 1. Controller: ____Q____Q____Q_____Q____Q Ant_

Daten vom 2. Controller: _____Q____Q Ant____Q_____

dabei bedeutet: Kdo: Kommando an den Controller (nur Empfangsquittung, keine Antwort erwartet)
 Anf: Anfrage an den Controller (Master erwartet nach Quittung eine Antwort)
 Q: Empfangsquittung nach einem Befehl oder einer Anfrage
 Ant: Eine Antwort auf eine Anfrage
 Adr1/Adr2: Ein Befehl, der die Adresse des angesprochenen Controllers festsetzt (ändert).
 Die weiteren Befehle gehen an den Controller mit Adresse x.

In vielen Fällen wird ein Master nur mit einem einzigen Controller kommunizieren (z.B. über RS232). Dann sind die obigen Adressierungs-Überlegungen zwar auch gültig, aber eigentlich nicht von Bedeutung. Der Master sendet seine Befehle zum (einzigem) Controller (Slave) und empfängt bei Bedarf die Antworten von dort.

Das Timing auf dem Bus sieht dann wie folgt aus:

Daten zum Controller: __Kdo__ Kdo __Anf__ Anf _____

Daten vom Controller: _____ Q Q Q_Ant Q_Ant _____

Kommandos sind Befehle bestimmender Art. Sie werden nur mit einer Quittung beantwortet. Diese Quittung umfasst ein einziges ASCII-Zeichen und zwar:

- ACK 06hex (acknowledge) = positive Bestätigung = Befehl empfangen
- NAK 15hex (negative acknowledge) = negative Bestätigung = Befehl nicht richtig empfangen

ACK bedeutet, dass der Befehl richtig übertragen wurde und auch gewisse Bedingungen erfüllt (z.B.: keine ungültigen Zeichen usw.).

NAK bedeutet, dass der Prüfsummentest ein falsches und / oder ein falsch übertragenes Zeichen bemerkt hat. Der Befehl muss noch einmal gesendet werden. Es ist zu beachten, dass der Fehler bei einer negativen Quittung nicht unbedingt in der Übertragung liegen muss, er kann auch im Befehl (Aufbau Rahmen), oder in der Berechnung der Prüfsumme) liegen.

Empfängt der Master innerhalb einer gewissen Zeit (typisch: 5 Millisekunden) weder ein ACK noch ein NAK, kann man davon ausgehen, dass die Übertragung unterbrochen ist, der Controller keine Speisung hat oder eine nicht existierende Controller - Adresse angesprochen wurde.

Neben den Kommandos gibt es Parameter-Befehle (setzen oder abfragen von Werten). Sie werden ebenfalls mit einer Quittung nach obigem Muster beantwortet. Falls die Quittung positiv ist (ACK), sendet der Controller zudem die Antwort mit Prüfsumme hinterher.

Diese Antwort kann im Master einer Prüfsummenkontrolle unterzogen werden. Wenn die berechnete Prüfsumme gleich der vom Controller gesendeten ist, ist die Übertragung verifiziert. Ist die Verifizierung positiv, kann (muss aber nicht, da keine Auswirkung) der Master ein ACK - Zeichen an den Controller senden. Ist die Verifizierung negativ, kann der Master ein NAK - Zeichen (negative Quittung) senden, damit der Controller die gleiche Antwort unverzüglich wiederholt. Dies macht aber nur bei statischen Abfragen Sinn. Werden sich ändernde Parameter abgefragt (beispielsweise die Istposition während einer Bewegung) ist es sinnvoller, eine neue Anfrage zu machen.

Beispiel einer Übertragung mit Fehlern:

Daten zum Controller: __Kdo__ Kdo __Anf1__ Anf1 N Anf2__ Anf3 _____

Daten vom Controller: _____ N Q N Q_Ant1__ Ant1 N Q_Ant3 _____

Legende:

- N ist eine negative Quittung
- sonst wie im Beispiel oben.

Die Indizes dienen der Verdeutlichung, welche Antwort zu welcher Anfrage gehört.

Beachten Sie, dass das NAK des Controllers auf die Anfrage2 vom Master unbeachtet blieb und kein zweites Mal übertragen wurde. Anfrage3 kann die gleiche wie Anfrage2 sein. Statt dem alten Wert werden aber die aktuellsten Daten übertragen.

4.2.2 parallele Kommunikation (SPS - Makro)

ACHTUNG: Parallele Kommunikation ist nur mit zusätzlich installiertem und aktiviertem (DIP-Switch 8) SPS-Makro möglich! Da wir festgestellt haben, dass das "SPSMac2" Kommunikationsfehler auf der seriellen Schnittstelle verursachen kann (auch deaktiviert), ist es **ab Version 1.09L nicht mehr installiert!**

Für parallele Kommunikation gelten in der Übertragungsrichtung SPS -> Controller die gleichen Überlegungen wie für seriellen Betrieb (vgl. Kap. 4.2.1). Es handelt sich ebenfalls um ein Master - Slave - System. Der einzige Unterschied besteht darin, dass die ASCII-Zeichen nicht bitweise seriell, sondern ein ganzes ASCII-Zeichen auf einmal über 7 Datenleitungen an den Controller übertragen werden.

In der Übertragungsrichtung Controller -> SPS gibt es aber grosse Unterschiede zur seriellen Kommunikation: Auch in dieser Richtung müssen Informationen übertragen werden, z.B., ob das System die Zielposition erreicht hat. Mit serieller Kommunikation kann das mit dem Lesen des Statusbytes festgestellt werden. Weil die 7 Datenleitungen nur in einer Richtung (schreiben) funktionieren, müssen die acht Digitalausgänge zuhelfe genommen werden:

Funktionen der Digitalausgänge:

- "Ready" aktiv zeigt an, dass der Controller mit Spannung versorgt und bereit ist.
- "In Position" aktiv zeigt an, dass sich das System in Zielposition befindet.
- "Position Error" passiv zeigt an, dass das System einen unzulässig hohen Positionsfehler aufweist.
- "Limit Error" aktiv zeigt an, dass sich das System momentan auf einem Endschalter befindet.
- "Torque" aktiv zeigt an, dass der Motor Moment hat.
- "Trigger" Zeigt Zustand des Triggerpunktes. (siehe Kommando "S", entspr. Flag)
- "TrigDir" Zeigt momentane Richtung der Istposition zum Triggerpunkt an.
- "SPSAck" aktiv zeigt an, dass das soeben parallel übertragene Telegramm mit Hilfe der Prüfsumme getestet und verifiziert wurde.

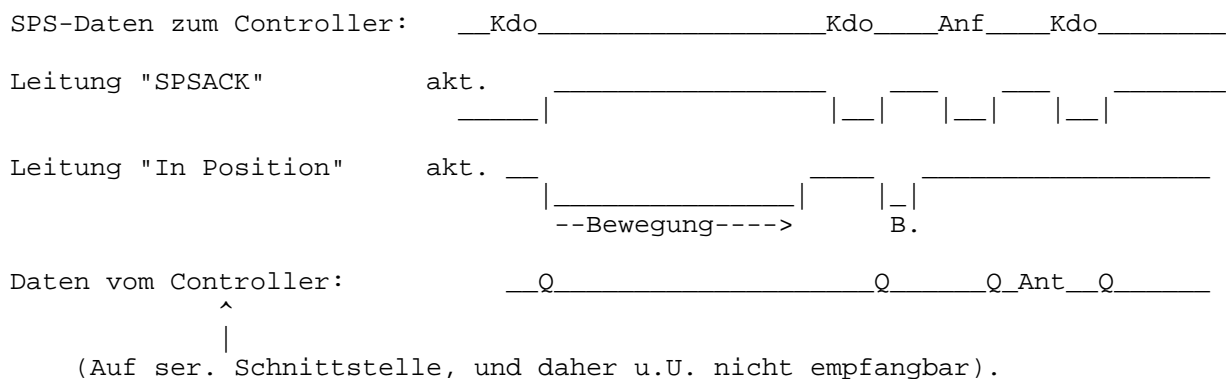
"SPSAck" (=SPS - Acknowledge =SPS - Bestätigung) wird gebraucht, um dem Sender mitzuteilen, ob das Telegramm erfolgreich übertragen wurde oder nicht. Der Sender wird dann entscheiden, ob das nächste Telegramm übertragen werden kann, oder ob das soeben fehlerhaft übertragene Telegramm wiederholt werden muss.

Um die Übertragung eines Telegramms zu verzögern, bis beispielsweise das System in Zielposition ist, muss die SPS das "In Position" - Signal auswerten. So können, je nach Bedarf, auch die anderen Signale herangezogen und ausgewertet werden.

Es ist prinzipiell möglich an den 7 Datenleitungen mehrere Controller anzuschliessen (Netzwerk). Entweder können getrennte Taktleitungen (vgl. Kap. 4.7) eingesetzt werden, oder - ähnlich zum RS485-Betrieb - die Controller können über verschiedene Adressen angesprochen werden. (Vgl. Kap. 4.9)

Die Digitalausgänge des Controllers können ebenfalls zusammengehängt werden. Dies ist allerdings nur sinnvoll, wenn eine ODER - Verknüpfung der Aussagen erzielt werden muss.

Das Timing für parallele Kommunikation sieht wie folgt aus:



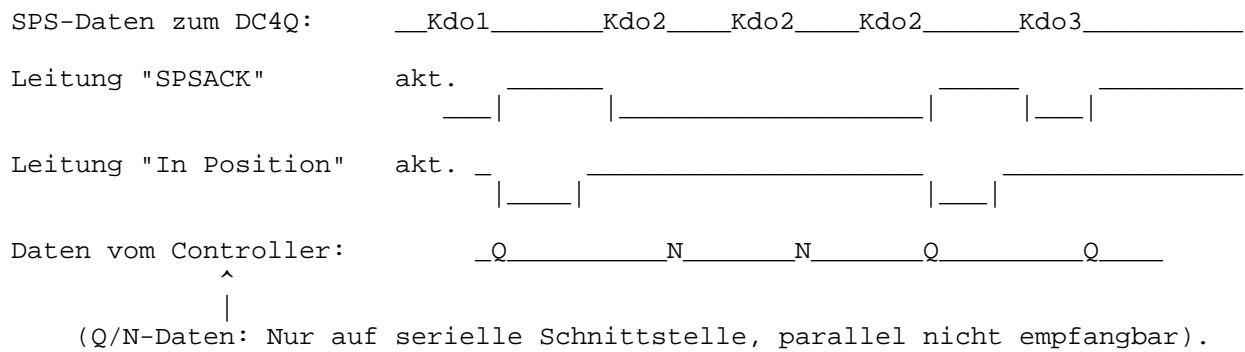
es bedeutet: Kdo: Kommando an den Controller (nur Empfangsquittung, keine Antwort erwartet)
 Anf: Anfrage an den Controller (Master erwartet nach Quittung eine Antwort)
 Q Empfangsquittung nach einem Befehl oder einer Anfrage
 Ant Eine Antwort auf eine Anfrage

Beachten Sie: Die Befehle und Anfragen wurden erst abgeschickt, als sich das System in Zielposition befand.

Ein Telegramm, das aufgrund des Prüfsummentests als fehlerhaft erkannt wurde, erzeugt eine NAK - Quittung (vgl. Kap. 4.2.1).

ACHTUNG: Die entsprechende Leitung für den parallel - Betrieb fehlt: Empfängt die SPS nach dem Senden des letzten Zeichens eines Telegramms innerhalb einer gewissen Zeit (typ. 5 Millisekunden) keinen aktiven Pegel an der "SPSACK" - Leitung, kann man davon ausgehen, dass das soeben gesendete Telegramm fehlerhaft übertragen wurde, die Übertragung unterbrochen ist, der Controller keine Speisung hat oder eine Adresse angesprochen wurde, die nicht existiert.

Beispiel einer Übertragung mit Fehlern:



Bedeutung: N ist eine negative Quittung, sonst wie im Beispiel oben.

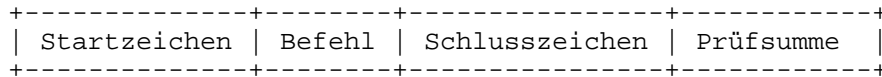
Beachten Sie: Das Kommando2 musste dreimal übertragen werden.

4.3 Das Telegramm

Die Daten müssen nach einer bestimmten Vorschrift übertragen werden. Der Rahmen innerhalb dem die eigentlichen Befehle übertragen werden, heisst Telegramm. Telegramme dieser Art werden im seriellen - Betrieb in beiden Richtungen, im parallelen - Betrieb aber nur in Richtung vom Controller zum Master gebraucht (vgl. Kap. 4.2.1 und 4.2.2).

4.3.1 Allgemeines

Ein Telegramm ist prinzipiell folgendermassen aufgebaut:



Es spielt keine Rolle, ob das Telegramm über die serielle oder die parallele Schnittstelle gesendet wird.

Alle Zeichen in diesem Telegramm sind ASCII-Zeichen, mit Ausnahme des Zeichens, welches die Prüfsumme repräsentiert, es hat einen Wert von über 127.

Die Parameter (innerhalb des Befehles, vgl. Kap. 5 und 6) werden generell HEXADEZIMAL (16er-System, vgl. Kap. 12.1) übertragen. Eine allfällige Umwandlung aus Dezimalziffern (Zehnersystem) muss im überlagerten Leitrechner geschehen.

Für die parallele Schnittstelle in Richtung (Controller -> Master) sind die einzelnen Leitungen festen Aussagen zugeordnet (z.B.: Eine Leitung repräsentiert den Zustand "Antrieb hat Ziel erreicht"), und nehmen entsprechend einen passiven oder aktiven Zustand ein, um die zugeordnete Aussage zu verneinen oder zu bejahen. Diese "Rückleitungen" und ihre zugeordneten Aussagen werden im Kap.7.2 beschrieben.)

Im Betrieb mit der parallelen Schnittstelle, wird nur die positive Quittung übertragen, und zwar mit der Leitung "SPSAck".

4.3.2 Aufbau des Telegramms

Die folgenden Zeichen beziehen sich auf die ASCII-Norm. Auch Zahlen werden ASCII - codiert.

Das Startzeichen ist das Zeichen 02, auch STX oder "Start of TeXt" genannt.

Der Block "Befehl" (vgl. Kap. 5) aus Abschnitt 3 (Bild oben) enthält:

- Ein Kommando: Ein Zeichen, z.B.: "T" (ASCII 84) steht für "set Target" ("setze Zielposition").
- Keine oder eine Kommandoerweiterung, ein Zeichen, z.B.: "-" um nachfolgenden Parameterwert mit negativem Vorzeichen zu versehen.
- Parameter, je nach Kommando: Null bis sechs Zeichen, z.B. "4", "6", "E" und "A". (Beachten Sie die hexadezimalen Ziffern "E" und "A" für 14 und 10).

Das Schlusszeichen ist Zeichen 3 , auch ETX oder "End of TeXt" genannt.

Die Prüfsumme dient dazu, ein falsches oder nicht komplett übertragenes Zeichen (und somit einen fehlerhaften Befehl oder eine fehlerhafte Antwort) zu erkennen. Sie muss für jedes Telegramm neu berechnet werden und wird als letztes Zeichen nach dem Schlusszeichen gesendet. Der empfangende Rechner berechnet dann seine eigene Prüfsumme (nach den gleichen Regeln wie der sendende Controller) aufgrund der empfangenen Daten und vergleicht diese mit der empfangenen Prüfsumme, die ja im Sender berechnet wurde. Sind die berechnete und die empfangene Prüfsumme identisch, ist die Übertragung des Befehles verifiziert. Die Berechnung ist einfach, sie wird im Anhang, Kap. 12.4 beschrieben.

Beispiel: Befehl: "T-1000"

Text :	STX	T	-	1	0	0	0	ETX	Sum
ASCII-Code :	02	54	2D	31	30	30	30	03	FB

4.4 RS232-Schnittstelle

RS232 ist die Standard - Schnittstelle, welche einen Controller mit einem Master (PC o.ä.) verbindet. Die maximal zulässige Übertragungsdistanz sowie die Störverträglichkeit ist beschränkt. Die serielle Schnittstelle kann im Werk optional auf RS422 oder RS485 umgebaut werden.

4.4.1 Eigenschaften und physikalische Daten

Baudrate:	9600 Baud	fix
Übertragene Datenbits:	8 Bit	"
Stop - Bits:	1 Bit	"
Parity - Kontrolle:	keine	"

max. Übertragungsdistanz:	15 Meter (50 feet) nach RS232-Norm
Störanfälligkeit:	hoch, da keine Gleichtaktunterdrückung und nicht galvanisch getrennt.

Protokoll (am PC):	wie in Kap. 4.2.1 beschrieben
Handshake:	nein
Anzahl Controller:	max. 1
Anzahl Leiter:	2 Leiter + Bezugspotential

→ Die Schnittstelle ist nicht galvanisch getrennt!

4.4.2 Anschluss

Üblicherweise genügt ein 3-Leiter-Kabel und bei gewissen PCs müssen einige Verbindungen im Stecker vorgenommen werden (z.B.: RTS und CTS - es handelt sich dann allerdings um ein reines PC-Problem), das PC-Manual sollte darüber Auskunft geben. Zudem sollte das Kabel abgeschirmt sein.

Kabel:	Seite PC		Seite Controller
	AT / XT		
RxD	2	3	<----- Pin 12c TX
TxD	3	2	-----> Pin 13c RX
Sig.GND	5	7	----- Pin 11c 0V
	9 / 25 polig Sub-D, Kabelende weiblich		

4.5 RS422-Schnittstelle

Bei der RS422-Schnittstelle handelt es sich um einen optionalen Ersatz der RS232. RS422 überträgt Gegentaktsignale auf einem Leitungspaar und ist daher für eine Punkt zu Punkt Verbindung geeignet. Dabei können die beiden verbundenen Geräte weit auseinander, oder deren Verbindung in stark von elektrischen Störungen beeinträchtigtem Gebiet liegen.

Der Unterschied zu RS485 besteht nur in der Tatsache, dass an der eigentlich gleichen Schnittstelle bei RS485 mehrere Controller angeschlossen werden können.

4.5.1 Eigenschaften und physikalische Daten

Baudrate:	9600 Baud	fix
Übertragene Datenbits:	8 Bit	"
Stop - Bits:	1 Bit	"
Parity - Kontrolle:	keine	"

max. Übertragungsdistanz: 1000 Meter nach RS422-Norm (mit Leitungsabschlusswiderstand)
 Störanfälligkeit: niedrig, da Differenzsignale übertragen werden; gute Gleichtaktunterdrückung möglich.
 Gleichtaktsignal bis +/- 5 Volt erlaubt

Protokoll (am PC):	wie in Kap. 4.2.1 beschrieben
Handshake:	nein
Anzahl Controller:	max. 1
Anzahl Leiter:	<ul style="list-style-type: none"> • Falls der Sender des Masters in den Sendepausen hochohmig gesteuert werden kann: 2 Leiter plus Bezugspotential. • Falls der Sender des Masters nicht hochohmig gesteuert werden kann: 4 Leiter plus Bezugspotential.

→ Die Schnittstelle ist nicht galvanisch getrennt!

→ Der Leitungsabschlusswiderstand ist eine übertragungstechnische Massnahme um Reflexionen der Signale an den beiden Leitungsenden zu vermeiden. Es handelt sich um einen ohmschen Widerstand, der an den beiden Leitungsenden angeschlossen wird. Sein Wert sollte möglichst der Leitungsimpedanz entsprechen und ist somit von der verwendeten Leitung abhängig. Bei kürzeren Verbindungen (einige 10 Meter) kann darauf verzichtet werden.

ACHTUNG: Weil sich die Abschlusswiderstände nur an den beiden Leitungsenden befinden sollten, unser Controller aber mehrmals vorhanden sein kann und nicht zwingend an einem Ende ist, haben die Controller keine Abschlusswiderstände eingebaut.

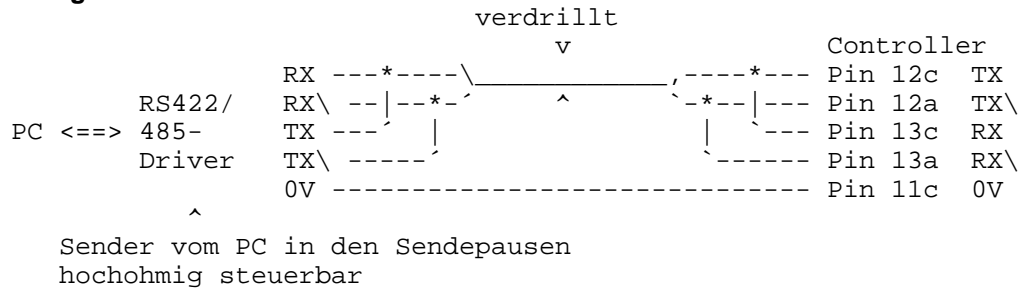
4.5.2 Anschluss

Es genügt ein 3 Leiter - Kabel, falls der Master seinen Sender in den Sendepausen hochohmig steuert. Dann kann die Leitung für Senden und die für Empfangen verbunden werden.

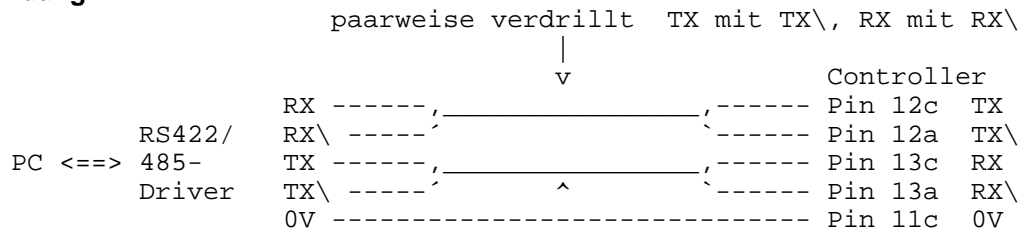
Falls der Sender des Masters aber nicht hochohmig steuerbar ist, müssen die Sende- und Empfangsleitungen getrennt zum Master geführt werden.

Die Spezifikation für RS422/RS485 sieht verdrehte ungeschirmte Leiterpaare vor. Es kann u. U. auch mit abgeschirmten unverdrillten Leitungen gearbeitet werden.

2-Draht-Verbindung

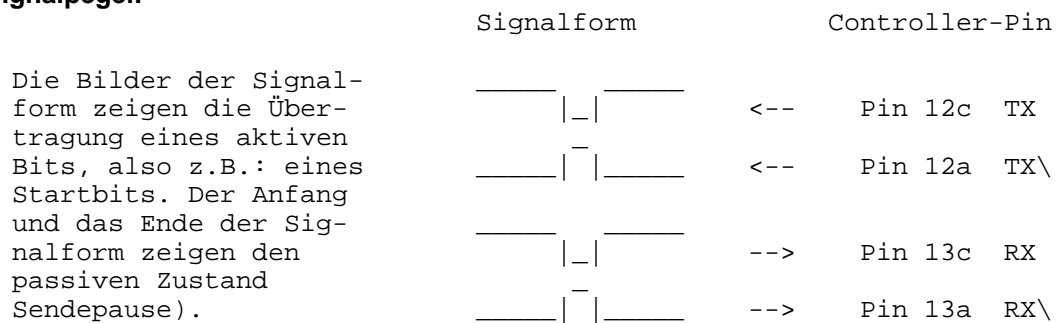


4-Draht-Verbindung



Bemerkung: Die 4-Draht-Verbindung hat den Vorteil, dass einfache Leitungs - Zwischenverstärker benutzt werden können.

logische Signalpegel:



4.6 RS485-Schnittstelle

Bei der RS485-Schnittstelle handelt es sich um einen optionalen Ersatz der RS232.

Mit RS485 werden Gegentaktsignale auf einem Leitungspaar übertragen. Diese Schnittstelle ist für ein Netzwerk (d.h. mehr als ein Controller) geeignet. Dabei können der Master (der PC o.ä.) und die Controller weit auseinander liegen, oder deren Verbindung in stark von elektrischen Störungen beeinträchtigtem Gebiet liegen.

Der Unterschied zu RS422 besteht in der Spezifikation, dass an der eigentlich gleichen Schnittstelle bei RS422 nur ein Controller (Punkt zu Punkt Verbindung) angeschlossen werden kann.

4.6.1 Eigenschaften und physikalische Daten

Baudrate:	9600 Baud	fix
Übertragene Datenbits:	8 Bit	"
Stop - Bits:	1 Bit	"
Parity - Kontrolle:	keine	"

max. Übertragungsdistanz: 1000 Meter nach RS485-Norm für die entferntesten Stationen.
(mit Leitungsabschlusswiderstand)

Störanfälligkeit: niedrig, da Differenzsignale übertragen werden; gute Gleichtaktunterdrückung möglich.
Gleichtaktsignal bis +/- 5 Volt erlaubt

Protokoll (am PC): wie in Kap. 4.2.1 beschrieben

Handshake: nein

Anzahl Controller: Die Controller - Standardsoftware vermag maximal 8 Geräte zu unterscheiden.
(Vgl. Kap. 7.1)

Anzahl Leiter:

- Falls der Sender des Masters in den Sendepausen hochohmig gesteuert werden kann: 2 Leiter plus Bezugspotential.
- Falls der Sender des Masters nicht hochohmig gesteuert werden kann: 4 Leiter plus Bezugspotential.

→ Die Schnittstelle ist nicht galvanisch getrennt!

→ Der Leitungsabschlusswiderstand ist eine übertragungstechnische Massnahme um Reflexionen der Signale an den beiden Leitungsenden zu vermeiden. Es handelt sich um einen ohmschen Widerstand, der an den beiden Leitungsenden angeschlossen wird. Sein Wert sollte möglichst der Leitungsimpedanz entsprechen und ist somit von der verwendeten Leitung abhängig. Bei kürzeren Verbindungen (einige 10 Meter) kann darauf verzichtet werden.

