

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Handbuch SLCAN

Handbuchversion V1.08

Gültig ab Hardware-Version SLCAN106 / Software-Version SLCAN V1.35

Allgemeines	3
Spannungsversorgung	3
LED-Betriebsanzeige	3
Motoranschlüsse	3
Anschluss bürstenbehafteter Motore.....	3
Anschluss bürstenloser Motore	3
Anforderungen an den Motor	3
Enkoderanschlüsse	4
Anschluss Enkoder mit massebezogenen Ausgangssignalen.....	4
Anschluss Enkoder mit differenziellen Ausgangssignalen	4
Endschalteranschlüsse	4
Anschluss der RS232	5
Digitale Ausgänge	5
Digitale/analoge Eingänge	5
Befehlssatz	6
Drehzahlregelmodus	6
sv n	6
Positioniermodus	6
ma n.....	7
mr n.....	7
Befehlstabelle	7
shex n	9
rv.....	9
sa n.....	9
ra.....	9
kp n / ki n / kd n	9
qp / qi / qd.....	9
sp n.....	9
rp.....	9
ca n.....	10
rcal.....	11
scv n	11
rcv	11
sca n	11
rca.....	11
pe.....	11
ssyscon n.....	11
rsyscon	12
rrsyscon	12
sipw n	12
ripw	12
sipt n	12
ript.....	12
scl n	12
rcl.....	12
ss	12
Auswertung der Flags	13

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

rin.....	13
sout n.....	13
rad n.....	14
id.....	14
pg.....	14
sl n.....	14
rl.....	14
sil n.....	14
ril.....	14
scv n.....	15
rcv.....	15
sca n.....	15
rca.....	15
saddr n.....	15
spwm n.....	16
sneglimit n.....	16
rneglimit.....	16
sposlimit n.....	16
rposlimit.....	16
sparama n.....	16
rparama.....	16
sparamb n.....	16
rparamb.....	16
sparamc n.....	16
rparamc.....	16
Befehle für SLLCD.....	17
slcd n.....	17
rlcd.....	17
Fehlermeldungen.....	18
Fehlermeldungen über RS232.....	18
Fehlermeldungen über CAN.....	18
RS232 und CAN.....	18
Abmaße und Anschlüsse.....	19
Jumper.....	19
Option LC-Display.....	19
Schematics.....	20
Anhang.....	21
Spannungsversorgung.....	21
Winkelenkoder.....	21
Welche Einstellungen werden dauerhaft gespeichert?.....	22
Anmerkungen zum Positionierfehler-Fenster (Flag inpos).....	22
Kommunikationsprotokoll (RS232).....	22
PID-Parameter.....	24
Verdrahtungshinweise.....	25
CAN-Modul Beschreibung V1.00.....	25
CAN 2.0A.....	25
CAN 2.0B.....	25
CAN Baudrate.....	25
CAN Identifier.....	25
Auslieferungszustand.....	25
CAN Datenformat.....	26
Empfang.....	26
Senden.....	26
CAN Protokoll.....	26
Befehlskennungen.....	27

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Allgemeines

Die SLCAN verfügt über zwei Kommunikationsschnittstellen: Eine RS232-Verbindung und eine CAN-Bus-Verbindung. Beide Schnittstellen können weder aus- noch eingeschaltet werden. Sie laufen beide parallel nebeneinander. Ein gleichzeitiger Anschluss über RS232 und CAN ist möglich, sollte aber nur zu Testzwecken erfolgen. So kann der Anwender in der Lernphase leicht und unmittelbar überprüfen ob CAN-Nachrichten tatsächlich angekommen sind und ob sie richtig interpretiert wurden.

Die SLCAN verfügt über einen Forth-Interpreter (grForth) mit dem eigene Programme geschrieben werden können. Es gibt auch eine Autostart-Funktion, die nach dem Einschalten grForth startet und die Funktion „autostart“ sofort ausführt (Stand-alone-Betrieb).

Spannungsversorgung

12..30V DC, Stromaufnahme abhängig vom Motor. Leerlaufstromaufnahme ca. 100mA.

Die Spannungsversorgung erfolgt nur über die Anschlüsse Power 24V und Power GND.

Das ist die einzige notwendige Spannungsversorgung. Aus ihr werden alle intern benötigten Spannungen abgeleitet und teilweise auch zur Verfügung gestellt.

Wird an den Anschlüssen VCC (+5V) / GND eine Spannung eingespeist, wird die Baugruppe sofort zerstört !