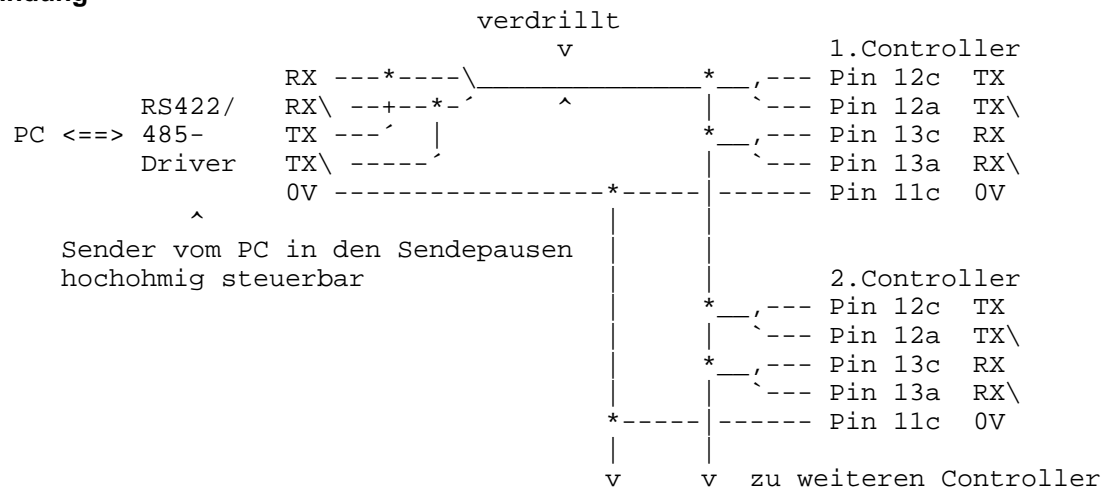
ACHTUNG: Weil sich die Abschlusswiderstände nur an den beiden Leitungsenden befinden sollten, unser Controller aber mehrmals vorhanden sein kann und nicht zwingend an einem Ende ist, haben die Controller keine Abschlusswiderstände eingebaut.

4.6.2 Anschluss

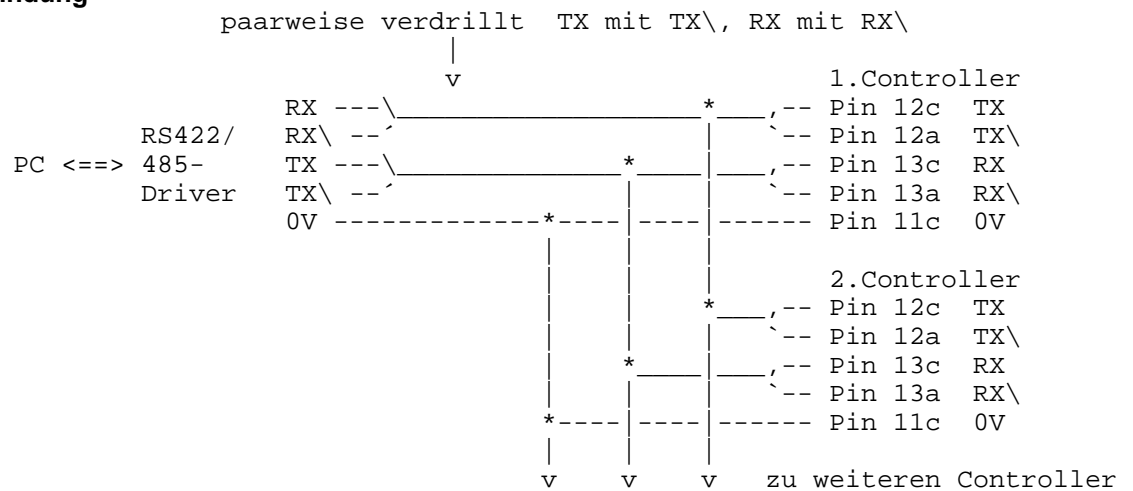
Üblicherweise genügt ein 3 Leiter - Kabel, falls der Master seinen Sender in den Sendepausen hochohmig steuern kann. Dann kann mit einem Leitungspaar für Senden und Empfangen gearbeitet werden. Falls der Sender des Masters aber nicht hochohmig steuerbar ist, dann müssen die Sende- und Empfangsleitungen getrennt zum Master geführt werden. Da PCs üblicherweise keine direkten Anschlüsse für RS422/RS485 besitzen, muss ein Schnittstellenwandler von RS232 in RS422/RS485 unmittelbar beim PC installiert werden, auch sind Einsteckkarten für diese Aufgabe erhältlich. Die Spezifikation für RS422/RS485 sieht verdrehte ungeschirmte Leiterpaare vor. Es kann eventuell auch mit abgeschirmten unverdrillten Leitungen gearbeitet werden. (Für die logischen Signalpegel vgl. Kap. 4.5.2 unten)

Die verschiedenen Controller werden mit Stichleitungen an die Übertragungsleitung angeschlossen.

2-Draht-Verbindung



4-Draht-Verbindung



Bemerkung: Die 4-Draht-Verbindung hat den Vorteil, dass einfache Leitungs - Zwischenverstärker benutzt werden können.

4.7 parallele Schnittstelle

Die parallele (oder SPS -) Schnittstelle ist die standardmässige Verbindung zu einem Master, welcher im Parallelbetrieb mit einem Controller kommuniziert. Genau genommen handelt es sich bei der eigentlichen parallelen Schnittstelle um einerseits sieben Datenleitungen und der dazugehörigen Taktleitung und andererseits um einzelne Rückleitungen mit fest zugeordneten Aussagen (vgl. Kap. 4.2.2).

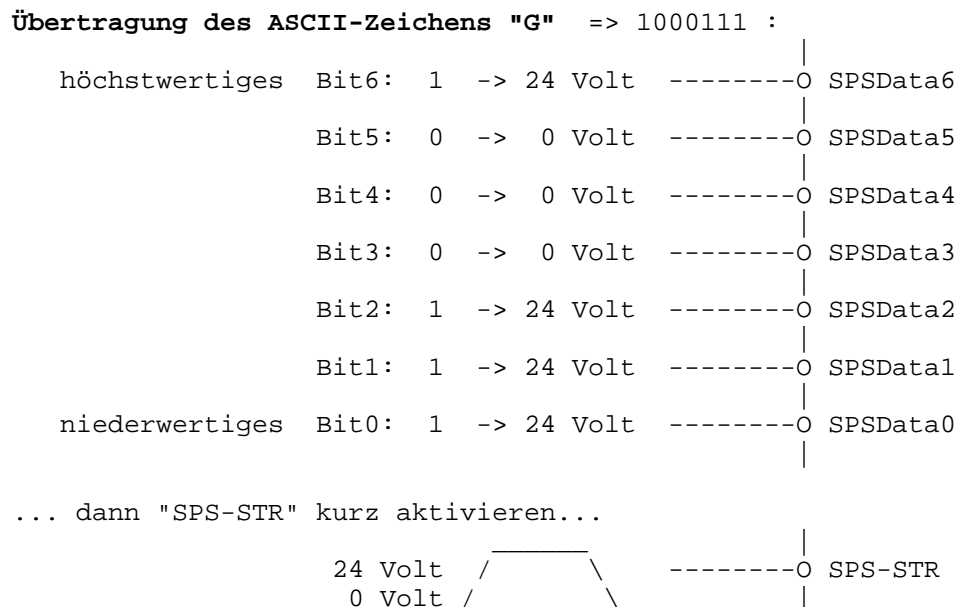
ACHTUNG: Die parallele Schnittstelle steht nur mit dem "SPS-Makro" zur Verfügung. Detail siehe Kap. 4.2.2

4.7.1 Eigenschaften und physikalische Daten

Baudrate:	min ca. 2,4 Millisekunden pro ASCII-Zeichen. Entspricht ca. 4200 Baud
Übertragene Datenbits:	7 Bit parallel, d.h.: gleichzeitig (ähnlich Centronics - Schnittstelle)
max. Übertragungsdistanz:	einige hundert Meter.
Störanfälligkeit:	niedrig, da hohe Signalspannung und langsame Signaländerungen
Protokoll (am PC):	wie in Kap. 4.2.2 beschrieben
Handshake:	ja
Anzahl Controller:	Die Controller - Standardsoftware erlaubt maximal 8 Controller zu unterscheiden (vgl. Kap. 7.1).

→ ACHTUNG: Die Schnittstelle ist nicht galvanisch getrennt!

Die exakt gleichen ASCII-Zeichen, die über die serielle Schnittstelle gesendet würden, werden nun über die parallele Schnittstelle an den Controller gesendet, indem man das Bitmuster des ASCII-Zeichens an die Datenleitungen anlegt (Bitwert 1 = aktiv) und dann die Taktleitung "SPSSTR" (=SPS - Strobe - Leitung) kurz aktiviert. Entscheidend ist die passiv - aktiv Flanke der Taktleitung, diese schiebt das angelegte ASCII-Zeichen gewissermassen in den Controller.



Beachten Sie bitte, dass nur 7 Datenleitungen zur Verfügung stehen. Das reicht jedoch aus, um alle verwendeten ASCII-Zeichen zu übertragen. Alle verwendeten ASCII-Zeichen haben als Bit7 eine 0, welches nicht übertragen wird. Vom Prüfsummenzeichen werden ebenfalls nur Bit 0..6 übertragen.

Nachdem ein ganzes Telegramm übertragen wurde, setzt der Controller die Leitung "SPSAck" aktiv, falls die Übertragung verifiziert werden konnte. Sobald das erste ASCII-Zeichen des folgenden Telegramms beim Controller eintrifft (das STX - Zeichen), setzt der Controller die Leitung "SPSAck" wieder auf passiv (vgl. Kap 4.3).

Ein Telegramm, mit dem Befehl, eine Zielposition an der Stelle -46EAhex zu definieren, wird wie folgt übertragen:

Unser Beispiel von Abschnitt 2: (1): 7 Datenleitungen, (2): Taktleitung, (3): Rückmeldeleitung

(1): $_ | \overline{STX} | _ | \overline{T} | _ | \overline{-} | _ | \overline{4} | _ | \overline{6} | _ | \overline{E} | _ | \overline{A} | _ | \overline{ETX} | _ | \overline{PS*} | _ | \overline{STX} | _$

(2): $_ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _ | \overline{-} | _$

(3): $_ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _ | _$

Die Übertragung geschieht zeichenweise vom Master zum Controller. Das Startzeichen (7 Bit) wird über 7 Leitungen an den Controller angelegt und die Taktleitung kurz aktiviert. Die passiv - aktiv Flanke teilt dem Controller mit, dass dieses Zeichen gelesen werden kann. Danach wird die Taktleitung wieder passiv. Dann wird das Kommandozeichen an den 7 Datenleitungen angelegt und die Taktleitung erneut aktiviert, der Controller weiss nun, dass er das Kommandozeichen lesen kann. Auf diese Art werden alle Zeichen eines Telegramms (inkl. des Prüfsummenzeichens) übertragen, jeder Übergang der Taktleitung von passiv nach aktiv "schiebt" gewissermassen ein Zeichen in den Controller. Sobald er das Prüfsummenzeichen gelesen hat, also ein komplettes Telegramm mit einem Befehl darin empfangen hat, macht er die Prüfsummenkontrolle und setzt seinerseits die SPSAck - Leitung, falls die Prüfsumme korrekt war. Dies signalisiert dem Master, dass der Befehl fehlerfrei angekommen ist und nun eine entsprechende Aktion im Controller ausgelöst wird. Sobald das nächste Startzeichen eingelesen wird, wird die SPSAck - Leitung wieder passiv.

→ Da nur 7 Datenleitungen vorhanden sind, können auch vom Prüfsummenzeichen nur die Bits 0..6 übertragen werden, in unserem Beispiel also das Zeichen 7C hexadezimal.

Die anderen Leitungen vom Controller zur SPS sind in Kap. 4.2.2 beschrieben.

4.7.2 Anschluss

Wie in Kapitel 4.2.2 beschrieben, werden die Befehle vom Master zum Controller über die parallele Schnittstelle gesendet. Informationen vom Controller zur SPS werden über Einzelleitungen mit fest zugeordneten Aussagen übertragen. Im folgenden Anschlussbeispiel werden alle diese Leitungen an den Master (eine SPS) angeschlossen. Je nach Bedarf kann aber auf vereinzelt Rückmeldeleitungen verzichtet werden.

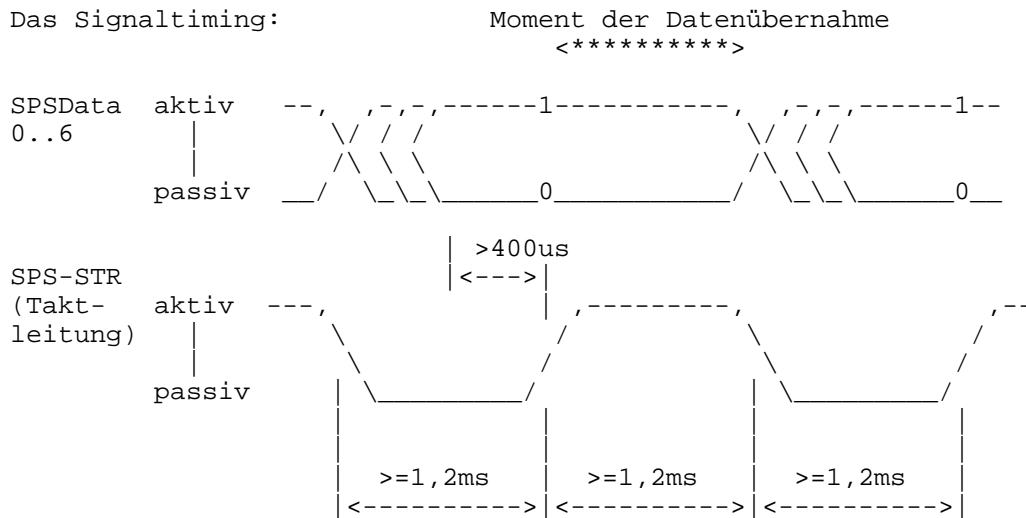
Verbindungen in einem System mit SPS und einem Controller

* SPS *				* Controller *
=====				=====
24V	O----->-----O	31a	+AO	Speisung Source - Drivers A00 bis A07
In0	O-----<-----O	24c	A00	"Ready" (=Regler bereit)
In1	O-----<-----O	25c	A01	"In Position"
In2	O-----<-----O	26c	A02	"Position Error"
In3	O-----<-----O	27c	A03	"Limit Error"
In4	O-----<-----O	28c	A04	"Torque"
		29c	A05	"Trigger"
		30c	A06	"Trigger Direction"
Inx	O-----<-----O	31c	A07	"SPSAck" (=Rückmeldung)
Out0	O----->-----O	16c	DIO0	"SPSData0" \
Out1	O----->-----O	17c	DIO1	"SPSData1" \
Out2	O----->-----O	18c	DIO2	"SPSData2" \
Out3	O----->-----O	19c	DIO3	"SPSData3" \
Out4	O----->-----O	20c	DIO4	"SPSData4" /
Out5	O----->-----O	21c	DIO5	"SPSData5" /
Out6	O----->-----O	22c	DIO6	"SPSData6" /
Out7	O			
Outy	O----->-----O	23c	DIO7	"SPS-STR" (=Taktleitung)
0V	O----->-----O	15c	F0V	"0 Volt"

Bit 0..6 des
zu über-
tragenden
Zeichens

Timing beim Betrieb mit der parallelen Schnittstelle:

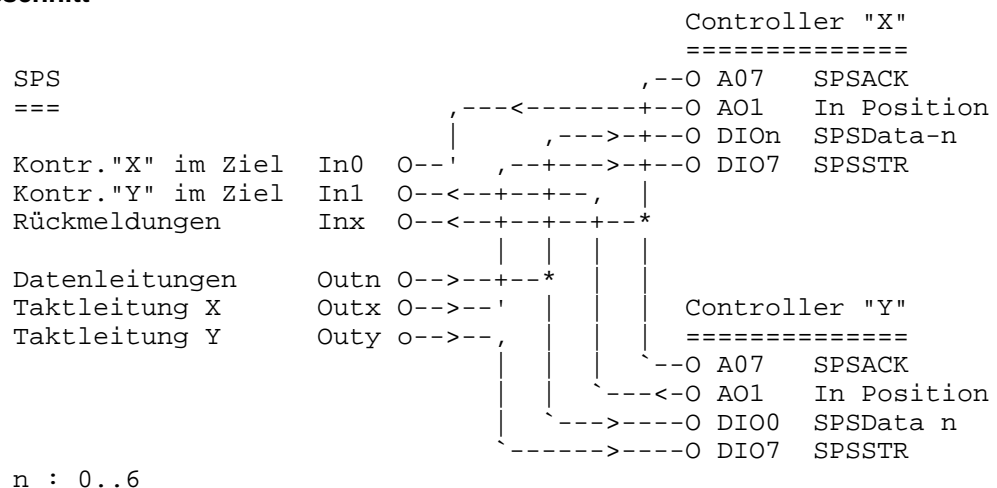
- Die Daten an den Pin DIO0 bis DIO6 müssen wenigstens 400 Mikrosekunden vor der passiv - aktiv Flanke des Taktsignals anliegen. Bei einer langsameren SPS werden also die Daten einen ganzen Zyklus vor dem Taktsignal ausgegeben.
- Danach müssen das Taktsignal und die Datenleitungen für mindestens 1,2 Millisekunden stabil bleiben, sie können also üblicherweise im darauf folgenden SPS - Zyklus gelöscht werden.
- Die Taktleitung muss jeweils ebenfalls mindestens 1,2 Millisekunden passiv sein. In dieser Zeit können aber schon die nächsten Daten ausgegeben werden.
- Der Digitalausgang "In Position" wird innerhalb von 100 Mikrosekunden nach dem SPS - ACK - Signal auf einen "Go" - Befehl gelöscht.
- Die Übertragungszeit für ein Zeichen ist somit mindestens 2,4 Millisekunden (bei genügend schneller SPS) oder allgemeiner: 2 Zykluszeiten einer langsameren SPS. Ein Telegramm, wie unser Beispiel mit 9 Zeichen, kann in 18 SPS - Zyklen übertragen werden.



Netzwerk mit paralleler Schnittstelle: (Vgl. Kap. 4.2.2)

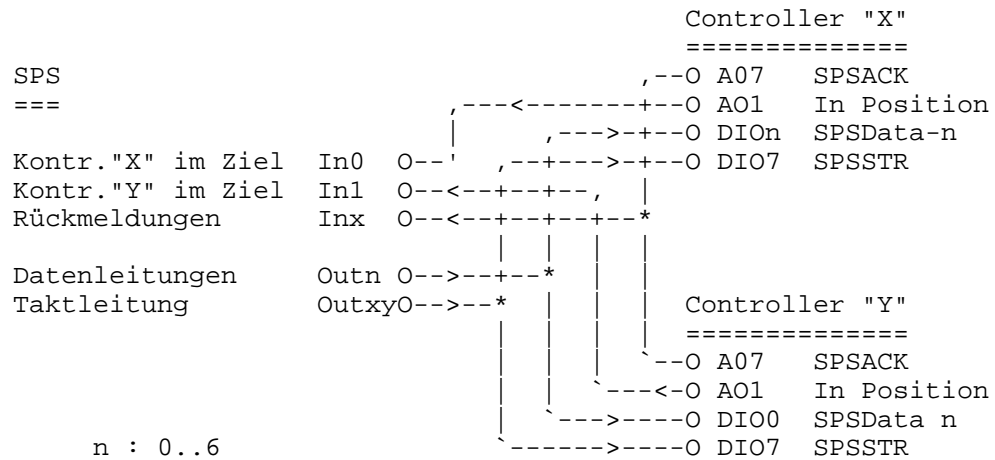
Verbindungen in einem System mit SPS und mehreren Controllern und getrennten Taktleitungen, mit welchen die Controller getrennt angesprochen werden können:

Ein Schemaausschnitt



Verbindungen in einem System mit SPS und mehreren Controller und gemeinsamer Taktleitung. Die Controller werden anhand ihrer Adresse unterschieden:

Ein Schemaausschnitt



4.8 Unterschiede zwischen serieller- und paralleler Schnittstelle

- Über die serielle Schnittstelle kann ein vollständiger Informationsaustausch zwischen Master und Controller stattfinden. Über die parallele Schnittstelle ist es möglich alle Informationen an den Controller zu senden, es werden allerdings nur die allerwichtigsten Informationen (gewisse Statussignale) vom Controller an den Master zurückgesendet.
- Im Makro-Modus (DIP - Switch 8 aktiv) ist es bei installiertem SPS-Makro möglich, auch über die serielle Schnittstelle Daten auszutauschen. Im PC-Betrieb (DIP - Switch 8 passiv) ist die parallele Schnittstelle nicht existent, die Digitaleingänge besitzen ihre gem. Datenblatt zugeordneten Funktionen.

4.9 Adresse im Netzwerk

Sobald mehr als ein Controller an der gleichen Schnittstelle angeschlossen werden (RS485 oder parallele Schnittstelle mit gemeinsamer Taktleitung), muss jeder dieser Controller individuell angesprochen werden können. Dies geschieht mit der Adresse, welche einen Wert von 0 bis 7 einnehmen kann und mit den DIP - Switches S1 bis S3 eingestellt wird (vgl. Kap. 7.1). Mit der Standardsoftware können somit maximal acht Controller in einem Netzwerk vorhanden sein. (Hardwaremässig sind bei RS485 bis 32 möglich)

Adresse	DIP - Switches		
	S3	S2	S1
0	off	off	off
1	off	off	on
2	off	on	off
3	off	on	on
4	on	off	off
5	on	off	on
6	on	on	off
7	on	on	on

- Wir empfehlen nach dem Umschalten der Adresse mit den DIP - Switches S1 bis S3, den Controller kurzzeitig von der Speisung abzutrennen.
- Die Adresse 0 ist besonderer Art: Ein Controller, welcher beim Einschalten die Adresse 0 hat, setzt sich automatisch in den "angesprochenen Zustand", der Master kann mit diesem Controller sofort kommunizieren, ohne vorgängig mit dem Kommando "@" diese Adresse anzusprechen.
- Wir raten dringend davon ab, in einem Netzwerk eine Adresse mehr als einmal zu vergeben!
- Kap. 5 zeigt alle Befehle, welche verwendet werden können. Das "@" - Kommando ist spezieller Natur und wird nur in einem Netzwerk benötigt. Mit dem "@" - Kommando kann derjenige Controller definiert werden, mit welchem die ganze nachfolgende Kommunikation stattfinden soll, und zwar solange, bis ein neues "@" - Kommando einen neuen Controller definiert. Ist in diesem Netzwerk ein Controller mit der Adresse 0 installiert, wird nach dem Einschalten automatisch mit diesem kommuniziert.

5 Befehle

Dieses Kapitel beschreibt die Befehle des Controllers. Die Befehle bestehen aus einem Kommando, ggf. gefolgt von einem oder mehreren Parametern.

Bemerkung zur Kurzschreibweise der Befehle:

- Das Erste Zeichen (der Grossbuchstabe) entspricht dem eigentlichen Kommandozeichen.
- "&" ist ein Platzhalter und steht für die fakultative Befehlserweiterung: "-", "+" oder "*" (Das Zeichen "&" darf nicht gesendet werden).
- Jeder Kleinbuchstabe steht für ein ASCII - Zeichen - Parameter, die Anzahl der Kleinbuchstaben entspricht der maximalen Anzahl Zeichen.

5.1 Aufbau

<pre> +-----+ +-----+ VV </pre>	<p>Kommando: Gültig sind die ASCII-Zeichen "@".."_" (40hex..5Fhex), alle anderen ASCII-Zeichen erzeugen eine Fehlermeldung.</p> <p>Kommando - Erweiterung (optional): Gültige Möglichkeiten sind je nach Kommando:</p> <ul style="list-style-type: none"> - Space, wird wie "kein Erweiterungszeichen" behandelt. - Zeichen "*", "+" oder "-". - Alle anderen Zeichen werden ignoriert.
<p>Beispiel:</p> <pre> T&ttttt --+ </pre>	<pre> +-----+ </pre> <p>Parameterblock (optional). Er unterliegt je nach Kommando unterschiedlichen Restriktionen, vgl. Kommandobeschreibung (Kap. 5.3)</p>

Die Software wird ständig weiterentwickelt und den neuen Gegebenheiten angepasst. Bei geänderten Befehlen steht im Kapitel 5.3 jeweils beim Befehl (am rechten Seitenrand und in Klammern), seit welcher Softwareversion der entsprechende Befehl (wie dann folgend beschrieben) gilt. z.B. (V1.07) heisst, dass dieser Befehl ab Version 1.07 so wie beschrieben, implementiert ist. Beim Betrieb mit älterer Software muss das entsprechende Handbuch benutzt werden.

ACHTUNG:

Einige Kommandos haben eine andere Funktion, je nachdem, ob sie mit oder ohne Parameter übertragen werden.

In einem Parameterblock können führende Nullen weggelassen werden.

Die im Kap. 5.3 mit der Anzahl Kleinbuchstaben angegebene maximale Länge des Parameterblockes ist in diesem Sinne unrichtig, als die Anzahl der Kleinbuchstaben nur die maximale Anzahl der relevanten Parameterzeichen angibt. Es ist möglich, aus welchen Gründen auch immer, einen Parameterblock vor den relevanten Zeichen mit Nullen zu füllen, die dann allerdings nicht gelesen werden.

Der Befehl "A0025" kann kürzer auch als "A25" oder aber länger als "A00025" übertragen werden. Mehr als sechs Zeichen im Parameterblock sind möglich, werden aber nicht empfohlen. Bei etwa 12 Zeichen entsteht eine Fehlermeldung.

5.2 Zusammenfassung der Befehle

Netzwerk - Befehle

@	Get Address	Aktuelle Kommunikationsadresse des aktiven Controllers abfragen
@a	Set Address	Neue Kommunikationsadr. setzen (dieser Controller kommuniziert nun)

Allgemeine Befehle

A	Get Acceleration	Beschleunigungswert abfragen
Aaaaa	Set Acceleration	Beschleunigungswert setzen
B	Get Breakpoint / Capture	Triggerpunkt abfragen
B&bbbbbb	Set Breakpoint	Triggerpunkt setzten und aktivieren
B*	Reset Breakpoint	Triggerpunkt wieder setzten
C	Get Controller parameter	Regelparameter des Controllers abfragen
Cppiidd	Set Controller parameter	Regelparameter des Controllers setzen
D	Get Definitions	Systemparameter abfragen
Dxxxxvv	Set Definitions	Systemparameter setzen
E	Get Error	Fehlercode abfragen und löschen
F	Get Flags	System - Bit - Parameter abfragen
F&fff	Set Flags	System - Bit - Parameter setzen
G	Go	Auslösen einer geladenen Bewegung
H	Home	Fahrt an den Referenzpunkt (0)
Ii	Get Inputs	Digitaleingang, DIP - Switches oder Spannungen abfragen
J	Get Jump	Sprungweite (oder Endstufentyp) abfragen
J&jjjjjj	Get Jump	Sprungweite setzen (relative Zielangabe)
K	Get K-Value	Koppelfaktor des Handrades (Indexer) abfragen
K&kkkkkk	Set K-Value	Koppelfaktor des Handrades (Indexer) setzen
L	Get Lower Limit	Untere (negative) Softwaregrenze abfragen
L&llllll	Set Lower Limit	Untere (negative) Softwaregrenze setzen
Mm	Set Mode	Betriebsart setzen
O	Get Outputs	Digitale Ausgänge abfragen
O&oo	Set Outputs	Digitale Ausgänge setzen
P	Get Position	Momentane Ist - Position abfragen
S	Get Status	Status des Controllers abfragen
T	Get Target	Zielposition abfragen
T&tttttt	Set Target	Zielposition setzten (absolute Zielangabe)
U	Get Upper Limit	Obere (positive) Softwaregrenze abfragen
U&uuuuuu	Set Upper Limit	Obere (positive) Softwaregrenze setzen
V	Get Velocity	Drehzahl (-grenz-) - Wert abfragen
V&vvvv	Set Velocity	Drehzahl (-grenz-) - Wert setzen
X	Get eXtremity Value	Grenzwerte (Positionsgrenzen, I-Anteil) abfragen
Xiiddss	Set eXtremity Value	Grenzwerte (Positionsgrenzen, I-Anteil) setzen
Y	Get Y-Value	Stellgröße (I rated) abfragen
Yyy	Set Y-Value	Stellgröße (I rated) setzen
Zz	Set Zero	Referenzierung des Antriebes
Z&zzzzzz	Shift Zero	Verschieben des Masstabes relativ

Spezialbefehle

\	Get SW - Identification	Softwareidentifikation abfragen
_*xx	System Reset	Neustart der Controllersoftware

5.3 Detailbeschreibung der Befehle

@

Ask Address - Erfragt die Adresse des momentan im Netzwerk kommunikativen Gerätes. Die Anfrage geht an alle Geräte im Netzwerk. Nur das zurzeit aktive sendet seine Adresse zurück. Erhält man keine Antwort, ist kein Gerät kommunikationsaktiv. Überprüfen Sie die DIP - Switch Einstellungen und setzen sie eine vorhandene Adresse. Erhalten Sie immer noch keine Antwort, müssen Sie die Kommunikationsverbindungen überprüfen.

Parameter

keine

Antwort

@aa

Die Antwort umfasst das Kommando "@", danach 2 ASCII-Zeichen für die Adresse. Adresse 0..7, aa also "00" bis "07".

Beispiel

@ Erfragt die aktuell angesprochene Adresse. Als Antwort zum Beispiel:
@02 Der momentan aktive Controller hat die Netzwerkadresse 2.

@a

Set Address - Setze neue Kommunikationsadresse.

Alle weiteren Befehle werden nur noch vom Gerät mit der angegebenen Adresse behandelt. Ein unter Umständen bisher aktives Gerät schaltet sich auf stumm und sendet auch keine Quittung zurück. Sie erhalten die Quittung des neu angesprochenen Gerätes (vgl. Kap. 4.9). Erhalten Sie keine Quittung, ist unter der neuen Adresse kein Gerät verfügbar.

Parameter

1 ASCII-Zeichen

Wert

0..Fhex

Mit den DIP - Schaltern S1..S3 kann an jedem Gerät eine Kommunikationsadresse im Bereich 0..7 (3 Bit) eingestellt werden. Werte ausserhalb dieses Bereichs sprechen eine nicht existente Adresse an. Die Folge ist keine Antwort aus dem Netzwerk.

Beispiel

@1 Aktiviert Kommunikation mit Controller 1.
S Erfragt Status des Controller 1. Antwort z.B.:
S04 Controller 1 ist in Zielposition
@3 Aktiviert Kommunikation mit Controller 3.
M0 Entfernt das Moment am Motor 3

Bemerkung

- Sind die DIP - Schalter eines Reglers auf Adresse 0 gestellt, wenn dieser eingeschaltet wird, wird automatisch die Kommunikation mit diesem Regler freigegeben. In einem System mit nur einer Reglerkarte (und diese auf Adresse 0) kann ohne dem Befehl @ gearbeitet werden.
- Umschalten der Adresse (DIP – Schalter verstellen) während des Betriebs wird nicht empfohlen, da das Gerät bis auf weiteres (Neuinitialisierung der Controller - Software) an der bisherigen Adresse aktiv bleibt.
- Der Betrieb in Mehrachssystemen mit mehreren gleichen Adressen wird nicht empfohlen.

A

Get Acceleration - Erfragt (den zuletzt geladenen) Beschleunigungswert.

Aufgrund dieses Befehls gibt das Gerät den zuletzt mit Aaaaa geladenen Beschleunigungswert als 2 Byte in 4 ASCII-Zeichen zurück.

Parameter

keine

Antwort

Aaaaa

Die Antwort umfasst das Kommando "A" danach 4 ASCII-Zeichen für die 2 Byte des momentanen Beschleunigungswertes.

Der Dezimalpunkt liegt zwischen dem 2. und dem 3. ASCII-Zeichen (vgl. Kap. 6 und Anhang Kap. 12.1 und 12.3)

Einheit ist Inkremente pro Regelzyklus im Quadrat. Das heisst, pro Reglerzyklus wird die Drehzahl um die angegebene Anzahl Inkremente pro Regelzyklus verändert (gilt nur im Rampenbereich).

Beispiel

A Erfragt den aktuellen Beschleunigungswert. Antwort zum Beispiel:

A0040 Er ist 00,40 hex (0,25 dezimal) Inkremente pro Regelzyklus im Quadrat.

Aaaaa

Set Acceleration - Setzt einen neuen Beschleunigungswert.

Die neue Steilheit der Beschleunigungs- bzw. Bremsrampe wird erst bei nächsten Übergang von einer Drehzahl auf eine andere übernommen.

Parameter

4 ASCII-Zeichen

Wert

lhex bis 7FFhex und aaaa + vvvv > FFFFhex.

Werte ausserhalb werden zurückgewiesen und erzeugen einen Kommandoerror.

Der Dezimalpunkt liegt zwischen dem 2. und dem 3. ASCII-Zeichen (vgl. Kap. 6 und Anhang Kap. 12.1 und 12.3)

Ferner muss die Summe aus Beschleunigung und Geschwindigkeit kleiner als FFFFhex sein.

Einheit ist Inkremente pro Regelzyklus im Quadrat. Das heisst, pro Reglerzyklus wird die Drehzahl um die angegebene Anzahl Inkremente pro Regelzyklus verändert (gilt nur im Rampenbereich).

Beispiel

A204 Setzt die Rampe ab dem nächsten Nulldurchgang auf 2.04hex (2,0156 dezimal).

Also jede 1024us wird bei einer Beschleunigung (oder Verzögerung) der Drehzahlsollwert in der Rampe um 2.04hex Inkremente verändert.

Bemerkung

- Der Wert kann jederzeit geändert werden, er wird aber erst beim nächsten Drehzahlnulldurchgang aktiviert - dies also auch während einer Drehrichtungsänderung.
- Die kleinstmögliche Drehzahl (ausser Stillstand) ist immer gleich oder grösser dem Beschleunigungswert (vgl. Kap. 6)
- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist 0020, nach einer Nullfahrt 0010

B

(V1.09c)

Ask Breakpoint and Load Capture - Trigger - Erfragt den Inhalt des B - Registers. Eine allfällig "scharfe" Trigger - Funktion wird gesperrt. Das B-Register wird durch gewisse Z-Befehle gesetzt (Distanz zwischen Vorreferenz und Z-Impuls) oder, sofern Capture Vorgang, durch einen externen Trigger (aktuelle Position).

Parameter

keine

Antwort

Bbbbbbb

Die Antwort umfasst das Kommando "B" danach 6 ASCII-Zeichen für die 3 Byte der Position in Inkrementen.

Beispiel

B Startet einen Capture - Vorgang. Die Antwort auf diesen Befehl kann u.U. ohne Bedeutung sein. Nun erscheint im Statusbyte an der Stelle "CapVal" (Bit 5) eine 0. Sobald am Eingang DIO7 eine 0-1-Flanke eintrifft, wird die damals aktuelle Position ins B - Register gespeichert und Status - Bit 5 auf 1 gesetzt. Danach kann die gespeicherte Position abgefragt werden:

B Als Antwort erscheint z.B.:

B6432AE Die zum Zeitpunkt des 0-1-Flanke am digitalen Eingang DIO7 aktuelle Position war 6432AEhex.

B&bbbbbb

(V1.09c)

Set Brake-Point - Setze neuen Triggerpunkt.

Setzt an die angegebene Position einen Punkt, bei dessen Überfahren gewisse Aktionen ausgelöst werden können (das Flag "BrPass" wird gesetzt und ggf. eine Bewegung ausgelöst, falls das Flag "AXeq" aktiv ist.). Zugleich wird mit dem Setzen oder Versetzen der Auslösemechanismus aktiviert.

ACHTUNG: Das B-Register wird bei einer Referenzierung überschrieben (siehe Z-Befehl)

Parameter

Eine oder keine Kommandoerweiterung, dann 6 ASCII-Zeichen

Kommandoerweiterung

Nur das Zeichen "-" hat einen Einfluss: Es gibt einen negativen Wert für bbbbbbb an. (wenn "" vgl. Befehl B*)

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt (vgl. Anhang Kap. 12.2).

Einheit ist Inkremente (vgl. Kap. 6)

Beispiel

BA02F36 Setzt den Triggerpunkt sofort an die Stelle A02F36hex und entriegelt die Fähigkeit zur Auslösung gewisser Aktionen (falls nicht schon von früheren Befehlen entriegelt), löscht das "BrPass" - Flag

B11F Setzt Triggerpunkt nach 00011Fhex.

Bemerkung

- Mit dem Flag AXEQ = 1 wird bei einer Überquerung des Triggerpunktes eine Bewegung (deren Kennwerte, wie Zielposition, Drehzahl und Beschleunigung vorangehend definiert wurden) ausgelöst.
- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist 000000
- Nach einer Referenzierung enthält das B-Register die Distanz zum Nullimpuls (siehe Z-Befehl).

B*

(V1.05h)

Reactivate Brake-Point - Erneure "alten" Triggerpunkt.

Das B* - Kommando reaktiviert den zuletzt definierten Triggerpunkt.

Parameter

keine

C

Ask Controllerparameter - Erfragt die Regelparameter.

Aufgrund dieses Befehls gibt der Controller folgende Regelparameter zurück:

- Proportionalfaktor KP (8 Bit)
- Integralfaktor KI (8 Bit)
- Differentialfaktor KD (8 Bit)

Parameter

keine

Antwort

Cppiidd

Die Antwort umfasst das Kommando "C", danach je 2 ASCII-Zeichen für KP, KI und KD.

Für Wert und Einheit (vgl. Befehl "Cppiidd")

Beispiel

C Fragt die aktuellen Regelparameter ab. Als Antwort zum Beispiel:

C061F4A Die Regelparameter sind: KP=06hex, KI=1Fhex, KD=4Ahex.

Cppiidd

Set Controllerparameter - Setzt die Regelparameter.

- Proportionalfaktor KP (8 Bit)
- Integralfaktor KI (8 Bit)
- Differentialfaktor KD (8 Bit)

Parameter

Pro Regelparameter 2 ASCII-Zeichen, total also 6 ASCII-Zeichen.

Wert

Jeder Regelparameter für sich: 00hex bis FFhex, (nicht alle Werte sind regeltechnisch sinnvoll).

Einheiten:

- KP: $(10 \text{ Volt} / 128) * \text{Volt pro Inkrement}$. Der KP-Wert 1 bedeutet, dass sich die Ausgangsspannung am DA - Wandler um 78mVolt pro Inkrement ändert.
- KI: $(10 \text{ Volt} / 32768) * \text{Volt pro [Inkrement mal Regelzyklen]}$. Der KI-Wert 1 bedeutet, dass sich die Ausgangsspannung am DA - Wandler um 0,305mVolt pro Inkrement und pro Regelzyklus ändern würde. Da der DA - Wandler 12-Bit (+-11 Bit) auflöst, ist die kleinste Schrittweite 16mal grösser, also 4,89mVolt.
- KD: $(10 \text{ Volt} / 128) * \text{Volt} * \text{Regelzyklen pro Inkrement}$. Der KD-Wert 1 bedeutet, dass sich die Ausgangsspannung am DA - Wandler 78mVolt pro [Inkrement pro Regelzyklus] ändert.

Beispiel

C021E34 Setzt (unmittelbar) die Regelparameter: KP:=02hex, KI:=1Ehex und KD:=34hex

C40 Setzt die Regelparameter KP und KI auf 00hex, KD auf 40hex

Bemerkung

- Werden alle Parameter 0 gesetzt, ist der Motor im Leerlauf.
- Die Werte werden im NOVRAM abgelegt.
- Der Initialisierungswert ist 200220

D

(V1.07)

Ask Definitions - Erfragt die D - Parameter.

Die Antwortparameter erscheinen in derselben Reihenfolge, wie sie mit Dxxvii gesetzt werden können.

Parameter

keine

Antwort

Dxxvii

Für Wert und Einheit (vgl. Befehl "Dxxvii")

Dxxvii

(V1.09c)

Definition Of Zero - Velocity - Window and Polarity Of Opto - Inputs - Setzt die D - Parameter.

xx hat keine Bedeutung.

vv Fenstergröße der V-Null-Geschwindigkeit (Blockiergeschwindigkeit siehe auch VZero - im F - Register)

ii Bitmuster bestimmt die Polarität der Optokopler - Eingänge.

Parameter

xx 0..2 ASCII-Zeichen

vv V - Null Wert 2 ASCII-Zeichen,

ii Polaritäts - Bitmuster 2 ASCII-Zeichen

Total also bis 6 ASCII-Zeichen.

Wert

vv 00hex bis FFhex

ii 00hex bis FFhex Bit = 1: Eingang powered = Aktiv (Schliesser)

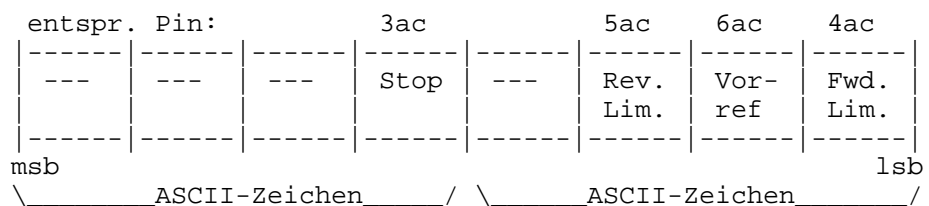
Bit = 0: Eingang not powered = Aktiv (Kabelbruch sicher, Öffner)

ACHTUNG: Am Eingang VorRef muss ein Schliesser sein, da die Referenzierung eine 0-1 -Flanke voraussetzt (siehe Z-Befehl)!**Einheit:**

vv Inkremente pro Reglerzyklus.

Geschwindigkeitswert Testgrenze: 8 Nachkommastellen. 0.80hex als vv=80 übergeben.

Zuordnung des Bitmusters zu den Optokopler - Eingängen:

**Beispiel**

D004002 Setzt die Vergleichsdrehzahl auf 0.4hex Inkremente pro Regelzyklus. Die Eingänge Stop, vorderer und hinterer Endschalter sind Öffner, die Vorreferenz ist Schliesser

Bemerkung

- Die Werte werden im NOVRAM abgelegt, xx wird nicht verwendet.
- Der Initialisierungswert ist 000000
- VorRef muss eine 0-1 - Flanke sein

E

(V1.09d)

Ask Error - Erfragt die Fehlerflags.

Aufgrund dieses Befehls gibt der Controller den Zustand der Fehlerflags als 8-Bit-Wert in 2 ASCII-Zeichen zurück. Ob mindestens ein Flag gesetzt ist, zeigt das Flag ErrFlg des Registers S.

ACHTUNG: Durch eine Abfrage werden folgende Register gelöscht: CmdErr, TarErr, PwDwn und MemLst.

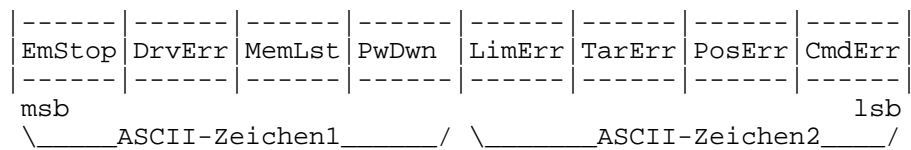
Parameter

keine

Antwort

Eee Die Antwort umfasst das Kommando "E", danach 2 ASCII-Zeichen welche 8 Flags repräsentieren: (Jedes ASCII-Zeichen repräsentiert mit seinem hexadezimalen 4 Bit - Wert je 4 Flags)

Bitmuster des Fehlerbytes:



- **EmStop** Emergency - Stop, Notstopp ausgelöst.
Wird entsprechend dem "Stop" - Eingang (Pin 3a/c) gesetzt oder gelöscht.

- **DrvErr** Power-Driver Error: Fehler in der Endstufe.
Je nach Hardware (Enstufentyp und Hardware-Version) kann dieser Fehler bedeuten: Überspannung, Übertemperatur oder Kurzschluss. Das Flag wird von der Endstufe gesetzt und auch wieder gelöscht (Ausnahme: Nach einem Kurzschluss muss die Endstufe durch einen M0-Befehl zurückgesetzt werden).
ACHTUNG: Dieses Flag wird mit dem internen Reglerfreigabesignal gefiltert. Das heisst, damit dieser Fehler angezeigt wird, muss das Moment eingeschaltet sein (Run-Signal am Eingang aktiv, Reglerbetriebsart: M1, M2 oder M3).

- **MemLst** Memory - Lost, Parameter - Daten gingen verloren.
Bedeutet, dass alle Konstanten mit den Urdadewerten neu initialisiert wurden.
Wird gesetzt, wenn die Checksumme im NovRam nicht mehr stimmt, wodurch eine Neuinitialisierung ausgelöst wird. Tritt dieser Fehler häufig auf, muss der Batteriegestützte Speicherbaustein ersetzt werden.
Mit dem Lesen dieses Registers wird auch das Flag zurückgesetzt.

- **PwDwn** Power - Down, Neustart des Controllers.
Wird gesetzt, wenn der Prozessor neu gestartet wird weil die Speisung ausgefallen war.
Mit dem Lesen dieses Registers, wird das Flag zurückgesetzt.

- **LimErr** Limit - Error, Bereichsfehler.
Wird gesetzt, wenn ein Endschalter überfahren wird. Dieses Flag wird wieder gelöscht, wenn die Bewegung, aus dem durch den Endschalter markierten Bereich erfolgt.

- **TarErr** Target - Error, Zielfehler.
Flag wird gesetzt, wenn versucht wurde, ein Ziel hinter der Softwarelimite anzufahren.
Das Flag wird durch das Abfragen des E-Registers gelöscht.

- **PosErr** Position - Error. Positions- ilder Drehzahlfehler.
Aktiv, wenn während der Fahrt die Differenz von Soll- zu Istposition zu gross wird (Schwelle siehe Befehl Xdd).
Wird gelöscht, wenn der Positionsfehler wieder unter der Schwelle liegt, oder wenn die Zielposition erreicht wurde.

- **CmdErr** Command - Error, Fehler im empfangenen Befehl.
Com Err wird gesetzt, wenn beim Befehl:
A der Wert >7FFFhex ist, der Wert=0 ist, oder die Summe von A und V > FFFE hex ist.

- G** ein Go nicht ausgeführt wird, weil entweder das EmStop-Flag (Reg E), das DrvErr-Flag (Reg E) oder das DrvOff-Flag (Reg S) aktiv ist.
- H** ein Home nicht ausgeführt werden kann, weil entweder das EmStop-Flag (Reg E), das DrvErr-Flag (Reg E), das DrvOff-Flag (Reg S) oder entweder das PosBlk- oder das NegBlk-Flag (Reg F) aktiv ist.
- I** der Wert grösser als 10 ist.
- M** der Parameter fehlt oder der Wert nicht 1, 2 oder 3 ist oder wenn der RUN - Eingang (Pin 16c) passiv ist, wenn der Makro-Modus nicht aktiv ist (DIP - Switch 8).
- V** die Summe aus den Werten von A und V grösser als FFFE hex ist.
- X** der Parameterwert Xii grösser als 7F hex ist, oder Xdd oder Xss Null ist.
- Y** der Wert Null oder grösser als 7F hex ist.
- Z** kein Parameter angegeben ist, oder der Wert grösser als 8 ist.
- _** wenn der Wert nicht *_3A oder *_7F ist.
- Dieses Flag wird mit dem Auslesen des E-Registers gelöscht (E Befehl).

Beispiel

- E Erfragt die aktuellen Fehlerflags.
Als Antwort zum Beispiel:
- E00 bedeutet: Keine Fehler
- E10 bedeutet: Seit dem Controllerstart ist das Register E das erste Mal abgefragt worden (PwDwn immer noch aktiv).

```
02hex -> 0000-0010
          |___ Flag "POSERR" aktiv
```


F

(V1.09)

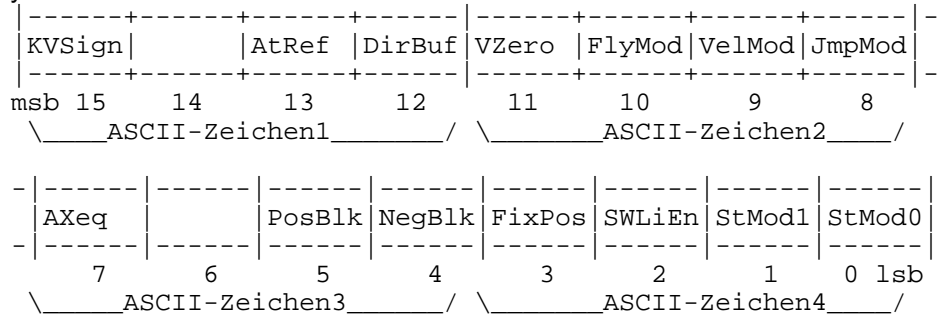
Ask Flags - Erfragt die Reglerflags.

Die Reglerflags sind als zwei Bytes organisiert. Wobei das höhere Byte nur lesbar ist.

Antwort

`Fff` Die Antwort umfasst das Kommando "F", danach 4 ASCII-Zeichen, welche 16 Flags repräsentieren:

Bitmuster der Reglerflagbytes:



Bits 15..8 sind "Nur Lese - Flags": Sie können nicht überschrieben werden:

- **KVSgn** Sign of K - Value. Vorzeichen des mit dem Kommando "K" ladbaren Getriebefaktors.
- **AtRef** At Reference - Position. Falls das System an der Referenzposition (ohne Lagefehler) steht, wird dieses Flag (1) gesetzt (sofern vorher referenziert wurde).
- **DirBuf** Direction - Buffer, Vorzeichen der zurzeit hängigen Bewegung. (0=positive Drehrichtung)
- **VZero** Velocity-Zero (Blockiertest): Gesetzt, wenn der Betrag der momentanen Drehzahl (Encoder) kleiner als der V-Null-Wert ist (vgl. Kommando "D"), die Sollzahl aber höher ist.
- **FlyMod** Fly - Mode, zeigt Reglerbetriebsart 3 an, in welcher Sprünge (Jump) angehängt werden können, bevor der Antrieb im Ziel ist. (Setzen / Löschen mittels M-Befehl).
- **VelMod** Velocity - Mode, zeigt Regelbetriebsart 2 "Drehzahl - Betrieb" an. (Setzen / Löschen mittels M-Befehl).
- **JmpMod** Jump - Mode, zeigt an, dass ein GO-Kommando eine relative Bewegung auslöst. (Wird durch Schreiben ins J-Register gesetzt und gelöscht durch Schreiben ins T-Register).

Bits 7..0 sind Schreib- und Lese - Flags, sie können verändert werden (vgl. Befehl "Fff"):

- **AXeq** Auto - Execution, Auslösung einer Bewegung, sobald der aktivierte Breakpoint überfahren wird.
- **PosBlk** Positive - Direction blocked. Bewegungen in positive Richtung werden nicht angenommen.
- **NegBlk** Negative - Direction blocked. Bewegungen in negativer Richtung werden nicht angenommen.
- **FixPos** Bindet die Position des Systems bei einer Massstabsverschiebung (Kommando "Z") an den sich verschiebenden Massstab (1) oder an die mechanische Lage (0).
- **SwLiEn** Software - Limit - Enable. Aktiviert die Lagegrenzen, welche nicht überfahren werden können
- **StMod1** Stop - Mode, Art des Anhaltens, durch externer "Stopp" - Eingang. Drehmoment am Motor weg (1) oder "geführter" Sofort - Stopp des Motor - Drivers (0) (vgl. STMOD0).
- **StMod0** (1) Stopp entlang der programmierten Rampe oder (0) Stopp sofort, mit maximalem Strom.

STMOD1	STMOD0	Bem.
0	0	Schnell - Stopp
0	1	Rampen - Stopp
1	x	Moment am Motor weg

Beispiel

`F` Erfragt die aktuellen Reglerflags. Als Antwort zum Beispiel:

`F0186` Automatische Bewegungsauslösung bei Triggerpunkt ein, Jump - Modus, SW-Limit ein, Stoppmodus: Moment weg.

`0186hex -> 0000-0001-1000-0110`

				_ Stop-Mode "ohne Moment"
				_ Endschalter ein
				_ Bewegungsauslösung ein
				_ Jump-Mode aktiv

Bemerkung

- Für die Flags Fly-, Vel- und JmpMod siehe auch Befehl Mm

F&f f

Set Flags - Setzt die Reglerflags.

Von den Reglerflags, welche die Flags 0..15 umfassen, können die Flags 0..7 überschrieben werden. Die F - Flags sind als zwei Byte organisiert.

Parameter

Eine oder keine Kommandoerweiterung, dann 4 ASCII-Zeichen.

Kommando Erweiterung

kein Zeichen	Setzen der Flags, unmittelbare Definition welche 0 und welche 1
*	logische AND-Verknüpfung der Flags (0-Setzen (maskieren) bestimmter Flags)
+	logische OR-Verknüpfung der Flags (1-Setzen (maskieren) bestimmter Flags)
-	Löschen der Flags (0-Setzen bestimmter Flags)

Wert

00hex bis FFhex

Die Flags sind beim Befehl F (ohne Parameter) beschrieben.

Beispiel

F01	Setze Rampenstopp ein, keine Softwaregrenzen, binde Position des Systems bei einer Massstabverschiebung an die mech. Lage, beide Drehrichtungen frei, Breakpoint nicht scharf. xxxx-xxxx => 0000-0001
F1	wie "F01"!
F+10	Unterbinde Bewegung in negative Richtung. Alle anderen Flags bleiben unbeeinflusst. xxxx-xxxx or 0001-0001 => xxxx-x1xx
F*CF	Entriegle beide Drehrichtungen, alle anderen Flags unbeeinflusst. xxxx-xxxx and 1100-1111 => xx00-xxxx
F-30	wie "F*CF", wo eine 1 steht wird 0 gesetzt. xxxx-xxxx and(not(0011-0000))=>xx00-xxxx

Bemerkung

- Die veränderten Flags werden im Parameterspeicher (NovRAM) abgelegt.

G

Go Unmittelbares Auslösen einer Bewegung.

Go startet eine Bewegung, deren Kennwerte (Ziel, max. Drehzahl und Beschleunigung) zuvor mit den entspr. Befehlen übergeben wurden. Dies gilt sowohl aus der Zielposition, wie auch aus einer laufenden Bewegung heraus. Dieser Befehl ist gleichbedeutend mit dem entsprechenden digitalen Eingang.

- Wurde eine Zielposition relativ mit J ("Jump") angegeben, führt jedes "Go" diesen Sprung einmal aus.
- Im Drehzahlbetrieb wird sofort in den neuen Drehzahlwert übergegangen, welcher dann endlos beibehalten wird.

Beispiel

M1	Setze Positionsbetrieb
T30A6	Setze Zielposition
V3FF	Setze Maximaldrehzahl
A3A	Setze Beschleunigung
G	Löse diese Bewegung aus. Dann, nachdem Ziel erreicht wurde:
TFD3014	Setze neues Ziel
G	Löse Bewegung zu neuem Ziel aus. (V- und A - Wert unverändert, Veffektiv in diesem Fall mit umgekehrtem Vorzeichen, da Ziel in anderer Richtung). Dann, unterwegs:
V68F3	Setze neue Maximaldrehzahl
G	Fahre gleiches Ziel mit grösserer Drehzahl an

Bemerkung

Folgende Zustände lösen bei einem Go einen Kommandoerror aus und die Position wird beibehalten:

- Momentloser Zustand (M0).
- Versuch in gesperrte Richtung zu bewegen.
- Versuch hinter eine aktivierte Softwaregrenze zu bewegen.
- Go während Stopp (Eingang 3a/3c) aktiv ist

H

(V1.09)

Home - Unmittelbares Fahren an die Referenzposition.

"Home" startet eine Bewegung, bei der die bestehenden Kennwerte Maximaldrehzahl und Beschleunigung übernommen werden, deren Ziel aber die Referenzposition (000000hex) ist. Ein mit "J" (Jump) eingegebener Sprung und der Jump - Modus wird dadurch aber nicht gelöscht. Die Software - Endschalter werden nicht beachtet. (Die Befehlssequenz "T0" und "G" löscht den Jump - Modus und beachtet bei aktivem SwLiEn die Software - Endschalter).

Dieser Befehl kann auch während der Fahrt ausgelöst werden. Er kann auch vor Erreichen des Ziels mit einem neuen Wert für "T" und anschliessendem "G" abgebrochen werden.

Parameter

keine

Beispiel

T4302F	Setze Zielposition
V100E	Setze Maximaldrehzahl
AA2	Setze Beschleunigung
G	Löse diese Bewegung aus. Nachdem Ziel erreicht wurde, oder unterwegs:
H	Sofort nach Position 0hex fahren

Bemerkung

- "Home" beachtet die blockierten Drehrichtungen (HW - Endschalter).
- Ein Home über eine Softwaregrenze wird ausgeführt, es wird aber ein Target - Error gesetzt, solange die Position ausserhalb der Softwaregrenzen liegt.
- Eine Sprungweite ("J") und der Jump - Modus werden nicht verändert.
- Wurde das System noch nicht referenziert, fährt der Controller an die Position mit dem Wert 0, welche ja immer besteht. (Normalerweise ist dies die Position zum Zeitpunkt des Einschaltens des Controllers).
- Folgende Zustände lösen bei einem Home einen Kommandoerror aus und die Position wird beibehalten:
 - Momentloser Zustand (M0).
 - Versuch bei aktivem Endschalter zu bewegen.

Ii

(V1.04E)

Input Port, DIP - Switches or Voltage - Liest die Digitaleingänge, die DIP - Switches auf dem Controller oder eine Analogspannung (8 Bit Auflösung).

Parameter

Ein ASCII-Zeichen:

- 0 Lesen der 8 Digitaleingänge
 - 1 Lesen der DIP - Switches
 - 2 Spannung X2Pin3 (Endstufentyp)
 - 3 Spannung X2Pin2 (Betrag Ankerstrom)
 - 4 Spannung Potentiometer P3
 - 5 Spannung Potentiometer P2
 - 6 Spannung Potentiometer P1
 - 7 Spannung vom Analogeingang Uin1 (17a)
 - 8 -
 - 9 Spannung vom Analogeingang Uin2 (18a)
 - A Bitmuster der Optokopplereingänge (vgl. Kommando "D")
- andere Keine Antwort, Kommandoerror gesetzt.

Antwort

Iii Die Antwort umfasst das Kommando "I", danach 2 ASCII-Zeichen welche die 8 Eingänge bzw. die 8 DIP - Switches repräsentieren.

Bei Spannungen wird der 8 Bit - Wert in den 2 ASCII-Zeichen übergeben.

Beispiel

- I0 Liest die 8 digitalen Eingänge und gibt als Antwort z.B.:
 - I73 Digitaleingänge sind 01110011 (1 bedeutet aktiv, z.B.: 24 Volt, 0 bedeutet passiv, z.B.: 0V)
- I3 Erfasst den momentanen Betrag des Ankerstromes und erhält als Antwort z.B.:
 - I20 Der Strom beträgt 50% Nennstrom (3Fhex ist 100%) (vgl. Kap. 3.11 und Kap. 6)

J

(V1.11A)

Get Jump-width Liest die zuletzt geladene Sprungweite.

Aufgrund dieses Befehls gibt das Gerät die zuletzt geladene Sprungweite als 3 Byte in 6 ASCII-Zeichen zurück.

Wird kein NovRam eingesetzt, steht in diesem Register nach dem Einschalten des Gerätes der Typ der verwendeten Endstufe. Nach einem Umladen (zurücksetzen auf "factory setting" mit dem Befehl `_*3A`) wird das J-Register und das NovRam ebenfalls auf den Wert der erkannten Endstufe gesetzt. (Tabelle mit den Endstufen siehe unten bei "Bemerkung").

Parameter

keine

Antwort

Jjjjjjj

Die Antwort umfasst das Kommando "J" danach 6 ASCII-Zeichen für die 3 Byte der Sprungweite.

Einheit

Inkremente (vgl. Kap. 6)

Beispiel

J Erfragt die aktuelle Sprungweite. Antwort zum Beispiel:
J00AB40 Sprungweite ist auf AB40hex Inkremente gesetzt.

Bemerkung

- Der Wert wird angegeben, unabhängig davon, ob der Controller im Jump - Modus ist oder nicht. (Vgl. Befehl "F", Jump - Flag)
- Wert nach dem Umladen:

Endstufe	J-Reg	Y-Reg	Dyn. Strom
MD4Q Bauform EK. & DKK	0x10	0x60	50%
MA4Q	0x20	0x7F	100%
MD4QC	0x30	0x60	50%
MA4QC	0x40	0x7F	100%
MD4Q Bauf. BKx ab Rev.E	0x50	0x60	50%
MD4Q Bauf. BKx bis Rev.D	0x60	0x7F	0%

J&jjjjjj

Set Jump-width - Setzt die Sprungweite

Lädt das Ziel der nächsten Bewegung relativ zur momentanen Position. Jedes nachfolgende "G" (Go) setzt das Ziel um diesen Wert in der gleichen Richtung weiter, bis ein Ziel mit dem Befehl "T" (Target), oder eine neue Sprungweite mit dem Kommando "J", eingegeben wird.

Die Sprungweite wird auch relatives Ziel oder Kettenmass genannt.

Parameter

Eine Kommandoerweiterung, dann 6 ASCII-Zeichen

Kommandoerweiterung

Nur das Zeichen "-" hat einen Einfluss, es gibt einen negativen Wert für jjjjjj an.

Alle anderen Zeichen werden ignoriert!

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt (vgl. Anhang Kap. 12.2).

Einheit

Inkremente (vgl. Kap. 6)

Beispiel

J5402 Setzt die Zielposition um 5402hex Inkremente von der momentanen Position in die positive Richtung.

G Der Sprung von 5402hex Inkrementen wird ausgelöst.

G Ein Sprung von 5402h Inkrementen wird nochmals ausgelöst.

H Absolute Fahrt nach 0h.

G Der Sprung von 5402h Inkrementen wird nochmals ausgelöst.

J-12 Setzt die Zielposition um 12hex Inkremente von der momentanen Position in die negative Richtung.

G Löst eine Bewegung an die oben gesetzte Zielposition aus.

G Noch eine Bewegung um 12 Inkremente in negativer Richtung.

Bemerkung

- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist 000000
- Im M1-Modus kann während einer Bewegung mit Jump kein weiterer Jump "angehängt" werden. Dies ist erst möglich, sobald der Antrieb im Ziel (Target) ist.
- Im M3 Modus können die Jumps während der Bewegung bereits mit weiteren Go verlängert werden.
- Wird ein Jump unterbrochen (mit V0, G oder HWStop), kann mit einem G (bzw. Vxxxx, G) das gleiche Ziel angefahren werden.

K

(V1.07)

Get K - Value - Erfragt den Koppelfaktor für den Indexer.

Dieser Wert bestimmt das Übersetzungsverhältnis oder Drehzahlverhältnis des Führungs - Encoders (Indexers) zum Folge - Encoder (System - Encoder). Das Drehrichtungsverhältnis wird mit dem Vorzeichen bestimmt.

Achtung: Dieses Kommando ist nur sinnvoll einsetzbar, wenn die Hardware für die Verarbeitung des zweiten Encoders (=Indexer) ausgerüstet ist! Ansonsten sollte der Faktor Null sein. Null heisst keine Kopplung.

Parameter

keine

Antwort

Kkkkkkkk (vgl. Befehl Kkkkkkk)

K&kkkkkk

(V1.05g)

Set K - Value - Setzt den Koppelfaktor für den Indexer.

Dieser Faktor bestimmt das Übersetzungsverhältnis oder Drehzahlverhältnis des Führungs - Encoders (Indexers) zum Folge - Encoder (System - Encoder). Das Drehrichtungsverhältnis wird mit dem Vorzeichen bestimmt.

Achtung: Dieses Kommando ist nur sinnvoll einsetzbar, wenn die Hardware für die Verarbeitung des zweiten Encoders (=Indexer) ausgerüstet ist! Ansonsten muss der Faktor auf Null gesetzt werden.

Parameter

6 ASCII-Zeichen

Kommando - Erweiterung

Nur das Zeichen "-" hat einen Einfluss, es gibt einen negativen Wert für kkkkkk an. Alle anderen Zeichen werden ignoriert!

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt.

Der Dezimalpunkt liegt zwischen dem 2. und dem 3. ASCII-Zeichen.

Der K - Wert 1,0 (übergeben als "K100") erzeugt eine Encoderumdrehung (Motor) pro Indexerumdrehung (Handrad), falls die Encoder gleiche Impulszahl haben! (Vgl. Kap. 6 und Anhang Kap. 12.1 und 12.3)

Der Wert ist Einheitslos (Inkrement pro Inkrement, vgl. Kap. 6).

Beispiel

K-4000 Sobald das System in der Zielposition steht, kann es durch Drehen des Indexers gedreht werden und zwar, sofern beide Encoder die gleiche Pulszahl aufweisen, mit exakt 25% der Drehzahl des Indexers, aber in entgegengesetzter Richtung.
(0.4000hex = -0.25 dezimal = 25%, vgl. Kap. 12.3)

Bemerkung

- Bei einem Standard - Controller, der die Hardware zur Verarbeitung des Indexers nicht besitzt, sollte der K - Wert sicherheitshalber auf Null ("K0") gesetzt werden, somit ist das System vom (nichtvorhandenen) Indexer entkoppelt.
- Der Wert wird im NovRAM abgelegt.
- Der Initialisierungswert ist 000000

L (V1.07)

Get Lower Logical Limit - Erfragt die untere Softwarelagegrenze.
Der Fahrbereich kann mit diesem Register nach unten Begrenzt sein. (F - Flag SWLEN muss aktiviert sein).

Parameter

keine

Antwort

L111111 (vgl. Befehl L111111)

L&111111 (V1.07)

Set Lower Logical Limit - Setzt die untere Softwarelagegrenze.
Sinnvollerweise sollte 111111 tiefer (mehr negativ) als uuuuuu (vgl. Kommando "U") sein. Aktivierung der Grenzen mit setzen des Flags "SWLEN" (vgl. Kommando "F"). Ein nachfolgender Befehl mit einem Ziel unter dieser Grenze wird zurückgewiesen und Limiterror gesetzt.

Parameter

Kommandoerweiterung, dann 6 ASCII-Zeichen

Kommandoerweiterung

Nur das Zeichen "-" hat einen Einfluss, es gibt einen negativen Wert für 111111 an.
Alle anderen Zeichen werden ignoriert!

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt (vgl. Anhang Kap. 12.2).

Einheit ist Inkremente (vgl. Kap. 6)

Beispiel

L-10000 Setzt die untere (negative) Grenze für Zielangaben auf -10000hex und:
F+4 aktiviert diese Grenze (zusammen mit der ev. noch zu definierenden oberen Grenze).

Bemerkung

- Ungültig ist eine Zielposition, wenn sie unterhalb dieser Softwarelagegrenze liegt: Ziel < Lower Limit erzeugt Target-Error.
- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist 000000

Mm

(1.05h)

Set Mode - Setzt die Betriebsart des Controllers.

Dieser Befehl setzt eine der möglichen Reglerbetriebsarten. Der Befehl kann jederzeit ausgelöst werden, jedoch ist damit vorsichtig umzugehen, um das gewünschte Verhalten zu erzielen!

Parameter

1 ASCII-Zeichen:

- 0 Freischalten des Antriebs. Das Drehmoment am Motor wird entfernt, der Motor läuft frei.
- 1 Positionsbetrieb. Bewegungen werden "Punkt zu Punkt" ausgeführt. Es werden jeweils eine Zielposition, eine Maximaldrehzahl und eine Beschleunigung übergeben. Dann wird mit "G" (Go) die Bewegung ausgelöst. Relative Bewegungen (Jumps) können "unterwegs" NICHT verlängert werden.
- 2 Drehzahlbetrieb. Es werden nur Drehzahl (effektive Drehzahl mit Vorzeichen) und Beschleunigung übergeben. Mit "G" (Go) wird diese Drehzahl eingenommen.
- 3 Fly - Modus, wie "1", aber Jumps können "unterwegs" verlängert werden.

Ein anderer Parameter wird ignoriert und Kommandoerror wird gesetzt.

Beispiel

M1 Der Controller wird in Positionsbetrieb gesetzt.

M0 Der Controller resp. der Motor wird in den drehmomentlosen Zustand geschaltet.

Bemerkung

- Die Flags ""DrvOff", "VelMod" und "FlyMod" werden verändert.
- Beim Übergang vom drehmomentlosen Zustand (Mode 0) in Positions- oder Drehzahlbetrieb wird unmittelbar vor dem Zuschalten des Drehmoments die momentane Position als Zielposition definiert, so dass kein "Windback" stattfindet, falls der Motor von Hand verschoben wurde.
- Im PC-Betrieb, kann mit einem passiven Pin "Run" verhindert werden, dass der Befehl "M1", "M2 oder "M3" ausgeführt wird, es wird Kommandoerror gesetzt.
- Der Jump - Modus ist ein Untermodus im Positionsbetrieb. Das Flag "JmpMod" hat deshalb nur im normalen- und im Fly - Modus eine Bedeutung. Es wird mit `Jjjjjjjj` gesetzt und mit `Tttttttt` gelöscht.

Aufstellung der Modi:

Bezeichnung:	MOTOR OHNE MOMENT	POSITIONS- MODUS	DREHZAHL- MODUS	FLY- MODUS
Einrichten mit:	"M0 "	"M1 "	"M2 "	"M3 "
Flagparameter FlyMod:	0	0	0	1
VelMod:	0	0	1	0
Statusbit DrvOff:	1	0	0	0
Drehzahl während der Fahrt veränderbar: (mittels G-Befehl)	--	ja	ja	ja
Jumps während der Fahrt verlängerbar: (mittels G-Befehl)	--	nein	--	ja

O

(V1.07)

Get Outputs - Zustand des O - Registers abfragen.

Der Rückgabewert wird durch den Befehl "Ooo" bestimmt. Wird diese Anfrage angewendet, wenn DIP - Switch 7 OFF ist, wird nicht der Zustand der Ausgänge angezeigt, sondern das O - Register.

Parameter

keine

Antwort

Ooo (vgl. Kommando Ooo)

O&oo

(V1.07)

Alter Outputs - Ändert den Zustand der digitalen Ausgänge

Ändere die digitalen Ausgänge AO0..AO7 mit logischer Verknüpfung mit den bestehenden Zuständen, sofern der Modus "Commanded Outputs" (DIP - Switch S7 auf "ON") eingestellt ist.

Parameter

2 ASCII-Zeichen. Sie sind die 8-Bit-Daten, die geschrieben werden sollen.

Kommandoerweiterung

kein Zeichen Unmittelbares Setzen der Ausgänge

* AND-en der Ausgänge, 0-Setzen

+ OR-en der Ausgänge, 1-Setzen

- 0-Setzen bestimmter Ausgänge

Wert

00hex bis FFhex.

Beispiel

O A3 Schreibt die Daten A3hex an die 8 Ausgänge. (DIO0 .. DIO7 := 1010'0011)

O *84 alle digitalen Ausgänge, ausser AO2 und AO7 werden gelöscht.

O +01 Digitaler Ausgang AO0 wird gesetzt

O -20 Digitaler Ausgang AO5 wird gelöscht

Bemerkung

- Die veränderten Flags werden im NOVRAM abgelegt.
- Der Initialisierungswert ist 000000
- Allgemein kann eine 1 im Bitmuster als "Ausgang = AKTIV" interpretiert werden, das heisst:
Bei Source - Driver und Pull - Down => HIGH. (bei Open - Collector und Pull - Up -> LOW)

P

Ask Position - Erfragt die momentane Istposition.

Aufgrund dieses Befehls gibt der Controller die momentane Istposition als 3 Byte in 6 ASCII-Zeichen zurück. Bei schnellen Bewegungen sollte man die Übertragungsverzögerung mitberücksichtigen!

Parameter

keine

Antwort

Ppppppp

Die Antwort umfasst das Kommando "P" danach 6 ASCII-Zeichen für die 3 Byte der momentanen Position.

Einheit

Inkremente (vgl. Kap. 6)

Beispiel

P Erfragt die aktuelle Istposition. Als Antwort zum Beispiel:
P0023DA Die momentane Istposition war (zur Zeit der Lesung) 23DAhex.

Bemerkung

- Der Initialisierungswert ist 000000

S

(1.05h)

Ask Status- Erfragt den Reglerstatus.

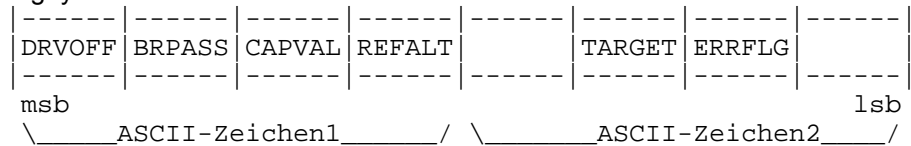
Der Reglerstatus beschreibt den Zustand des Controllers, hardware- und softwaremässig. Es sind acht Statusflags definiert, welche als Byte organisiert sind.

Antwort

Sss

Die Antwort umfasst das Kommando "S", danach 2 ASCII-Zeichen welche 8 Statusflags repräsentieren:

Das Bitmuster des Statusflagbytes:



- **DrvOff** Drive Off, Motor stromlos und dadurch Drehmomentlos, also "ausgeschaltet". (1) bedeutet: kein Drehmoment am Motor.
- **BrPass** Breakpoint Passed, Triggerpunkt wurde überfahren. Sobald der Triggerpunkt aktiviert wird (siehe B&bbbbbb und B*) wird das Flag gelöscht. Es wird durch Überfahren gesetzt.
- **CapVal** Capture Valid. Eine Position, welche infolge einer 0-1-Flanke am digitalen Eingang DIO7 eingelesen wurde, ist bereit, ausgelesen zu werden.
- **RefAlt** Reference Altered or Undefined. Referenzposition wurde verändert (indem der Massstab verschoben wurde) oder Referenz wurde noch nicht definiert. Verschieben des Massstabes macht die Referenz unbrauchbar, was mit (1) angezeigt wird. Vor der ersten Referenzierung ist RefAlt (=1). Eine Referenzfahrt setzt das Flag auf (0).
- **Target** Motor in der Zielposition und innerhalb einer (mit Kommando X) vorgegebenen Toleranz. Es besteht also kein gewichtiger Lagefehler.
- **ErrFlg** Error - Flag - set: Mindestens ein Bit im Errorflagbyte ist gesetzt. ErrFlg wird gelöscht, wenn alle Error - Flags 0 sind.

Beispiel

S Erfragt Reglerstatus. Als Antwort z.B.:

S44 Bedeutet: Der Motor ist im Betrieb. Triggerpunkt inaktiv, Referenz gültig, Regler in Zielposition, kein Fehler aufgetreten.

44hex -> 0100-0100

	_____	in Zielposition
_____		Triggerpunkt inaktiv

T

Get Target - Liest zuletzt geladenes absolutes Ziel.

Aufgrund dieses Befehls gibt der Controller das zuletzt mit Tttttttt geladene absolute Ziel als 3 Byte in 6 ASCII-Zeichen zurück.

Parameter

keine

Antwort

Tttttttt

Die Antwort umfasst das Kommando "T" danach 6 ASCII-Zeichen für die 3 Byte des momentan gültigen absoluten Zieles.

Einheit ist Inkremente (vgl. Kap. 6)

Beispiel

T Erfragt das aktuelle absolute Ziel. Antwort zum Beispiel:
TF044AC Es liegt an der Stelle F044AChex. (Der Antrieb ist entweder unterwegs dorthin, oder aber er könnte mit "G" dorthin geschickt werden)

T&ttttttt

Set Target - Setzt absolutes Ziel, auch Koordinatenmass.

Lädt die Zielposition der nächsten Bewegung absolut. Ein folgendes "Go" löst eine Bewegung zu diesem Punkt aus. Danach ist jedes weitere Go unsinnig, weil damit eine Bewegung an die momentane Position gestartet wird.

Parameter

Kommandoerweiterung, dann 6 ASCII-Zeichen

Kommandoerweiterung

Nur das Zeichen "-" hat einen Einfluss, es gibt einen negativen Wert für ttttttt an. Alle anderen Zeichen werden ignoriert!

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt (vgl. Anhang Kap. 12.2).

Einheit ist Inkremente (vgl. Kap. 6)

Beispiel

T4A333 Setzt das Ziel an die Stelle 4A333hex.
G Die Bewegung dorthin wird gestartet.
T-111 Setzt das Ziel auf -111hex
G Start der Bewegung dorthin.

Bemerkung

- Der Initialisierungswert ist 000000

U

(V1.07)

Get Upper Logical Limit - Erfragt obere Softwarelagegrenze.

Der Fahrbereich kann mit diesem Register nach oben Begrenzt sein. (F - Flag SWLEN muss aktiviert sein).

Parameter

keine

Antwort

Uuuuuuu (vgl. Befehl Uuuuuuu)

U&uuuuuu**Set Upper Logical Limit** - Setzt obere Softwarelagegrenze.

Setzen der oberen programmierbaren Grenze. Sinnvollerweise sollte uuuuuuu höher (mehr positiv) als lllllll (vgl. Kommando "L") sein. Aktivierung der Grenzen mit setzen des Flags "SWLEN" (vgl. Kommando "F"). Ein nachfolgender Befehl mit einem Ziel über dieser Grenze wird zurückgewiesen und Limit - Error gesetzt.

Parameter

Eine oder keine Kommandoerweiterung, dann 6 ASCII-Zeichen

Kommandoerweiterung

Nur das Zeichen "-" hat einen Einfluss, es gibt einen negativen Wert für uuuuuu an. Alle anderen Zeichen werden ignoriert!

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt (vgl. Anhang Kap. 12.2).

Einheit: Inkremente (vgl. Kap. 6)

Beispiel

U40000 Setzt die obere (positive) Grenze für Zielangaben auf 40000hex und:

F+4 aktiviert diese Grenze (zusammen mit der ev. noch zu definierenden unteren Grenze).

Bemerkung

- Ungültig ist eine Zielposition, wenn sie oberhalb dieser Softwarelagegrenze liegt: Ziel >= Upper Limit erzeugt Target - Error.
- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist 000000

V

Get Velocity - Liest die zuletzt geladene Drehzahl

Im Positionsbetrieb wird die maximal erreichbare Drehzahl ausgegeben.

Aufgrund dieses Befehls gibt der Controller die zuletzt mit Vvvvv geladene Drehzahl als 2 Byte in 4 ASCII-Zeichen zurück (vgl. Befehl Vvvvv).

Parameter

keine

Antwort

Vvvvv

Die Antwort umfasst das Kommando "V" danach 4 ASCII-Zeichen für die 2 Byte der Drehzahl.

Einheit ist Inkremente pro Regelzyklus (vgl. auch Vvvvv).

Beispiel

V	Erfragt die aktuelle max. Drehzahl. Antwort zum Beispiel:
V0880	Sie ist 8.80hex Inkremente pro Regelzyklus. (1024 Mikrosekunden)
V4444	Mit einem 1000Ink/U - Encoder dreht der Antrieb mit 1000 U/min
VE0	Drehzahl zum suchen des Z-Impuls: Entspricht 00.E0h = 0.875 Vierfachinkremente pro 1024µs entspricht 12,8 U/min bei einem 1000er Encoder.

Bemerkung

- Der Wert gibt im Drehzahlbetrieb nur den Absolutwert der zu erreichenden Drehzahl zurück. Mit dem Flag "DirBuf" kann das Vorzeichen desselben zurückgelesen werden.
- ACHTUNG: Ein Wert kleiner als A kann zwar gesetzt werden, bei einem "Go" erfolgt aber keine Bewegung, da die Mindestgeschwindigkeit grösser oder gleich dem Wert des A – Registers sein muss!

V&vvvv

Set Velocity - Setzt die Drehzahl oder maximale Drehzahl.

Im Positionsmodus ist dies der Absolutwert, mit dem der Motor maximal auf die Zielposition zufahren darf.

Im Drehzahlmodus ist dies die effektive Drehzahl die eingenommen werden soll.

(Vgl. Kap. 3.2)

Parameter

Kommandoerweiterung, dann 4 ASCII-Zeichen.

Kommandoerweiterung

Nur das Zeichen "-" hat einen Einfluss (und dies ausschliesslich im Drehzahlmodus): Es gibt einen negativen Wert für vvvv an. Alle anderen Zeichen werden ignoriert!

Wert

0hex bis (FFFhex - momentan programmierte Beschleunigung a).

Werte ausserhalb werden zurückgewiesen und erzeugen einen Kommandoerror.

Der Dezimalpunkt liegt zwischen dem 2. und dem 3. ASCII-Zeichen (vgl. Anhang Kap. 12.2 und 12.3).

Einheit ist Inkremente pro Regelzyklus.

Im Positionsbetrieb wird die Position pro Regelzyklus höchstens um die angegebene Anzahl Inkremente verändert.

Im Drehzahlmodus wird die effektive Drehzahl um die angegebene Anzahl Inkremente pro Regelzyklus in der angegebenen Richtung (Vorzeichen) verändert.

Beispiel

M1	Setzt den Controller in den Positionsmodus
V32C0	Setzt die maximal erlaubte Drehzahl (ab nächstem Go - Befehl) auf 32.C0hex. Also pro 1024us wird die Lage um 32.C0hex Inkremente verändert...
M2	Setzt den Regler in den Drehzahlmodus. Dies muss vor dem vorzeichenbehafteten V - Kommando geschehen!
V-2A63	Die Geschwindigkeit ab dem nächsten Go - Befehl beträgt 2A.63hex Inkremente pro 1024us in negativer Richtung

Bemerkung

- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist 1000, Nach einer Referenzierung 00E0.
- Die Umschaltung der Betriebsart (Kommando "M" -> Positionsmodus mit Maximaldrehzahl oder Drehzahlmodus mit effektiver Drehzahl) muss vor der entsprechenden Übergabe der Maximaldrehzahl bzw. Drehzahl mit dem Kommando "V" geschehen.
- Der Wert kann jederzeit geändert werden, er wird aber erst beim nächsten Go - Kommando aktiviert - das allerdings auch während einer laufenden Bewegung!

X

(V1.07)

Get eXtremity - values - Erfragt Regler- und Lagegrenzwerte.

Diese Werte umfassen folgende drei Grenzwerte:

- Integrationsgrenze des PI-Reglers
- Fenstergrösse für Schleppfehler (POSERR)
- Fenstergrösse für Lagefehler (TARGET)

Parameter

keine

Antwort

Xiiddss (vgl. Kommando Xiiddss)

Xiiddss

(V1.09)

Set Extremity - Values - Setzt Regler- und Lagegrenzwerte.

Unmittelbares setzen von drei Grenzwerten:

- **ii**: Integrationsgrenze für die Aufsummierung des Integralanteils (vgl. Kap. 3.9). Es werden 2 ASCII-Zeichen, also 8 Bit übertragen. Der jeden Zyklus aufsummierte Positionsfehler wird auf den 256fachen Wert der Angabe begrenzt.
- **dd**: Maximale Differenz zwischen Soll- und Istposition, innerhalb derer kein Schleppfehler POSERR auftritt (Wird nur während einer Bewegung erzeugt). Es werden 2 ASCII-Zeichen, also 8 Bit übertragen. Der momentane Positionsfehler während einer Bewegung wird mit dem 256fachen Wert der Angabe verglichen.
- **ss**: Maximale Differenz zwischen Soll- und Istposition, innerhalb derer die Zielposition akzeptiert wird. Dieser Wert heisst auch Lagefehler - Grenze. Es werden 2 ASCII-Zeichen, also 8 Bit übertragen. Der momentane Positionsfehler in der Zielposition wird mit dem Wert der Angabe verglichen. (Vgl. Target - Flag im S - Register). (Wird nur in Zielposition erzeugt)

Parameter

Pro Grenzwert 2 ASCII-Zeichen, total also 6 ASCII-Zeichen.

Wert

Integrationsgrenze	ii	00hex..7Fhex	Einheit: 256 Inkremente x Regelzyklen
Schleppfehler	dd	01hex..FFhex	Einheit: 256 Inkremente
Lagefehler	ss	01hex..FFhex	Einheit: Inkremente

Beispiel

X3F2BAA Setzt die Integrationslimite auf +/-3F00hex Zyklen x Inkremente.

Ein Positionsfehler von mehr als +/- 2B00hex Inkremente während einer Bewegung setzt das "POSERR" - Flag.

Ein Positionsfehler von mehr als AAhex Inkremente in der Zielposition unterdrückt das "Target" - Flag.

Bemerkung

- Die Werte werden im NOVRAM abgelegt.
- Der Initialisierungswert ist 404010
- Ist ein Wert ausserhalb seines Definitionsbereiches, wird "CmdErr" gesetzt und keiner der drei Werte übernommen.

Y

(V1.07)

Get Y - Value - Limit - Erfragt Grenzwert der Ausgangsstellgrösse.

Die Ausgangsstellgrösse wird auch Rated I oder Stromsollwert genannt.

Dieser Wert wird bei der Initialisierung entspr. der Endstufe gesetzt und kann mit dem Yyy - Befehl verändert werden.

Parameter

keine

Antwort

YYY (vgl. Befehl YYY)

Yyy

(V1.07)

Set Y - Value - Limit - Setzt den Grenzwert der Ausgangsstellgrösse

Diese Grenze kann als maximaler Strom interpretiert werden, da dieses Signal im Controller als Stromsollwert verwendet wird.

Von dieser Grenze wird nur der positive Wert definiert, der negative Grenzwert entsteht automatisch auf der anderen Seite der Nulllinie (symmetrisch zum positiven Grenzwert).

Parameter

2 ASCII-Zeichen

Wert

01hex..7Fhex Werte ausserhalb erzeugen einen Kommandoerror.

- 01hex entspricht einem Wert nahe Null, was einer Fixierung der Ausgangsspannung auf etwa 0 Volt gleichkommt.
- 7Fhex bedeutet voller Ausgangsbereich, also Ausgangsspannungen für "I Rated" von +/-10 Volt (vgl. Kap. 3.11).

Einheit ist 10V / 128.

Der Wert 1 heisst, dass die Ausgangsspannung +/-78mVolt betragen kann.

Beispiel

Y3B Die Ausgangsspannung "Rated I" wird ab sofort auf $(\pm 59 * 10V / 128) = \pm 4,61$ Volt begrenzt.

Bemerkung

- Das Ausgangssignal von 10 Volt bedeutet 200% des statischen Nennstromes. Kapitel 3.11 gibt genau darüber Auskunft, welche "I Rated" - Spannung welchem Ankerstrom entspricht. Unser obiges Beispiel erzeugt also einen Grenzwert für den Ankerstrom bei ca. +/-92% des statischen Nennstromes.
- Da der DC - B - Typ (vgl. Kap. 2.2) keinen dynamischen Überstrom liefern kann, wurde die interne Strombezugsgrösse verdoppelt, Rated I ist beim B - Typ also 10 Volt für den statischen Nennstrom (beim MA4Q und beim MD4QE nur 5 Volt). Die Y-Werte müssen also beim DC-B Typ doppelt so gross sein wie bei den AC- und DC - E - Typen. (Vgl. Kap. 6)
- Der Wert wird im NOVRAM abgelegt.
- Der Initialisierungswert ist geräteabhängig:

MA4Q:	7F
MD4Q-B:	7F
MD4Q-C:	60
MD4Q-E:	60

Zz

(V1.09m)

Set Position or Velocity to Zero. - Referenzierung durchführen.

Falls kein Endschalter aktiv ist, wird eine Referenzierung (Nullfahrt) ausgelöst (Ausnahme Z0 und Z8). Anschliessend wird die gefundene Position als Referenzpunkt (0-Punkt des Massstabes) gesetzt. Dieser Befehl definiert somit eine schon gesetzte Referenzmarke neu. Die Drehzahl und die Beschleunigung zum suchen des Nullimpuls sind fix ($V=0E0h$, $A=010h$). Der Parameter definiert die Art, wie der Nullimpuls des Encoders gefunden wird. Für eine Referenzierung ausserhalb einer Encoderumdrehung, kann eine Vorreferenz (Schliesser) eingesetzt werden. Die Distanz zum Nullimpuls, resp. zwischen Vorreferenz und Nullimpuls wird im B-Register gespeichert. ACHTUNG: Der Wert kann je nach Drehrichtung negativ sein!

Achtung: Mit einer Kommandoerweiterung wird der Befehl **Z&zzzzzz** ausgeführt!

Parameter

1 ASCII-Zeichen.

Wert

- | | |
|---|---|
| 0 | Unmittelbarer Halt und Ist - Lage auf Null setzen. Dort, wo der Antrieb momentan steht, wird die Referenzmarke gesetzt. (Setzt somit auch die neue Startposition für weitere Jumps). |
| 1 | Unmittelbarer Halt. Der Befehl wird im Fahrbetrieb ausgelöst und lässt die Drehzahl sofort (ohne Rampe) auf Null setzen. (Für Nullsetzen mit Rampe verwende man die Befehle "V0" und "G"). Bei diesem Befehl wird die Referenz NICHT verändert. |
| 2 | Vorwärts fahren bis zum nächsten Nullimpuls des Encoders. Die Distanz steht dann im B-Register. |
| 3 | Rückwärts bis zum nächsten Nullimpuls des Encoders. Die Distanz steht dann im B-Reg.(negativ) |
| 4 | Vorwärts fahren bis zur Vorreferenz ①, dann rückwärts bis zum Nullimpuls des Encoders. Die Distanz von Vorreferenz zum Nullimpuls steht anschliessend im B-Register (negativ). |
| 5 | Rückwärts fahren bis zur Vorreferenz ①, dann vorwärts bis zum Nullimpuls des Encoders. Die Distanz von Vorreferenz zum Nullimpuls steht anschliessend im B-Register. |
| 6 | Vorwärts fahren bis zur Vorreferenz ①, dann vorwärts bis zum Nullimpuls des Encoders. Die Distanz von Vorreferenz zum Nullimpuls steht anschliessend im B-Register. |
| 7 | Rückwärts fahren bis zur Vorreferenz ①, dann rückwärts bis zum Nullimpuls des Encoders. Die Distanz von Vorreferenz zum Nullimpuls steht anschliessend im B-Register (negativ). |
| 8 | Löscht REFALT - Flag ohne andere Aktion. Nötig um über eine Mehrfach - Massstabsverschiebung im Bilde zu sein (Massstabsvergrößerung im Leitreechner). |

Andere:Keine Referenzierung, aber Kommandoerror (CmdErr) wird gesetzt.

① Die Vorreferenz muss eine 0-1-Flanke am Eingang (6a/6c) erzeugen (Schliesser)

Beispiel

Z4 Der Motor fährt mit den momentan gesetzten A- und V - Werten vorwärts bis zur Vorreferenz, dann mit den fixen Werten A0010 und V00E0 rückwärts bis zum Nullimpuls des Encoders. Dort wird die Position Null gesetzt und das System wartet in Positionsregelung. A0010 und V00E0 sind weiterhin aktiv und müssen ev. neu definiert werden. Im B steht die Distanz zw. Vorreferenz und Nullimpuls.

Bemerkung

- Nach einer erfolgreichen Referenzierung (Ausnahme Z1 und Z8) wird das AtRef - Flag (F) gesetzt und P enthält 0 (+/- Xss).
- Alle Z - Befehle löschen das "AXeq" - Flag (F).
- Nach einer Nullfahrt ist $A=10h$ und $V=E0h$, B wurde ebenfalls verändert (Z2..Z7).
- Alle Z - Befehle ausser "Z1" setzen "RefAlt" auf 0
- Z2 und Z3: Steht der Antrieb auf der Nullimpulsmarke des Encoders wenn der Befehl ausgelöst wird, ist undefiniert welche Nullimpulsmarke er anfährt (die gleiche oder diejenige um eine Umdrehung entfernte)!
- Z4 bis Z7: Vorreferenzflanke und Nullimpulsflanke müssen wenigstens 256µs auseinander liegen. Sonst ist nicht definiert, ob die erste Nullimpulsflanke erkannt wird. (B-Register auswerten).
- Z4 bis Z7: Sobald die Vorreferenz gefunden wurde, wird auf diese Position geregelt. Sobald der Antrieb innerhalb des Zielfensters (Xss) ist (Target aktiv), wird die Z-Flanke des Encoders gesucht.
- Fährt man mit einem Z2 bis Z7 auf einen Endschalter, wird das Flag "AtRef" nicht gesetzt. Ist ein Endschaltereingang bereits bei der Auslösung des Z - Befehles aktiv, wird das Flag "CmdErr" gesetzt. In beiden Fällen wird die Ist - Position nicht auf 0 gesetzt. Tipp: Möchte man trotz aktivem Endschalter eine Referenzierung starten muss zuerst mittels F - Befehl das gesetzte Blockierungsflag gelöscht werden.
- Wird einer der Befehle "Z4" bis "Z7" ausgelöst und ist gleichzeitig die Vorreferenz aktiv, dann bewegt sich der Antrieb zuallererst in der inversen Richtung bis die Vorreferenz passiv wird. Erst dann wird die eigentliche Referenzierung eingeleitet.
- Kann der Antrieb nicht unmittelbar bremsen (z.B. Überspannung), positioniert er anschliessend auf $P=0$.
- Störungen (Spannungsspitzen) auf der Vorreferenz oder auf dem Signal des Nullimpulses können eine Referenzierung verunmöglichen!

Z&z z z z z z

Shift Zero - Verschiebt den Massstab (=die Referenz).
Der Massstab wird um den angegebenen Betrag (mit Vorzeichen) verschoben.

Parameter

Eine Kommandoerweiterung, dann 6 ASCII-Zeichen

Kommandoerweiterung

Nur die Zeichen "+" und "-" haben einen Einfluss. Sie zeigen an, dass der Massstab um den angegebenen Wert (inkl. Vorzeichen) verschoben werden soll. Andere Zeichen werden ignoriert!

Achtung: Ohne Kommandoerweiterung wird ein entsprechender Zz - Befehl ausgelöst (vgl. Befehl Zz).

Wert

000000hex bis FFFFFFFhex, alle Werte erlaubt (vgl. Anhang Kap. 12.2).

Einheit

Inkremente (vgl. Kap. 6)

Beispiel

Z-600 Verschiebt den Massstab um 600hex Inkremente in die negative Richtung.

- Mit Parameterflag "FIXPOS" = 0: Wenn die Position des Systems vorher z.B.: +200hex war, ist sie jetzt neu -400hex (ohne, dass der Motor sich bewegt hat).
- Mit Parameterflag "FIXPOS" = 1: Wenn die Position des Systems vorher z.B.: +200hex war, ist sie jetzt immer noch +200hex, aber der Motor ist in eine neue mechanische Lage gesprungen, welche 600 Inkremente negativer, als die alte Lage ist. (Sofern Moment am Motor ein).

Achtung: Unkontrollierte Hinfahrt!

Z+600 Massstab wie zu Beginn des Beispielles. Position ist in jedem Fall wieder +200hex.

Bemerkung

- Da diese Massstabsverschiebung kontrolliert durchgeführt wird, wird das "REFALT"-Flag NICHT gesetzt!
- Diese Operation kann dadurch als Offset - Operation verwendet werden.
- Der Initialisierungswert ist 000000.

\

Get Software - Identification Code - Erfragt den Software - Identifikations - Code.

Jede Softwareversion hat eine Nummer und ein Freigabedatum. Diese Werte sind im EPROM gespeichert und können mit diesem Befehl erfragt werden.

Antwort

\vvssjm

Die Antwort umfasst das Kommando "\", danach erscheinen 6 ASCII-Zeichen mit folgender Bedeutung:

vv Versionsnummer

ss Sub – Versionsnummer (in hexadezimal)

jm Datum, wobei j : hinterste Ziffer der Jahreszahl m : Monat (durchnumeriert von 1 bis C hex)

Beispiel

\ Fragt den SW - Id - Code ab. Als Antwort z.B.:

\010B6A Heisst: Version 1.11 (0B hex =11 dezimal) vom Oktober (A hex=10 dezimal) 2006

_ *xx**System - Reset** - Initiiert ein System - Reset

Die Kommandoerweiterung "*" ist als Sicherheit gegen ungewolltes Auslösen eines Resets gedacht!

Parameter

Die Kommandoerweiterung "*" und die ASCII-Zeichenfolge "3A" oder "7C". Alle anderen Zeichenfolgen führen zu keinem Reset, setzen aber Kommandoerror.

Wert

- 3A System - Reset mit erzwungenem Memory - Lost, das NOVRAM wird mit den Initialisierungswerten geladen.
- 7C System - Reset ohne erzwungen Memory - Lost, das NOVRAM wird nicht überschrieben.

Beispiel

- _*3A Führt einen Hardware - Reset am Prozessor aus und das Memory wird mit den Initialwerten ("factory setting") überschrieben. Im E-Register wird "Memory Lost" gesetzt (vgl. Kommando "E").

6 Parameter

Dieses Kapitel beschreibt die vom Controller verwendeten Grössen.

Die numerischen Parameter werden als ASCII-Zeichenkette unmittelbar nach dem Kommando (und einer allfälligen Kommandoerweiterung) übertragen (vgl. Kap. 5).

Je nach der Art und Grösse kann ein Dezimalpunkt vorkommen, der dann allerdings nicht übertragen wird (vgl. Anhang Kap. A3).

Bei gewissen Parametern kann anstelle eines negativen Wertes (mit der Kommandoerweiterung "-") auch das Zweierkomplement übertragen werden. (Vgl. Anhang Kap. A2). Um beispielsweise für die Position der negativen Software - Lagegrenze den Wert 321hex Inkremente negativer als der Referenzpunkt anzugeben, übertragen wir als Befehl (vgl. Kap. 4.3) die ASCII-Zeichen "L-321".

Die direkt definierbaren booleschen Parameter (Parameter - Flags) werden mit einer Variante des Kommandos "F" (vgl. Kap. 5) definiert, auf 1 oder auf 0 maskiert. Um beispielsweise die definierte untere Software - Lagegrenze zu aktivieren (ohne ein anderes Parameter - Flag zu beeinflussen), verwenden wir den Befehl: "F+04" ("Flags ored mit 04hex")

Die nicht direkt beeinflussbaren Flags werden durch andere Einwirkungen (wie Zustände innerhalb der Controllers) beeinflusst (vgl. Kap. 5).

Um das Drehmoment am Motor zu entfernen, senden wir den Befehl: "M0".

Es werden folgend nur noch auf diejenigen Parameter eingegangen, die nicht im Kapitel 5 unter den jeweiligen Kommandos ausführlich beschrieben wurden. (Vgl. beispielsweise Kommando "E", "F", und "S").

Beachten Sie, dass in diesem Kapitel allfällige versteckte Dezimalpunkte bei den Hexadezimalwerten angegeben werden, dennoch fehlen diese Dezimalpunkte in der Übertragung! (Vgl. Anhang Kap. 12.3)

6.1 Die Beschleunigung

Die Einheit ist "Änderung der Drehzahl pro Zeiteinheit".

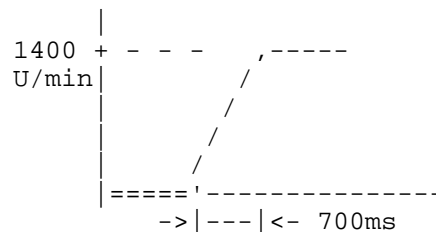
Da die Drehzahl (vgl. unten) in "Inkremente pro Regelzyklus" angegeben wird, entsteht die Beschleunigungseinheit "Inkremente pro Regelzyklus im Quadrat". Der Beschleunigungswert kann Werte von 00.01hex bis zu einem Maximalwert einnehmen. Der Maximalwert berechnet sich aus FF.FEhex abzüglich die momentan programmierte Drehzahl.

Ist die momentan programmierte Drehzahl z.B. FB.70hex, so kann die Beschleunigung keinen Wert grösser als 4.8Ehex annehmen.

Ein Beschleunigungswert von 0.00 wird ebenfalls nicht angenommen.

Berechnungsbeispiel für eine vorgegebene Rampe:

=> Ein Motor muss in einer Zeit von 700 Millisekunden von Null auf 1400 U/min beschleunigen. Der direkt montierte Encoder hat 1000 Impulse pro Umdrehung.



1400U/min --> 95,57 dezimal Inkremente/Regelzyklus
(vgl. Abschnitt "Berechnungsbeispiel für eine vorgegebene Drehzahl", unten)

$$700 \text{ Millisek. sind } \frac{700000 \text{ Mikrosekunden}}{1024 \text{ Mikrosekunden}} = 684 \text{ Regelzyklen (gerundet)}$$

95,57 Inkremente/Regelzyklus in 684 Regelzyklen ergibt

$$\frac{95,57}{684} = 0,140 \text{ dezimal Inkremente/(Regelzyklus)}$$

0.140 muss nun in eine gebrochene Hexadezimalzahl umgewandelt werden (vgl. Anhang Kap. 12.3), es ergibt:

0.24_{hex} Inkremente/(Regelzyklus) .
Der entsprechende Befehl lautet also: "A0024" oder "A24".

6.2 Die Inkremente entlang des Bewegungsweges

Die Position des Systems wird entlang des Bewegungsweges angegeben. Der Bereich umfasst eine 24 Bit mächtige Zahl. Dies ergibt ein Bereich von +/- 8'388'608 Inkremente in Bezug auf die Referenzposition. Dies entspricht einem Weg von 16 Metern, auf einen Tausendstel Millimeter aufgelöst!

Die Geberauflösung und alle im System anfallenden Getriebe- und Übersetzungskonstanten müssen bei der Umrechnung von Inkrementen in eine Wegstrecke oder einen Drehwinkel miteinbezogen werden.

Beispiel für die Berechnung einer Wegstrecke:

Ein Schlitten bewegt sich aufgrund der mechanischen Übersetzungsfaktoren 0.314 mm nach rechts pro positive Motorumdrehung. Ein Encoder mit 250 Impulsen ist an den Motor montiert. Wir wollen eine Position 86,3mm links der Referenzposition anfahren.

$$250 \text{ Impulse} = 4 * 250 \text{ Inkremente} = 1000 \text{ Inkremente}$$

$$1000 \text{ Inkr./Umdr.} = 1000 \text{ Inkr./}0,314\text{mm}$$

$$\text{Das Ziel an der Stelle: } \frac{-86,3 \text{ mm}}{0,314 \text{ mm}} * 1000 \text{ Inkr.} = -274'841 \text{ Inkr. (gerundet)}$$

Die Zahl 274'841 muss nun in eine Hexadezimalzahl umgewandelt werden (vgl. Anhang Kap. 12.1):

ergibt -43199hex. Die Befehlsfolge nach der Referenzierung lautet also "T-43199" und dann "G".

6.3 Die Regelparameter

Auf die Regelparameter wird in Kap. 5, Kommando "C" hingewiesen. In den meisten Fällen wird der Anwender ohne die Kenntnis der Einheiten dieser Regelparameter auskommen. Viel wichtiger ist das allgemeine Verständnis dieser Parameter. Um ein Regelgerät optimal an die gegebenen Lastverhältnisse anzupassen, ist es unumgänglich, falls nicht sehr unkritische Anforderungen bestehen, Grundkenntnisse in Regeltechnik zu besitzen. Unsere Applikationsabteilung kann Sie unterstützen.

Es existieren in der Regeltechnik mehrere empirische Optimierungsmethoden, welche (wie der Name sagt) zum Ziel haben, ein regeltechnisches System mit mehr oder weniger gezieltem Ausprobieren zu optimieren.

Einige allgemeine Aussagen zu den Regelparameter:

- Der KP-Wert (Proportionalfaktor) bestimmt die Steifigkeit des Systems. Ein grösseres KP führt zu einer steiferen Motorwelle. Zuviel KP führt zu Schwingungen. Wenn das Motor - Encoder - System so aufgebaut ist, dass eine ungenügende mechanische Kopplung besteht, kann auch ein kleinerer KP-Wert schon zu Schwingungen führen.
- Der KI-Wert (Integralfaktor) ist dafür verantwortlich, dass das System auch bei dauerndem Störmoment an der Motorwelle die Zielposition exakt findet, dass also keine Regelabweichung bestehen bleibt. Ein grosser KI führt ebenfalls zu Schwingungen, kann mit einem grossen KD kompensiert werden, aber auf Kosten der Dynamik.

- Der KD-Wert (Differentialfaktor) ist für die Trägheit (in einem gewissen Sinne also für die Stabilität) verantwortlich. Ein grosser KD erzeugt mehr Trägheit, er wirkt sich wie eine Wirbelstrombremse aus. Viel KD führt nicht zu Schwingungen, aber das System wird weniger dynamisch, kann also raschen Drehzahländerungen nicht mehr exakt folgen.

→ **Wichtig:** Die Impulszahl des Encoders geht in die Rechnung ein, wenn z.B. ein 1000er-Encoder durch einen 500er-Encoder ersetzt wird, müssen nicht nur die Werte für die Position, Drehzahl und Beschleunigung verdoppelt werden, sondern auch die Werte von KP, KI, KD und der Integrationslimite!

6.4 Der Kopplungsfaktor

Es handelt sich um einen softwaremässig einstellbaren Getriebefaktor, welcher das Drehzahlverhältnis zwischen dem Führungs - Encoder (Indexer) und dem Folge - Encoder bestimmt (vgl. Kap. 5, Kommando "K").

Beispiel zur Berechnung eines Übersetzungsverhältnisses:

```
(Führungs - Encoder) Indexer: 2500 Impulse
(Folge - Encoder):           400 Impulse
```

Gesucht: K - Wert für ein Getriebe. Pro Indexerumdrehung der Folge - Encoder 47 Grad in positive Drehrichtung folgen.

$$\begin{aligned} \text{K-Wert} &= \frac{\text{Impulszahl(Folge-Encoder)} * \text{Winkel am Indexer}}{\text{Impulszahl Indexer} * \text{Winkel (Folge-Encoder)}} = \\ &= \frac{400 * 360 \text{ Grad}}{2500 * 47 \text{ Grad}} = 1,2255 \quad \text{dieser Wert muss nun in} \end{aligned}$$

eine gebrochene Hexadezimalzahl umgewandelt werden (vgl. Anhang Kap. 12.3), es ergibt:

1.3A hex. Der Befehl lautet also "K13A". Da der Wert gerundet wurde, beträgt der effektive Drehwinkel des Folge - Encoders 46,9605... Grad pro Indexerumdrehung.

6.5 Die Drehzahl

Die Einheit ist "Änderung der Position pro Zeiteinheit"

Da die Position in "Inkrementen" angegeben wird, entsteht die Drehzahleinheit "Inkrementen pro Regelzyklus". Der Drehzahlwert (oder der Maximaldrehzahlwert, je nach Regelbetriebsart) kann Werte von 00.00hex bis zu einem Maximalwert einnehmen. Der Maximalwert berechnet sich aus FF.FEhex abzüglich die momentan programmierte Beschleunigung.

Ist die momentan programmierte Beschleunigung z.B. 4.8Ehex, so kann die Drehzahl keinen Wert grösser als FB.70hex annehmen. Der Drehzahlwert von 0.00 wird angenommen und kann verwendet werden, um das System während einer Bewegung (unterwegs) anzuhalten.

7 Bedienelemente und Leiterplattenanschlüsse

Dieses Kapitel enthält alle für den Anschluss und die Einstellung des Controllers wichtigen Informationen. Beschrieben werden die Bedienelemente (Potentiometer, DIP - Switches und die LED), alle Anschlüsse und zuletzt die Funktion der Jumper auf der Leiterplatte des Controllers.

7.1 Bedienelemente (Potentiometer, DIP - Switches und LED)

7.1.1 Potentiometer

#	Name	Beschreibung
P1	Nfast	Wert für die schnelle Drehzahl im Handbetrieb, wenn diese intern vorgegeben wird. Der Einstellbereich umfasst 0...100% der maximal programmierbaren Drehzahl.
P2	Nslow	Wert für die langsame Drehzahl im Handbetrieb, wenn diese intern vorgegeben wird. Der Einstellbereich umfasst 0...6% (1/16) der maximal programmierbaren Drehzahl.
P3	RIOffs	Rated-I Offset. Bei störender Drehrichtungs - Unsymmetrie des Momentes (besonders bei kleinen Y-Werten), bitte abgleichen.

7.1.2 DIP - Switches

#	Name	Beschreibung
S1	DrvAdrbit0	Binär codierte Adresse des Antriebes. In einem Netzwerk darf die gleiche Adresse nicht an mehrere Antriebe vergeben werden. Umschalten im Betrieb ist NICHT erlaubt, nach dem Umschalten Reset durchführen. Die Adresse 0 setzt den entsprechenden Controller beim Reset automatisch in den angesprochen-Zustand (vgl. Kap. 4.9).
S2	DrvAdrbit1	
S3	DrvAdrbit2	
S4	---	nicht verwendet, als DrvAdrbit3 reserviert!
S5	---	nicht verwendet
S6	---	nicht verwendet
S7	CommandedOut	Bestimmt, ob die Digitalausgänge 24c..31c nur durch das Kommando "0" (Stellung "ON") definiert werden, oder ob diese Ausgänge fest zugeordnete Aussagen besitzen, die standardmässig verwaltet werden ("OFF").
S8	SPS-Mode	Bestimmt die Funktion der Digitaleingänge 24c...31c. Stellung "ON" heisst SPS-Modus, die Digitaleingänge haben die Funktion einer parallelen Schnittstelle. Mit "OFF" erlangen die Digitaleingänge die Funktion "Run", "Manual Mode" etc. (Siehe Kapitel 7.3) Bei spezieller zus. Kundensoftware wird statt des SPS-Modus die Kundenspez. Software aktiviert.

7.1.3 LED

#	Farbe	Name	Beschreibung
H1	gn	Supply	Leuchtet, wenn der Regler mit Speisespannung versorgt wird.
H2	rt	Inhibit	Leuchtet, wenn der Motor ohne Drehmoment ist, die Ursache kann ein "M0"-Befehl, ein externer Stop, passivieren des Eingangs "RUN" oder zu hohe Betriebsspannung sein.
H3	gb	Manual Mode	Leuchtet, wenn mit Hilfe des Digital-eingangs "Manual Mode" in den Handbetrieb geschaltet wurde.
H4	gb	AT Ref. Pos.	"At Reference Position" . Leuchtet, wenn der Motor in Zielposition und diese zugleich die Referenzposition ist.
H5	gb	In Position	Leuchtet, wenn der Motor in Zielposition ist (also wenn der Ausgang "In Position" aktiv ist).
H6	rt	Speed Error	Leuchtet, wenn am System während einer Bewegung ein Schleppfehler auftritt (also wenn Ausgang "Speed Error" aktiv ist).
H7	rt	Limit Switch	Leuchtet, wenn einer der beiden Endschalter aktiv ist.
H8	gb	Enc. Signal	Leuchtet, wenn der Encoder Signaländerungen abgibt. Wenn sich der Encoder dreht, blinkt diese LED.

7.2 Detailbeschreibung der Printanschlüsse

(Für die Pegel der Ein- und Ausgänge, vgl. Kap. 8.1 und 8.2)

#	Name	Beschreibung
1a	+15V	+15 Volt Ausgangsspannung. Kann zur Speisung von externen Komponenten (z.B.: 10...30-Volt-Encoder) oder der Digitaleingänge 16c...23c benutzt werden. (Für die Belastung vgl. Kap. 8.3)
2a	0V	0 Volt-Bezugspotential. Dieser Pin ist mit allen anderen 0V-Pins galvanisch verbunden.
3a	Stop-	Externer Stop. Negative Seite des Optokopplereinganges (vgl. Pin 3c).
4a	Limit Fwd.-	Hardware-Endschalter in positiver Richtung. Negative Seite des Optokopplereinganges. Vgl. Pin 4c.
5a	Limit Rev.-	Hardware-Endschalter in negativer Richtung. Negative Seite des Optokopplereinganges. Vgl. Pin 5c.
6a	Referenz-	Vorreferenz. Negative Seite des Optokopplereinganges. Vgl. Pin 6c.
7a	(Enc2 Z)	nicht benutzt
8a	Encoder2 A	A-Kanal (U1-Kanal) des Indexers. Die Polarität ist aktiv high
9a	Encoder2 B	B-Kanal (U2-Kanal) des Indexers. Die Polarität ist aktiv high.
10a	Encoder2 +V	Speisepin für den Indexer. Üblicherweise + 5 Volt, kann intern abgetrennt und als Schlaufenpunkt für eine andere Speisespannung gebraucht werden. (Vgl. Jumper J10)
11a	0V	0 Volt-Bezugspotential. Dieser Pin ist mit allen anderen 0V-Pins galvanisch verbunden.
12a	TX\	RS232: nicht benutzt RS422/RS485: Invertierte Leitung des Sendeleiterpaares. Kann mit RX\ verbunden werden. (Vgl. Kap. 4.5 und 4.6) Achtung: Leitung hat keine Abschlusswiderstände.
13a	RX\	RS232: nicht benutzt RS422/RS485: Invertierter Eingang Kann mit TX\ verbunden werden. (Vgl. Kap. 4.5 und 4.6) Achtung: Leitung hat keine Abschlusswiderstände.

#	Name	Beschreibung
14a	+5V	5 Volt Speisespannung als Ausgang. Kann zur Speisung von externen Komponenten (z.B.: Potentiometer) gebraucht werden. Für die Belastung vgl. Kap. 8.3.
15a	F0V	0 Volt Bezugspotential. Dieser Pin ist mit allen anderen 0V-Pins galvanisch verbunden, kann jedoch unterbrochen (Jumper J13) und extern verdrahtet werden. "F" bedeutet "floating, dieser Pin ist die Rückleitung der Eingangsfilter der digitalen Eingänge 16c...23c (vgl. Jumper J13).
16a	(STR\)	(nicht benutzter Ausgang, nicht verdrahten !)
17a	N-fast ext.	Analogeingang. Pin zum Schleifer des externen Potentiometers "Eilgang extern" (vgl. Pin 20c)
18a	N-slow ext.	Analogeingang. Pin zum Schleifer des externen Potentiometer "Schleichgang extern" (vgl. Pin 20c)
19a	Rated I	Analogausgang (max: +-10 Volt bezüglich 0 Volt). Stellgrösse des digitalen PID - Lagereglers. Führungsgrösse des analogen Stromreglers. +5 Volt = 100% statischer Nennstrom (vgl. Kap. 3.11)
20a	0V	0 Volt Bezugspotential. Dieser Pin ist mit allen anderen 0V-Pins galvanisch verbunden.
21a	---	nicht benutzt
22a	---	nicht benutzt
23a	HWR\	Hardware-Reset. Ein- oder Ausgang. Geht auf log. 0, wenn Reset stattfindet oder löst Reset aus, wenn nach log.0 gezogen. Nach einer Aktivphase an diesem Pin, ist der Controller im gleichen Zustand wie nach Einschalten der Speisespannung.
24a	(ACK\)	nicht benutzt
25a	---	nicht benutzt
26a	---	nicht benutzt
27a	---	nicht benutzt
28a	---	nicht benutzt
29a	---	nicht benutzt
30a	---	nicht benutzt

#	Name	Beschreibung
31a	+AO	Speisepin für die Treiberspannung der Digitalausgänge Pin 24c...31c, falls der Controller mit treibern bestückt ist (Standard). Als Option "Source" kann der Controller mit Open-Kollektor-Treibern (Sink-driver) bestückt werden. Dann ist dieser Pin der Zusammenschluss aller Freilaufdioden und sollte mit der höchsten Schaltspannung aller Lasten verbunden werden (vgl. Kap. 8.2).
32a	MM0V	Gehäusemasse (Schutzerde) - üblicherweise nicht mit 0 Volt verbunden (vgl. Jumper J1).
1c	+15V	wie Pin 1a.
2c	0V	wie Pin 2a.
3c	Stop+	Externer Stop. Positive Seite des Optokopplereingangs (vgl. Pin 3a). Je nach Zustand des Parameterflags "ESWPOL", muss am Pinpaar 3c-3a Spannung oder keine Spannung anliegen, um den Optokopplereingang zu aktivieren. Bei "ESWPOL" = 0 ist: Stromführender Zustand = externer Stop passiv -> Drahtbruchsicher! (Vgl. Kap. 3.5)
4c	Limit Fwd.+	Hardware-Endschalter in positiver Richtung. Positive Seite des Optokopplereingangs (vgl. Pin 4a). Das unter Pin 3c gesagte gilt entsprechend (vgl. Kap. 3.5).
5c	Limit Rev.+	Hardware-Endschalter in negativer Richtung. Positive Seite des Optokopplereingangs (vgl. Pin 5a). Das unter Pin 3c gesagte gilt entsprechend (vgl. Kap. 3.6).
6c	Referenz+	Vorreferenz. Positive Seite des Optokopplereingangs (vgl. Pin 6a). Das bei Pin 3c gesagte gilt entsprechend (vgl. Kap. 3.4).
7c	Encoder1 Z	Z-Kanal (0-Kanal) des System-Encoders. Die Polarität ist aktiv high.
8c	Encoder1 A	A-Kanal (U1-Kanal) des System-Encoders. Die Polarität ist aktiv high.
9c	Encoder1 B	B-Kanal (U2-Kanal) des System-Encoders. Die Polarität ist aktiv high
10c	Encoder1 +V	wie Pin 10a.
11c	0V	0 Volt Bezugspotential. Dieser Pin ist mit allen anderen 0V-Pins galvanisch verbunden.

#	Name	Beschreibung
12c	TX	RS232: Sendeleitung für Daten vom Regelgerät zum PC. RS422/RS485: Nichtinvertierte Leitung des Sendeleiterpaares. Kann mit RX verbunden werden. (Vgl. Kap. 4.5 und 4.6) Achtung: Leitung hat keine Abschlusswiderstände.
13c	RX	RS232: Empfangsleitung für Daten vom PC zum Regler RS422/RS485: Nichtinvertierter Eingang. Kann mit TX verbunden werden. (Vgl. Kap. 4.5 und 4.6) Achtung: Leitung hat keine Abschlusswiderstände.
14c	+5V	wie Pin 14a.
15c	F0V	wie Pin 15a.
16c	Run	PC-Betrieb: Endstufen-Freigabe. Schaltet das Drehmoment am Motor an (aktiv) oder ab (passiv). (Vgl. Kap. 7.3)
	-oder- SPSData0	SPS-Betrieb: SPS-Datenleitung 0 (niederwertigstes Bit)
17c	Manual Mode	PC-Betrieb: Handbetrieb aktivieren. (Vgl. Kap. 7.3)
	-oder- SPSData1	SPS-Betrieb: SPS-Datenleitung 1
18c	Forward	Handbetrieb: Drehzahl im Handbetrieb ist positiv. (Vgl. Kap. 7.3)
	-oder- [SPSData2]	SPS-Betrieb: SPS-Datenleitung 2
19c	Reverse	Handbetrieb: Drehzahl im Handbetrieb ist negativ. (Vgl. Kap. 7.3)
	-oder- [SPSData3]	SPS-Betrieb: SPS-Datenleitung 3
20c	Slow / Fast\	Handbetrieb: Im Handbetrieb wird grosse (aktiv) bzw. kleine (passiv) Drehzahl gefahren. (Vgl. Kap. 7.3)
	-oder- [SPSData4]	SPS-Betrieb: SPS-Datenleitung 4
21c	Home	PC-Betrieb: Referenzierung auslösen, oder wenn schon referenziert worden ist: Fahrt an die Referenzposition (Position 0). (Vgl. Kap. 7.3)
	-oder- [SPSData5]	Handbetrieb: Drehzahlvorgabe ist intern (aktiv) oder extern (passiv). (Vgl. Kap. 7.3) SPS-Betrieb: SPS-Datenleitung 5
22c	Go	PC-Betrieb: Externer Start einer Bewegung. (Vgl. Kap. 7.3)
	-oder- [SPSData6]	SPS-Betrieb: SPS-Datenleitung 6 (höchstwertiges Bit)

#	Name	Beschreibung
23c	Capture	Laden des B-Registers mit der aktuellen Position (siehe B-Befehl)
	SPSSTR	SPS-Modus: SPS-Strobe- oder Taktleitung. Jede passiv-aktiv-Flanke lädt die 7 Bits an den SPS-Datenleitungen SPSData0..6 als ASCII-Zeichen in den Controller. (Vgl. Kap. 4.7)
24c	Ready	"Bereit"-Meldung vom Controller. (aktiv = bereit)
25c	In Position	System steht in der Zielposition und zwar innerhalb einer programmierbaren Toleranzbreite. (Vgl. Kap. 3.10) (aktiv = KEIN Lagefehler).
26c	Pos Error/	Kein Schleppfehler während einer Bewegung und kein Positionsfehler während einer abgeschlossenen Bewegung. (Passiv entspr. alles in Ordnung).
27c	Limit Error	Hinterer- oder Vorderer-Hardware-Endschalter ist aktiv. (Vgl. Kap 3.5) (aktiv = Endschalter aktiv)
28c	Torque	Sobald das Moment am Motor eingeschalten ist, ist der Ausgang aktiv.
29c	Trigger	Aktiv, wenn Triggerpunkt entweder deaktiviert oder Überfahren (und daher wieder deaktiviert) ist. Passiv, wenn der Triggerpunkt aktiviert wurde, aber noch nicht überfahren worden ist.
30c	Trigger Dir	Aktiv zeigt, dass sich die Istposition auf der negativen, bei passiv auf der positiven Seite des Triggerpunktes befindet. -> Es spielt keine Rolle, ob der Triggerpunkt aktiv oder passiv ist.
31c	SPSACK	SPS-Acknowledge-Leitung. Ein Übergang von passiv nach aktiv unmittelbar nach einem gesendeten Kontrollsummenzeichen bestätigt die richtige Übertragung und eine gewisse Richtigkeit des soeben gesendeten Befehls. Wird mit einem gesendeten STX wieder passiviert. (Vgl. Kap. 4.7)
32c	MM0V	wie Pin 32a

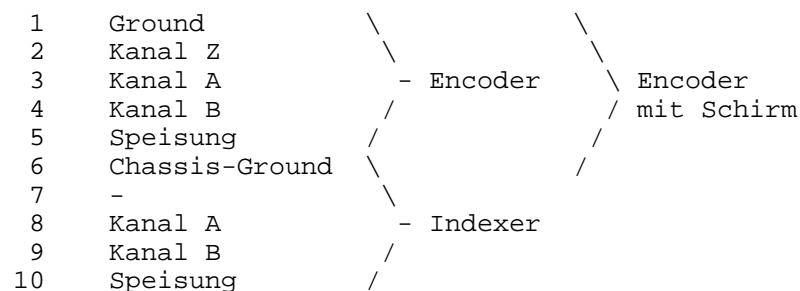
Pinbelegung

	c	a
+15V	o 1	o +15V
0V	o 2	o 0V
Stop +	o 3	o Stop -
Limit Fwd. +	o 4	o Limit Fwd. -
Limit Rev. +	o 5	o Limit Fwd. -
Referenz +	o 6	o Referenz -
Encoder1 Z	o 7	o ---
Encoder1 A	o 8	o Encoder2 A
Encoder1 B	o 9	o Encoder2 B
Encoder1 +V	o 10	o Encoder2 +V
0V	o 11	o 0V
TX	o 12	o TX\
RX	o 13	o RX\
+5V	o 14	o +5V
F0V	o 15	o F0V
SPSData0 / Run	o 16	o (STR\)
SPSData1 / Manual Mode	o 17	o N-fast ext.
SPSData2 / Forward	o 18	o N-slow ext.
SPSData3 / Reverse	o 19	o Rated I
SPSData4 / Slow/Fast\	o 20	o 0V
SPSData5 / Home	o 21	o ---
SPSData6 / Go	o 22	o ---
SPSStr	o 23	o HWR\
Ready	o 24	o (Ack\)
In Pos	o 25	o ---
Pos Err\	o 26	o ---
Lim Err	o 27	o ---
Torque	o 28	o ---
Trigger	o 29	o ---
Trigger Dir	o 30	o ---
SPS Ack	o 31	o +AO
MM0V	o 32	o MM0V

X5 - Pinbelegung

X5 dient dem Anschluss von einem oder beiden Encoder.

Mit der Verbindung von Ground mit Chassis - Ground (J1) werden die Anschlüsse von Pin 1 bis 5 zum Encoder gleichwertig zu den Anschlüssen 6..10 zum Indexer. Für Systeme mit nur einem Geber reichen Pin 1..6 aus, um ein Encoder mit getrennter Schirmung anzuschliessen.



7.3 Digitaleingänge im PC-Betrieb, Funktionen

Run: Kontrolliert das Drehmoment am Motor. Aktivieren schaltet Drehmoment ein, passivieren schaltet Drehmoment aus. Zugleich kann mit dem Befehl "M1" oder "M2" das Drehmoment NICHT eingeschaltet werden, wenn Run passiv ist. Daher bei Nichtgebrauch Run immer auf aktiv verdrahten.

Wenn das Drehmoment mit dem Befehl "M0" oder aufgrund des externen Stops entfernt wird, muss Run kurz passiv und dann wieder aktiv geschaltet werden, um das Drehmoment wieder einzuschalten (oder Befehl "M1" bzw. "M2" senden).

Wurde eine Bewegung vor dem Erreichen der Zielposition abgebrochen, wird sie durch Aktivieren von Run nicht automatisch wieder fortgesetzt, dazu braucht es zusätzlich ein Go - Kommando. (Ausnahme: Im Handbetrieb kann eine Drehzahl gestoppt und ohne Go wieder eingestellt werden.)

Wenn Run aktiviert wird, wird automatisch die aktuelle Position als Zielposition gesetzt. Eine programmierte Zielposition (einer unterbrochenen Bewegung) wird dabei nicht überschrieben und kann mit Go sofort erneut angefahren werden.

Manual Mode: Schaltet den Handbetrieb ein (vgl. Kap. 3.1 und 3.6). Mit Aktivieren des Handbetriebes, bekommen die Digitaleingänge Forward, Reverse, Slow / Fast\ und Home eine Funktion, welche sie nur im Handbetrieb besitzen. (Vgl. Bemerkungen, am Ende dieses Kapitels)

Forward: Funktion nur im Handbetrieb. Die von den internen Potentiometern oder von den Analogeingängen kommende Drehzahlinformation führt zu einer positiven Drehzahl.

→ Bewegung in positive Richtung. (Vgl. Bemerkungen, am Ende dieses Kapitels)

Reverse: Funktion nur im Handbetrieb. Die von den internen Potentiometern oder von den Analogeingängen kommende Drehzahlinformation führt zu einer negativen Drehzahl.

→ Bewegung in negative Richtung. (Vgl. Bemerkungen, am Ende dieses Kapitels)

Slow / Fast\ : Funktion nur im Handbetrieb. Von den, von den internen Potentiometern oder von den Analogeingängen kommende Drehzahlinformation, wird die kleine Drehzahl (aktiv) oder die grosse Drehzahl (passiv) ausgewählt. (Vgl. Kap. 3.6)

Home: Funktion im SPS-Betrieb: Eine passiv - aktiv - Flanke führt einen "Z5"-Befehl aus (vgl. Kap. 5, Kommando "Z"), falls die Referenz ungültig (d.h.: noch nicht definiert oder verschoben) ist. Falls die Referenz gültig ist (d.h.: Referenzierung schon ausgeführt), wird eine Bewegung an die Position Null ausgeführt. Funktion im Handbetrieb: Entscheidet, ob die Drehzahlinformation von den zwei Analogeingängen (passiv) oder vom internen Potentiometerpaar P1, P2 (aktiv) kommt.

Go: Hat die gleiche Wirkung wie das Senden des Kommandos "G". Die effektive Wirkung ist von der Vorgeschichte abhängig. Je nach den vorangehend geladenen Befehlen, kann damit eine Bewegung an eine absolute Zielposition oder ein Sprung ausgelöst werden. (Vgl. Kap. 5 Kommando "G") Bei den Sprüngen ist wichtig zu wissen, dass jede passiv - aktiv - Flanke den Sprung einmal ausführt! Es handelt sich dabei nicht um eine Zwischenspeicherung von Sprüngen in irgendeinem Befehlspuffer im Controller, sondern um die Verlängerung eines einzigen Sprunges.

Bemerkungen:

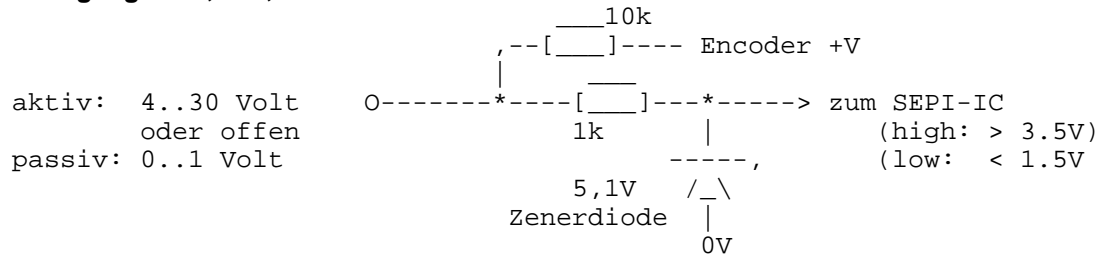
- Bei gleichzeitiger Aktivierung von Forward und Reverse hat Forward Vorrang. Mit aktivem Reverse kann also allein die Stellung von Forward über die Drehrichtung bestimmen.
- Wenn weder Forward noch Reverse aktiv ist, bleibt das System im Handbetrieb und in Lageregelung, ein allfällig zugeschalteter Indexer hätte keinen Einfluss auf die Position. (Vgl. Kap. 3.15)
- Wird durch Aktivieren oder Passivieren von Forward, Reverse, Slow / Fast\ oder Home eine Drehzahländerung eingeleitet, geschieht diese mit der momentan programmierten Beschleunigung (vgl. Kap. 3.8 und Kommando "A" in Kap. 5)
- Wird der Handbetrieb während der Fahrt verlassen, setzt der Controller die Drehzahl sofort (ohne Rampe) auf Null! Wir empfehlen deshalb den Handbetrieb immer im Stillstand zu verlassen.

8 Externe Beschaltung

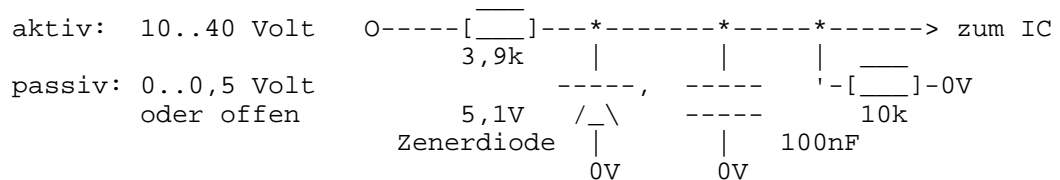
Dieses Kapitel beschreibt die Ein- und Ausgangsschaltungen des Motion - Controllers und ordnet die verwendeten Pegelbezeichnungen "aktiv" und "passiv" effektiven Spannungsbereichen zu. Die Belastbarkeit der Speisung wird beschrieben, so dass der Anwender seine externen Komponenten entsprechend auswählen oder eine eigene externe Speisung entsprechend konzipieren kann.

8.1 Eingangsschaltungen und Eingangspegel

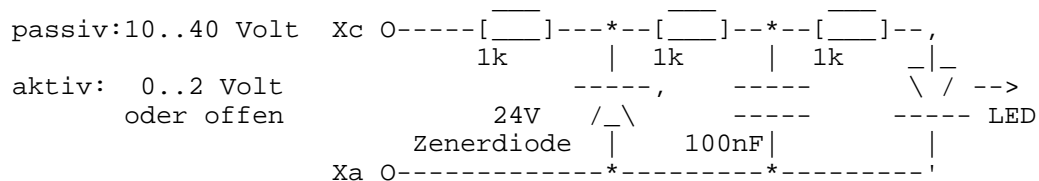
Impulsgeber - Eingänge 7ac, 8ac, 9ac:



Digitaleingänge 16c..23c:



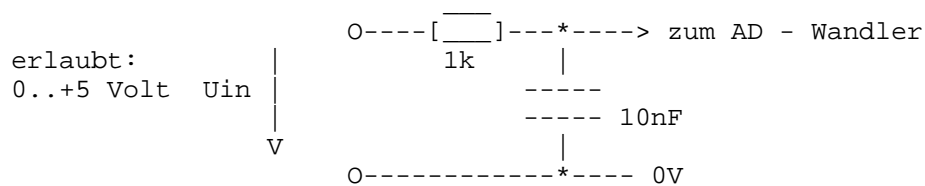
Optokopplereingänge 3ac, 4ac, 5ac, 6ac:



→ Die Angaben "aktiv" und "passiv" gelten für einen Öffner-Schalter, d.h. das Parameterflag Dii = 00

→ Um sicher anzusprechen, muss der entsprechende Pegel für mindestens 2 Millisekunden anliegen!

Analogeingänge: 17a, 18a

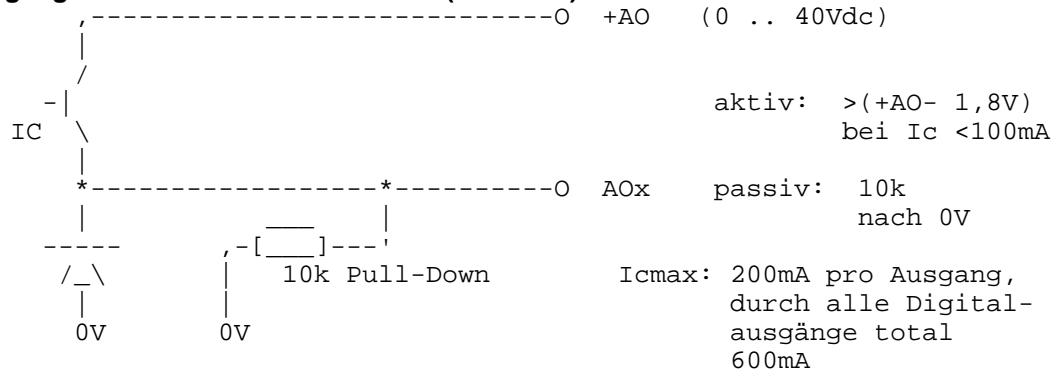


Uin	entspricht als hohe Drehzahl	entspricht als tiefe Drehzahl	ergibt beim Lesen mit Kommando "I"
5,0V	FFFFhex 100%	0FFFhex ~6%	FFhex
2,5V	7FFFhex 50%	07FFhex ~3%	7Fhex
0V	0hex 0%	0% ^	00hex

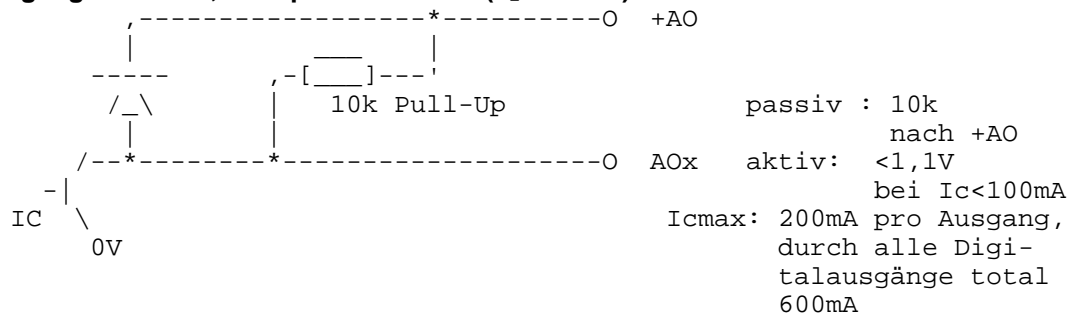
in % der maximal programmierbaren Drehzahl

8.2 Ausgangsschaltungen und Ausgangspegel

Digitale Ausgänge 24c..31c als Source - Driver (Standard):



Digitale Ausgänge 24c..31c, als Open - Collector (optional):



8.3 Externe Belastung der Speisung

Im Controller kann:

- an den Pin 1a und 1c +15 Volt gegen 0V und
- an den Pin 14a und 14c +5 Volt gegen 0V abgegriffen werden.

Als Belastung eines Controllers ist zulässig: max. 90 mA der +15 Volt ODER max. 200 mA der + 5 Volt

$$\text{Allgemein : } \boxed{I_{15} + 0,45 * I_5 \leq 90 \text{ mA}}$$

wobei: I_{15} : Strom ab der +15 Volt Speisung
 I_5 : Strom ab der + 5 Volt Speisung

Achtung:

Der Controller verfügt über keine Vorrichtung, die darauf aufmerksam macht, dass die Belastung zu gross ist. Insbesondere muss Kurzschluss oder Fast - Kurzschluss unbedingt vermieden werden!

Beachten Sie:

Die totale Belastung an den Speisungen muss berücksichtigt werden. Also Encoderspeisung, Speisung der Digital- und Optokopplereingänge etc!

9 Inbetriebnahme

9.1 Einrichten

Montage und Anschluss:

- Schliessen Sie den Controller genau nach Schema an. (Kap. 7, 8 und 10 kann Ihnen dabei helfen)
- Gewisse Leitungen müssen, speziell in industrieller Umgebung, verdrillt, abgeschirmt oder speziell verlegt werden. Dieses Handbuch gibt an entsprechender Stelle Auskunft.
- Beachten Sie, dass die Leistungsdaten der Controller nur bei senkrechter Montage (wie in einem Rack) und genügend Lüftung gewährleistet sind. Geräte mit "F" in der Typenbezeichnung müssen mit einem Fremdlüfter versehen werden.

Vorbereitung:

- Kontrolle der ganzen Verdrahtung

Einschalten:

- Wenn möglich sollte der Controller zuerst ohne Motor betrieben werden. Dazu wird die Ankerleitung unterbrochen. Sobald die Betriebsspannung an das System angelegt wird, kann über die serielle Schnittstelle kommuniziert werden. Verwenden Sie dazu z.B. das Demonstrationsprogramm (Kap. 1.3). Wenn die Verbindung nicht zum Stehen kommt, kann es ev. am verwendeten Kabel (gewisse PCs müssen RTS und CTS verbunden haben), an der Baudrate oder am Typ der Schnittstelle liegen (RS232 nicht mit RS422/485 mischen!).
- Sobald der Motor zugeschaltet wird, sollte mit dem Strompotentiometer (auf dem Motor - Driver oder dem Zwischenprint) der Ankerstrom so klein gemacht werden, dass das System gerade noch zu drehen imstande ist. Geben Sie nun mit dem Befehl "M1" oder dem Digitaleingang Run (nur im PC-Modus) Drehmoment an den Motor. Falls der Motor nun nicht positionsgeregelt ist, sondern auf eine Seite wegfährt, ist sehr wahrscheinlich der Encoder oder der Motor verkehrt herum angeschlossen. Sie können nun die beiden Motoranschlüsse oder die Encoderkanäle A und B vertauschen. Beachten Sie, dass das Letztere die Bezugsrichtung des Systems umkehrt.
- Stellen Sie jetzt, oder erst nach den ersten Probebewegungen, den Ankerstrom auf seinen vorgesehenen Wert ein.

9.2 Erste Bewegungen

Jetzt geht's los:

- Der Antrieb ist nach einem M1-Befehl in Positionsregelung und erwartet weitere Befehle:
- Geben Sie eine Zielposition ein. Im Demoprogramm (vgl. Kap. 1.3) ganz Einfach mit "T4000" <Return>
- Danach starten Sie die Bewegung mit "G" <Return> der Motor sollte nun mit der Defaultbeschleunigung und der Defaultdrehzahl eine Bewegung von 4000hex Inkrementen (das sind 4096 IMPULSE) ausführen.
- Falls die Bezugsrichtung falsch herum liegt, müssen Sie Encoderkanal A mit B und zusätzlich die Ankeranschlüsse vertauschen.
- Sobald Sie den Ankerstrom auf seinen vorgesehenen Wert eingestellt haben (mit Amperemeter im Ankerkreis des belasteten Motors), können Sie die Regelparameter einstellen. (vgl. Kap. 3.9 und speziell Kap. 6)

Beachten Sie:

- Der Controller ist nach dem Einschalten der Betriebsspannung grundsätzlich OHNE Drehmoment am Motor. Sie müssen als erste Aktion mit dem Befehl M1, M2 oder M3 (oder dem Digitaleingang Run) das Drehmoment einschalten.
- Die Kommunikation über die serielle Schnittstelle ist immer möglich, es sei denn, die Netzwerkadresse ist falsch eingestellt.
- Wenn Sie Verbindung zum Controller haben, aber Ihre Befehle keine Wirkung zeigen, lesen Sie das Fehlerbyte, es zeigt Ihnen woran's liegt.

10 Anwendungsbeispiele

Dieses Kapitel beschreibt anhand einiger Beispiele gegebene Problemstellungen aus der Praxis und deren Lösung mit dem Controller.

10.1 Einachsige Vorschubeinheit

(Dieses Beispiel wird von unserer Applikationsabteilung noch speziell bearbeitet. In Kapitel 6 haben Sie Berechnungsbeispiele zu einigen kleineren Problemstellungen).

11 Tipps und Tricks

11.1 Häufig gestellte Fragen

- Ich gebe einen "Go" - Befehl, aber der Antrieb bewegt sich nicht, obwohl die Verdrahtung und die Speisespannung stimmt und ich weder ein "NAK" - Zeichen empfangen, noch ein Fehlerflag gesetzt wird.
 - Leuchtet die Supply - LED H1 (DIP-Switch 7 auf off)? Falls nicht, müssen sie die Speisung in Ordnung bringen.
 - Haben sie Moment am Motor? (Siehe F - Flags: FlyMod, VelMod, Jump, PosBlk, NegBlk). (Einschalten des Momentes mit M1, M2 oder M3).
 - Ist kein Fehler aktiv (E-Flags). Achtung: DrvErr wird nur angezeigt wenn die Endstufe aktiviert ist (Run-Signal, nicht M0)
 - Eventuell ist die Drehzahl auf Null programmiert. Geben Sie eine Drehzahl ungleich Null und dann wiederum "Go".
 - Es könnte auch sein, dass der Y-Wert zu klein gewählt sind, dass der Motor das Reibungsmoment nicht aufbringen kann.
 - Im Positionsbetrieb fährt der Controller erst zur nächsten Position, wenn er „im Ziel“ ist.
- Im SPS-Betrieb erhalte ich auch eine Flanke des SPS - ACK - Signales, wenn der Befehl über die serielle Schnittstelle kommt - Aber nicht immer, warum das?
 - Die Digitalausgänge werden nur alle 8 Millisekunden bearbeitet. Sehr kurze Protokolle, wie z.B.: "Go" (4 ASCII-Zeichen mit dem Übertragungsrahmen) machen sich daher u.U. am Digitalausgang SPS-ACK gar nicht bemerkbar. Obwohl das SPS - ACK - Signal eigentlich nur für die Übertragung via SPS - Schnittstelle gedacht ist, kann z.B. ein Go - Befehl verlängert werden, so dass er ebenfalls eine Flanke erzeugt, wenn er via serielle Schnittstelle übertragen wird. (Also statt "G" wird "G000000" übertragen)
- Kann der Controller auch als (über eine Schnittstelle) programmierbare Spannungsquelle verwendet werden?
 - Verwenden Sie das Signal "Rated I" als Ausgangsspannung. Schliessen sie keinen Encoder am Controller an. Betreiben Sie den Regler nun im reinen P - Betrieb ("C" - Befehl: KP und KD gleich Null). Nun können Sie mit Eingabe eines Zieles ("T" - Befehl) eine Ablage erzeugen, deren proportionales Signal stabil am Ausgang "Rated I" ansteht.
- Wie kann ich den Antrieb bewegen, wenn ein Endschalter aktiv ist?
 - Normalerweise sollten sie Rückwärts wieder herausfahren können. Wurde jedoch auch der andere Endschalter irgendwie aktiv (der hintere Schalter beim Vorwärtsfahren), werden beide, d.h. die negative und die positive Blockierung gesetzt. Löschen Sie nun die Blockierungsflags (R-Register), kann mit einem erneuten "Go" eine Bewegung ausgelöst werden.
ACHTUNG: Auch in die falsche Richtung!

12 Anhang

12.1 Was ist "hexadezimal"?

Unser Zahlensystem im täglichen Gebrauch basiert auf der Zahl 10, es hat somit auch 10 Ziffern. Die Wertigkeit jeder Ziffer kann als 10er-Potenz angesehen werden:

Bsp: Zahl 47052 dezimal:

10er-Wertigkeit: $10^4=10000$ $10^3=1000$ $10^2=100$ $10^1=10$ $10^0=1$

Beispiel:	4	7	0	5	2
bedeutet:	$4 \cdot 10^4$	$+ 7 \cdot 10^3$	$+ 0 \cdot 10^2$	$+ 5 \cdot 10^1$	$+ 2 \cdot 10^0$
also:	40000	+ 7000	+ 000	+ 50	+ 2

Das verwendete Zahlensystem basiert auf der Zahl 16. Dieses System hat 16 Ziffern. Das 16er-System oder Hexadezimalsystem, wie es auch genannt wird, bringt erhebliche Vorteile bei der Verarbeitung im Controller.

Die 16 Ziffern sind "0" ... "9" mit dem gleichen Wert wie im Zehnersystem, dann geht es weiter mit "A" (Wert 10), "B" (11), "C" (12), "D" (13), "E" (14) und "F" (15).

Die Wertigkeit kann als Potenz der Zahl 16 angesehen werden:

Beispiel: Zahl 9EA06 hex

16er-Wertigkeit: $16^4=65536$ $16^3=4096$ $16^2=256$ $16^1=16$ $16^0=1$

Beispiel:	9	E	A	0	6
bedeutet:	$9 \cdot 16^4$	$+ 14 \cdot 16^3$	$+ 10 \cdot 16^2$	$+ 0 \cdot 16^1$	$+ 6 \cdot 16^0$
also:	9 * 65536	+ 14 * 4096	+ 10 * 256	+ 0 * 16	+ 6 * 1

wir schreiben hinter die Zahl "hex", um zu sagen, dass es sich dabei um eine Hexadezimalzahl handelt.

9EA06hex lässt sich natürlich in die entsprechende Dezimalzahl umrechnen, indem man die obige Summation der Produkte durchführt. Es ergibt: 649734. In dieser Art lässt sich jede Hexadezimalzahl in die entsprechende Dezimalzahl umwandeln.

Die Umwandlung von Dezimalzahlen in Hexadezimalzahlen erfolgt in einer etwas anderen Art: Man versucht die gegebene Dezimalzahl in einer Summe von möglichst grossen Potenzen der Zahl 16 darzustellen:

Die Potenzen von 16 sind:

16^5	=	1048576
16^4	=	65536
16^3	=	4096
16^2	=	256
16^1	=	16
16^0	=	1

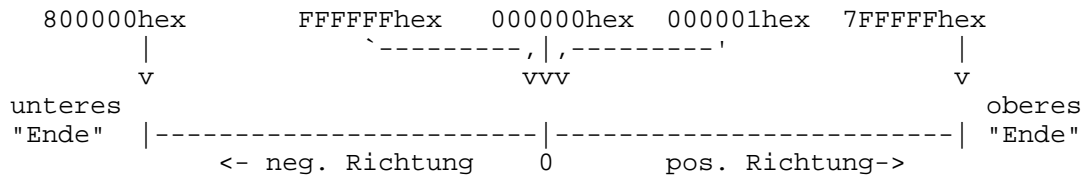
Die Dezimalzahl	2355274	also ist	2 * 1048576	→	"2"
übrig bleibt	258122	das ist	3 * 65536	→	"3"
übrig bleibt	61514	das ist	15 * 4096	→	"F"
übrig bleibt	74	das ist	0 * 256	→	"0"
übrig bleibt	74	das ist	4 * 16	→	"4"
übrig bleibt	10	das ist	10 * 1	→	"A"

Werden die Faktoren fortlaufend dargestellt, ergibt sich dieselbe Zahl in Hexadezimaldarstellung: 23F04Ahex. (Die Demodiskette beinhaltet ein Programm zur Berechnung solcher Hexadezimalzahlen).

12.2 Hexadezimale Zahlengerade, negative Hexadezimalzahlen

Auch die Hexadezimalzahlen lassen sich auf einer Zahlengerade darstellen. Das Bild entspricht dabei genau den Positionen des Controller - Massstabes entlang der Bewegungsachse des Systems.

Wir betrachten einen Zahlenbereich von 24 Bit, das entspricht dem Bereich einer 6-stelligen Hexadezimalzahl:



Man beachte:

Das vorderste Bit (Bit 23) ist das Vorzeichen, die "Enden" berühren sich genau genommen, vom positivsten Wert 7FFFFFFhex ist es nur ein Schritt in die positive Richtung weiter, um zum negativsten Wert 800000hex zu gelangen.

Bei allen Positionsangaben im Controller kann mit einem negativen Vorzeichen die Zahlengerade am Nullpunkt gespiegelt werden. Statt FFFFFFFhex, kann man also -000001hex, statt FFFFFFFhex also -000002hex eingeben.

12.3 Parameter mit verstecktem Dezimalpunkt

Genauso, wie im Zehnersystem die Zehnerpotenzen nach unten erweitert werden können, um gebrochene Werte anzugeben, kann das Gleiche auch im Hexadezimalsystem gemacht werden. Wir erhalten dann eine gebrochene Hexadezimalzahl.

Der Dezimalpunkt (genauso wie das Komma im Zehnersystem) ist dabei lediglich ein Zeichen, das anzeigt, wo sich die Einerstelle befindet, nämlich unmittelbar links von diesem Zeichen.

Anstelle der Zehnerpotenzen, werden im Nachkommabereich der Hexadezimalzahlen natürlich die Potenzen von 16 verwendet:

Bsp. 53,903 im Zehnersystem

10er-Wertigkeit: $10^1=10$ $10^0=1$ $10^{-1}=1/10$ $10^{-2}=1/100$ $10^{-3}=1/1000$

Beispiel:	5	3	9	0	3
bedeutet:	$5 \cdot 10^1$	$+ 3 \cdot 10^0$	$+ 9 \cdot 10^{-1}$	$+ 0 \cdot 10^{-2}$	$+ 3 \cdot 10^{-3}$
also:	50	+ 3	+ 0.9	+ 0.00	+ 0.003

Und E3.D2B im Hexadezimalsystem:

16er-Wertigkeit: $16^1=16$ $16^0=1$ $16^{-1}=1/16$ $16^{-2}=1/256$ $16^{-3}=1/4096$

Beispiel:	E	3	D	2	B
bedeutet:	$14 \cdot 16^1$	$+ 3 \cdot 16^0$	$+ 13 \cdot 16^{-1}$	$+ 2 \cdot 16^{-2}$	$+ 11 \cdot 16^{-3}$
also:	$14 \cdot 16$	$+ 3 \cdot 1$	$+ 13 \cdot 0.0625$	$+ 2 \cdot 0.0039..$	$+ 11 \cdot 0.0002..$

E3.D2Bhex lässt sich natürlich in die entsprechende Dezimalzahl umrechnen, indem man die obige Summation der Produkte durchführt, es ergibt: 227,823242....

In dieser Art lässt sich jede Hexadezimalzahl in die entsprechende Dezimalzahl umwandeln.

Da im Zusammenhang mit dem Controller nur Hexadezimalzahlen mit maximal 2 Nachkommastellen vorkommen (als Parameter z.B.: des Kommandos A, V oder K), werden wir die Umrechnung etwas vereinfachen.

Die Hexadezimalzahl 3A.6Dhex z.B., lässt sich wesentlich einfacher in 2 getrennten Schritten in die entsprechende Zahl im Zehnersystem umwandeln:

Die Stellen VOR dem Dezimalpunkt werden gemäss Kap. 12.1 umgewandelt --> 3Ahex => 58
Die Stellen HINTER dem Dezimalpunkt werden ebenfalls gemäss Kap. 12.1 umgewandelt, und zwar OHNE Rücksicht auf ihren Stellenwert: --> 6Dhex => 109
Da wir wissen, dass es sich bei der zweiten Stelle nach Dezimalpunkt um 256-stel handelt, wird die soeben erhaltene Zahl 109 mit 1/256 multipliziert und wir erhalten das richtige Resultat der Nachkommastellen exakt: $109 / 256$ oder 0,4257...
Nun rechnen wir noch den Wert der Vorkommastellen dazu: $3A,6Dhex = 58 \frac{109}{256}$ oder 58,4257...

Die Umwandlung von Dezimalzahlen in Hexadezimalzahlen erfolgt in der gleichen Art aber zeitlich umgekehrt:

Die Dezimalzahl 60,3082 teilen wir zuerst auf:

60 wird getrennt in hexadezimal umgewandelt (Kap. 12.1) --> 60 => 3Chex
Den Nachkommawert versuchen wir möglichst genau als Bruch mit Nenner 256 und ganzzahligem Zähler darzustellen: $0,3082 = \frac{79}{256}$ (0,3082 mit 256 multiplizieren und Ergebnis auf ganze Zahl runden)
Diesen Zähler wandeln wir nun gem. Kap. 12.1 in eine Hexadezimalzahl um: 79 = 4Fhex
Nun setzen wir die zwei Teilresultate zusammen: $60,3082 = 3C,4Fhex$

Man beachte:

Die Dezimalpunkte werden zur Übertragung wieder entfernt, die obige Zahl würde also als "3C4F" übertragen.

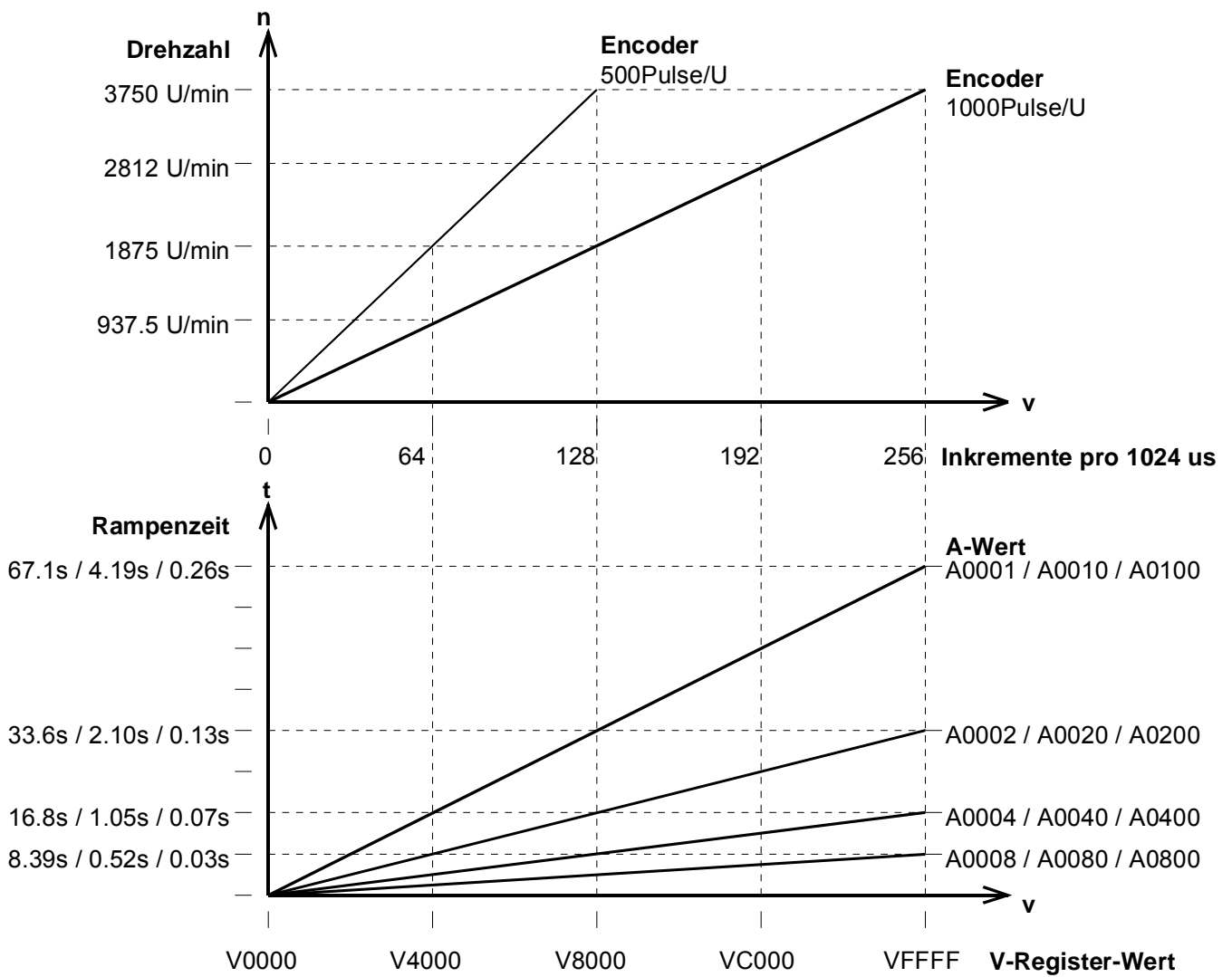
Speziell im Falle des Getriebefaktors (K - Value, Kommando "K") ist es wichtig zu wissen, dass nur Übersetzungsfaktoren programmiert werden können, deren Wert sich exakt als ein Vielfaches von 1/256 ausdrücken lässt.

0,5 möglich, aber 0,2 NICHT möglich !

Durch entsprechende Wahl der Encoderimpulszahlen kann man diese Einschränkung eventuell umgehen.

Die Demodiskette beinhaltet ein Programm, mit dem Sie gebrochene Hexadezimalzahlen beliebig umwandeln und berechnen können. (Vgl. Kap. 1.3)

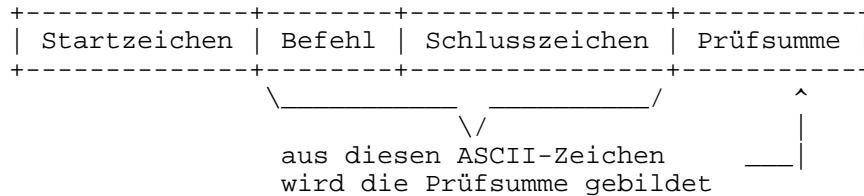
12.4 Grafik: V- und A - Wert, Rampenzeit, Drehzahl und Encodertyp



12.5 Berechnung der Prüfsumme

Die Berechnung der Prüfsumme geschieht nach folgender Regel:

Alle ASCII-Zeichen in einem Telegramm, mit Ausnahme des Startzeichens werden mit der logischen EXOR - Operation verknüpft. Am Schluss wird das Bit 7 (höchstwertiges Bit) auf 1 gesetzt.



Besteht ein Telegramm z.B.: aus folgenden Zeichen:

	ASCII-Zeichen	Hexwert	Bitmuster
Startzeichen	STX	02	
	"T"	54	01010100
	"-"	2D	00101101
	"1"	31	00110001
	"0"	30	00110000
	"0"	30	00110000
	"0"	30	00110000
Endzeichen	ETX	03	00000011

		EXOR - Verknüpfung der Bitmuster	01111011

		vorderstes Bit (Bit7) auf 1 => Prüfsumme	11111011
		Hexwert des Zeichen:	v
			FBhex

Dieses Zeichen wird unmittelbar anschliessend an das ETX - Zeichen gesendet und ist Bestandteil des Telegramms.

Im SPS-Betrieb werden nur die Bits 6..0 gesendet also 1111011, was Hexwert 7Bhex oder ASCII-Zeichen "{" bedeutet.

→ Die Demodiskette beinhaltet ein Programm, mit dem Sie die Prüfsumme zu einem beliebigen Befehl berechnen können. (Vgl. Kap. 1.3)

12.6 ASCII-Tabelle und Hexadezimale Ziffern

Dez	Hex	Char	Bez.	Dez	Hex	Char	Dez	Hex	Char	Dez	Hex	Char
0	00		NUL	32	20	(Space)	64	40	@	96	60	`
1	01		SOH	33	21	!	65	41	A	97	61	a
2	02		STX	34	22	"	66	42	B	98	62	b
3	03		ETX	35	23	#	67	43	C	99	63	c
4	04		EOT	36	24	\$	68	44	D	100	64	d
5	05		ENQ	37	25	%	69	45	E	101	65	e
6	06		ACK	38	26	&	70	46	F	102	66	f
7	07		BEL	39	27	'	71	47	G	103	67	g
8	08		BS	40	28	(72	48	H	104	68	h
9	09		HT	41	29)	73	49	I	105	69	i
10	0A		LF	42	2A	*	74	4A	J	106	6A	j
11	0B		VT	43	2B	+	75	4B	K	107	6B	k
12	0C		FF	44	2C	,	76	4C	L	108	6C	l
13	0D		CR	45	2D	-	77	4D	M	109	6D	m
14	0E		SO	46	2E	.	78	4E	N	110	6E	n
15	0F		SI	47	2F	/	79	4F	O	111	6F	o
16	10		DLE	48	30	0	80	50	P	112	70	p
17	11		DC1	49	31	1	81	51	Q	113	71	q
18	12		DC2	50	32	2	82	52	R	114	72	r
19	13		DC3	51	33	3	83	53	S	115	73	s
20	14		DC4	52	34	4	84	54	T	116	74	t
21	15		NAK	53	35	5	85	55	U	117	75	u
22	16		SYN	54	36	6	86	56	V	118	76	v
23	17		ETB	55	37	7	87	57	W	119	77	w
24	18		CAN	56	38	8	88	58	X	120	78	x
25	19		EM	57	39	9	89	59	Y	121	79	y
26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B		ESC	59	3B	;	91	5B	[123	7B	{
28	1C		FS	60	3C	<	92	5C	\	124	7C	
29	1D		GS	61	3D	=	93	5D]	125	7D	}
30	1E		RS	62	3E	>	94	5E	^	126	7E	~
31	1F		US	63	3F	?	95	5F	_	127	7F	i

ASCII-Zeichen	Hexwert des ASCII-Zeichen	Bitmuster des ASCII-Zeichen	Ziffer-Wert	Bitmuster des Zifferwertes
"0"	30hex	00110000	0	0000
"1"	31hex	00110001	1	0001
"2"	32hex	00110010	2	0010
"3"	33hex	00110011	3	0011
"4"	34hex	00110100	4	0100
"5"	35hex	00110101	5	0101
"6"	36hex	00110110	6	0110
"7"	37hex	00110111	7	0111
"8"	38hex	00111000	8	1000
"9"	39hex	00111001	9	1001
"A"	41hex	01000001	10	1010
"B"	42hex	01000010	11	1011
"C"	43hex	01000011	12	1100
"D"	44hex	01000100	13	1101
"E"	45hex	01000101	14	1110
"F"	46hex	01000110	15	1111

12.7 Geschichte bisheriger Versionen

1.03 [SW-ID 010319]

1.04 [SW-ID 01041b]

- Digitalausgänge durch AO5 (Trigger) und AO6 (Trigger - Direction) erweitert:
- Digitalausgang "In Pos." wird bei einer Home - Fahrt oder bei einer Referenzierung nun passiv.
- Falsche Anzeige des LED "Manual Mode" korrigiert.

1.04A [SW-ID 01041c]

- Probleme bei zu grossen Spannungen an den Uin - Eingängen behoben. Automatische Begrenzung, wenn: $a + v > FFFF$.
- Ein A - Wert von $>7FFF$ wird nicht mehr akzeptiert
- Profildgenerator wird anders geladen.

1.04B [SW-ID 010428]

- ZERO - Flag wurde fehlerhaft berechnet. Berechnung korrigiert und auch Funktion des Flags verändert: Es zeigt nun nur noch an, wenn der MC2 stillsteht, obwohl er drehen sollte (Blockierüberwachung).

1.04C [SW-ID 01042a]

- TARGET - Flag wird mit einem Go - Befehl im Positionsmodus sofort gelöscht. Bis anhin konnten bis zu 8 Millisekunden vergehen, bis das Target - Flag gelöscht wurde.

1.04D [SW-ID 01042c]

- Die Nullfahrt, bzw. deren erste Phase, nämlich die Fahrt zur Vorreferenz wird beschleunigt; sie wird neu mit den (zu dieser Zeit) geladenen Werten von A und V ausgeführt. Danach schaltet der Antrieb selbständig auf die Werte A0010 und V00E0 um, mit diesen Werten wird dann der eigentliche Nullimpuls gesucht.
- Die Befehle A, J, V und T können nun OHNE Parameter gesendet werden und geben als Antwort diejenigen Werte zurück, welche mit den entsprechenden Befehlen zuletzt geladen wurden. -> Um dieser Erweiterung Rechnung zu tragen, wurde vom Programm "MC2Com" die Version 1.1 erstellt.

1.04E [SW-ID 010433]

- Nullfahrt erweitert. Falls der Antrieb auf der Vorreferenz steht, wenn einer der Befehle "Z2" bis "Z7" ausgeführt wird, fährt dieser zuerst in der entsprechenden ("falschen") Richtung aus diesem Vorreferenzbereich hinaus, wendet dann und führt den aufgerufenen Nullfahrtsbefehl aus.
- Befehl "D" erweitert. Die Syntax lautet: "Dxxvii". Neben der Nullgeschwindigkeit vv (vgl. Version D) kann nun ein Bitmuster ii angegeben werden, mit welchem die Polarität jedes einzelnen Optokopplereingangs bestimmt werden kann. Das Flag "ESWPOL" (Befehl "F") wird "gestrichen", d.h. es hat keinen Einfluss mehr und verliert seinen Namen.
- Befehl "I" um einen Parameter erweitert: "IA" liest die Zustände der Optokopplereingänge direkt am Port ein. Die Stellung der individuellen Polaritätsflags (oberer Abschnitt) hat keinen Einfluss. 1 im Register heisst, der entsprechende Eingang wird bestromt, ob das nun "aktiv" oder "passiv" bedeutet, kann der Anwender bestimmen.
- Befehl "D": statt "set Defaults" neu "set Definitions".

(Die Release 1.05A .. D sind nicht vorhanden!)

1.05E

- Alle Funktionsmöglichkeiten der Version 1.04E wurden übernommen, mit Ausnahme des SPS - Mode (welcher zukünftig als Makro wieder ergänzt wird). Die Software ist nun fähig, mit einem Makro-Kompiler erzeugte Makros zu laden und auszuführen. (Makros werden nur durch das Werk programmiert).
- Der Dip - Switch 8 (ehemals "SPSMOD") wird nun mit "MACMOD" benannt. In seiner aktiven Stellung ruft er einen allfällig geladenen Makro auf, andernfalls erscheint der Mx4Q in seiner Standardkonfiguration mit Run - Eingang usw.
- Das NOVRAM erhält ein Datenfenster von 3D00hex bis 3EFFhex, diese Memory Zellen können nun mit dem Befehl "]"*aaaadd" auch dann beschrieben werden, wenn der Testmodus NICHT aktiviert ist. (Früher führte das zu einem CmdErr)

1.05F [SW-ID 010538]

- Während einer Bewegung kann mit Jump kein weiterer Jump "angehängt" werden. Dies ist erst möglich, wenn der Antrieb im Ziel (Target) ist.
- Wird ein Jump unterbrochen (mit V0, G oder HWStop), kann mit einem G (bzw. Vxxxx, G) das gleiche Ziel angefahren werden, bis anhin wurde das Ziel um die Sprungweite verschoben.
- Neuer Stoppmodus eingeführt, der mit dem freigewordenen Bit 0 der F-Bits codiert werden kann. Er veranlasst die Absetzung der Befehle "V0 ,G". Die Fahrdrehzahl wird zwischengespeichert. Mit einem G wird sie wieder übernommen und der Antrieb fährt dem ursprünglichen Ziel mit derselben Drehzahl entgegen!
- Z2 und Z3 werden vom Vorreferenzsignal nicht mehr beeinflusst.
- Neuer Befehl "Z8" kann verwendet werden, um das REFALT - Flag zu löschen. Nötig um über eine Mehrfach - Massstabsverschiebung im Bilde zu sein (Massstabsvergrößerung im Leitreechner).
- Z0 setzt die Startposition für weitere Jumps an die momentane Position.

1.05G [SW-ID 010439]

- Der Koppelfaktor (K - Faktor) wurde auf 24 Bit erweitert, davon neu 16 Bit hinter dem Dezimalpunkt. Die Syntax lautet neu: "Kkkkkkk" . Das Komma liegt zwischen dem 2. und dem 3. Parameter -> (kk.kkkk)

1.05H

- Zweiter Positions - Modus, Fly - Modus. Er erlaubt Jumps während der Bewegung zu verlängern (retriggerbare Jumps).
- Flagparameter erweitert: Bit FlyMod
- Out - Befehl erweitert. Mit dem Out - Befehl können nun einzelne Bits direkt gesetzt, ge-anded, ge-ored oder gelöscht werden:
- "Capture" - Funktion eingebaut. Sie erlaubt die Position aufgrund eines Signals (0-1-Flanke am Pin DIO7) in ein Register einzulesen und später über die Schnittstelle abzufragen und weiterzuverarbeiten.
- Die Funktion B wurde leicht angepasst, damit sie ins allgemeine Konzept passt, dass ein Befehl ohne Parameter eine Anfrage darstellt (Ausnahme G und H, die nie Parameter haben).
- Statuswort erweitert um den Zustand des Brake-point - Capture - Systemes anzuzeigen:

1.07 [SW-ID 010753]

- Verbesserung des Betriebes mit RS485: Die Pause beim Umschalten von Empfang auf Senden dauert jetzt um 700 Mikrosekunden, so dass auch ein PC-Sender Zeit hat, sich von der Schnittstelle zu verabschieden.
- Re - Implementation des SPS - Modus als Makro. Mit DIP - Switch 8 auf ON wird ab V1.07 ein Makro abgearbeitet, welches den SPS - Betrieb emuliert, der bis zur Version 1.04E als alternative Betriebsart möglich war.
- Konsequente Erweiterung der Kommando, so dass (mit Ausnahme vom "G" und "H") alle Kommando mit keinem Parameter als ANFRAGEN oder (in wenigen Fällen) als ungültige Kommando erkannt werden.

1.08A [SW-ID 010877]

- Automatisches Erkennen des Endstufentyps und Setzen der Parameter für Maximalstrom (Y - Register) und der Überstromzeitfläche (Register 4Dhex) beim Umladen (Befehl *_3A).

1.08B [SW-ID 01087a]

- Entkoppeln der Regelgerätefunktion HW - Stop vom EmStop - Flag.
- Einführen des Flags HWStop in den F - Flags.
- Abfrage der X - Parameter: dd und ss waren vertauscht.

1.08C und**1.08D** [SW-ID 010884]

- HW - Stop konnte nicht mehr zurückgesetzt werden. Fehler korrigiert.

1.08Dc [SW-ID 010853]

- Kommunikationsprobleme behoben indem die Parallelkommunikation in "mc2mac" deinstalliert wurde. ACHTUNG: Diese Version wurde aufgrund der Erkenntnisse bei Version 1.09L erstellt.

1.09 (1.09A) [SW-ID 010987]

- Der Emergency - Stop - Eingang und die Limit - Eingänge waren Flankengetriggert. Beim Lesen des E - Registers wurden sie zurückgesetzt, auch wenn sie weiter aktiv waren.
- Wenn im Handbetrieb der Motor vorwärts lief, stoppte er kurz beim Aktivieren des Rev - Endschalters.

- Ein Home oder eine Referenzierung wurde trotz aktiver Endschalter durchgeführt.
- Beim X - Befehl wurden die nicht angegeben führenden Nullen u.U. nicht gelöscht.
- Rücksetzen des Run bei Endstufenfehler "PowErr". (Bedingt auch HW - Änderung in der Endstufe).
- "PowErr" - Flag eingeführt (E - Register).
- Neue Zuordnungen der digitalen Ausgänge:
 - "Torque" statt "Rev-Lim" (Wenn der Controller Moment an die Endstufe kommandiert, ist dieser Pin High).
 - "LimErr" statt "Fwd-Lim"
 - "PosErr/" statt "SpeedError"

1.09B

- Bei einer Fehlermeldung der Endstufe verriegelte sich u.U. der Controller. (Bug seit 1.09)

1.09C

- Die Referenzierung mit einer Vorreferenz verbessert (Mindestabstand von 10ms auf 265 µs verkürzt).
- Das B-Register enthält nach einer Referenzierung die Distanz zum Nullimpuls.

1.09D [SW-ID 010924]

- Softwareanpassung an weitere Endstufen waren nötig.
- Umbenennung des PowErr (E-Flag) zu DrvErr. DrvErr wird nur noch aktiv, wenn der Endstufe auch ein Drehmoment kommandiert wird.
- Der Ausgang "Torque" wird auch dann passiv, wenn die Endstufe von sich aus das Drehmoment ausschaltet (Fehlerfall).
- Schaltet die Endstufe wieder ein (Fehler behoben z.B. Übertemperatur abgeklingen), wird das Drehmoment eingeschaltet und der Regler positioniert auf die aktuelle Position. Weiterfahrt mittels Go - Befehl möglich.
- PosErr wird durch das Lesen des E-Register nicht mehr gelöscht.
- Es wird ein CmdErr (Comand - Error) erzeugt, wenn ein Go Befohlen wird, die Endstufe aber deaktiviert ist. Ebenso bewirkt ein Home - Befehl ein CmdErr, wenn die Endstufe Fehler meldet.
- Die LED H2 (Inhibit) leuchtet nun nicht nur, wenn kein Drehmoment kommandiert wird, sondern auch, wenn das Drehmoment durch die Endstufe (Fehlerfall) ausgeschaltet ist. Sinngemäss verhält es sich mit dem "Torque" - Signal (Klemme 28c)

1.09E

- Eine neue Beschleunigung wird nun (bei ist - Drehzahl 0) auch übernommen, wenn sich das Vorzeichen der Drehzahl nicht ändert. Bisher wurde im M2 Modus eine allfällige Änderung der Beschleunigung nur übernommen, wenn sich gleichzeitig das Vorzeichen der Drehzahl änderte. (Work-around war: Von Pos Drehzahl auf n=0 statt ein "V0", ein "V-0" kommandieren).

1.09F

- Eine neue Beschleunigung wird neu im M2 - Modus bei beliebiger konstanter Drehzahl mit dem "G" - Kommando übernommen.

1.09G

- SW-Bug von 1.09f (im M2-Modus: Erhöhung des V ab gewissen V-Werten nicht möglich) korrigiert.

1.09H

- Sehr selten aufgetretener Fehler "Antwort ohne Parameterwert" korrigiert.

1.09i [SW-ID 010834]

- AltRef nur mit Z0 löschen, statt mit jedem Z-Befehl (Fehler seit 1.09)

1.09K

- ErrFlg im S-Register wurde gelöscht wenn kein Fehler mehr anstand, obwohl das E-Register (durch lesen) aber noch nicht zurückgesetzt wurde.

1.09L [SW-ID 01093a]

- Höchst selten aufgetretener Fehler "keine Antwort", trotz AKN (acknowledge) behoben.
- SPS - Mako: Parallelschnittstelle wird nicht mehr unterstützt (verursachte auch inaktiv den Fehler "keine Antwort").

1.09M [SW-ID 01095a]

Beim Referenzieren mit Z4..7 wird nach dem Auffinden der Vorreferenz (P wird dadurch auf 000000 gesetzt) und positionieren auf 000000 nicht auf Erreichen der Position 000000 getestet, sondern auf das Erreichen des Zielfensters, welches mit Xss definiert werden kann (Target=1), bevor der Z-Impuls gesucht wird.

1.10 Sonderversion, nicht weiter unterstützt**1.11A** [SW-ID 010B69]

- Erweiterte Erkennung und Ausgabe der angeschlossenen Endstufe im J-Register.
Neu wird die überarbeitete Endstufe (Revision E) des MD4Q 30/.. resp. 60/.. erkannt und der mögliche dynamische Strom von 150% zugelassen.
- Der Typ der Endstufe wird beim Power up resp. beim Umladen (Befehl *_3A) ins J-Register geschrieben.
ACHTUNG: Wird ein NovRAM verwendet, wird das J-Register beim Power-up NICHT mit dem abgespeicherten Wert initialisiert!

Die bisherigen Endstufen können problemlos mit dieser SW-Version eingesetzt werden.

Aber ein MD4Q der Bauform BKB resp. BKK erfordert ab Revision E (mit UP=10k und R234 & R235=49k) für ein korrektes Funktionieren eine SW ab Version 1.11!

13 Schlussbemerkungen

Der Inhalt dieser Dokumentation bezieht sich auf die aktuell gültige Softwareversion:

MC2 V1.11A (Software ID-Code: 010B69)

Autoren dieser Dokumentation:

25.09.1991	Thomas Walder	Erste Version
bis 02.10.2003	Daniel Maier	Überarbeitung bis SW-Version 1.11A
bis 20.09.2006	Daniel Maier	Überarbeitung Handbuch

Hardmeier Control hat den Inhalt dieses Handbuches auf Übereinstimmung mit Hard- und Software geprüft. Dennoch sind Abweichungen nicht auszuschliessen, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen.

Die Angaben in diesem Handbuch werden regelmässig überprüft und die notwendigen Korrekturen in den folgenden Ausgaben angebracht.

Sollten sie Unstimmigkeiten entdecken oder Verbesserungsvorschläge haben, sind wir Ihnen sehr dankbar für Ihre Mitteilung.

Sollten Sie Informationen zu älteren SW-Versionen benötigen, setzen Sie sich bitte mit uns in Verbindung.

Technische Änderungen vorbehalten.



Hardmeier Control

+41 (0)52 355 12 12



+41 (0)52 355 12 11

email: mailbox@hardmeier-control.ch

homepage: <http://www.hardmeier-control.ch>