Siehe auch: Anhang/Spannungsversorgung

LED-Betriebsanzeige

Die auf dem Board befindliche LED zeigt unmittelbar nach Power-On den Betriebsbereitzustand an. In den Modi Positioniermodus (“pm”), Drehzahlregelmodus (“vm”) und PWM-Modus (“spwm”) signalisiert sie durch ihr Verlöschen (oder Flackern), dass der Motorstrom von der internen Elektronik begrenzt wird. Die Motorstrombegrenzung ist im Auslieferungszustand auf ca. 500mA eingestellt. Zur Einstellung und Abfrage der Motorstrombegrenzung dienen die Befehle “scl n” und “rcf”.

Motoranschlüsse

Vor Beginn der Verdrahtung lesen Sie bitte im Anhang den Abschnitt Verdrahtungshinweise !

Es können bürstenlose Motoren mit integrierten Hallsensoren oder bürstenbehaftete DC-Motoren angeschlossen werden.

Es sollten stets Motoren mit 24V-Wicklungen bevorzugt werden. Anschließbar sind aber auch Motoren mit Wicklungen ab 12V.

Anschluss bürstenbehafteter Motore

Die Motorwicklungen werden an Phase A und Phase C angeschlossen. Die Anschlüsse Phase B, Hall A, Hall B und Hall C bleiben unbeschaltet.

Anschluss bürstenloser Motore

Die drei Wicklungen dieser Motoren werden an Phase A, Phase B und Phase C angeschlossen. Die Hallsensoren entsprechend an Hall A, Hall B und Hall C. Die Spannungsversorgung der Hallsensoren wird an die Anschlüsse VCC (+5V) und GND angeschlossen.

Anforderungen an den Motor

Anschließbar sind bürstenlose und bürstenbehaftete Motoren mit inkrementalem Winkelenkoder. Die **minimale Anschlussinduktivität** des Motors beträgt **1mH**. Sollte der eine geringere Induktivität haben, sind in die Motorzuleitungen Drosseln mit entsprechender Strombelastbarkeit

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

einzufügen. Der Grund für diese Maßnahme ist, die Stromwelligkeit herabzusetzen um die Endstufen nicht zu zerstören.

Beherzigen Sie diese Vorschrift nicht, kann das zu abgebrannten Leiterbahnen und defekten Endstufen-Transistoren führen.

Enkoderanschlüsse

Anschließbar sind nur inkrementale Enkoder mit Quadratur-Ausgängen.

Es lassen sich nur Enkoder mit massebezogenen 5V-Ausgangssignalen (TTL oder Open Collector) oder Enkoder mit differenziellen Ausgangssignalen nach RS422 anschließen. Eine Nullspur ist für den Betrieb nicht zwingend notwendig, sie wird lediglich beim Kalibrieren benutzt, falls ein entsprechender Kalibrierbefehl verwendet wird (siehe „ca“). Verfügt der Enkoder über keine Nullspur kann lediglich auf einen der beiden Endschalter kalibriert werden.

Anschluss Enkoder mit massebezogenen Ausgangssignalen

Die Signale der Spuren A und B werden an die Anschlüsse CH A und CH B angeschlossen. Eine vorhandene Nullspur an den Anschluss INDEX. Die Anschlüsse CH A\, CH B\ und INDEX\ bleiben unbeschaltet. Die Spannungsversorgung kann den Anschlüssen VCC (+5V) und GND entnommen werden. Die Eingänge verfügen über Pull-Up-Widerstände. So lassen sich auch Enkoder mit Open-Collector-Ausgängen betreiben.

Anschluss Enkoder mit differenziellen Ausgangssignalen

Die Signale A, A\, B, B\ und ggf. N und N\ werden mit den Anschlüssen CH A, CH A\, CH B, CH B\ und ggf. INDEX und INDEX\ verbunden. Eine Spannungsversorgung steht an den Anschlüssen VCC (+5V) und GND zur Verfügung.

Um der Elektronik mitzuteilen, welcher Typ Enkoder angeschlossen ist, wird ein entsprechendes Bit gesetzt. Das Bit ist Bestandteil der Systemkonfiguration (siehe ssyscon).

Es sollte stets der korrekte Typ eingestellt werden, da es sonst zu Zählverlusten kommen kann.

Die Empfängerbausteine besitzen einen guten ESD-Schutz.

Die Signale A und B sind gegeneinander um 90° phasenverschoben. Da die Elektronik der SLCAN alle Flanken zählt, ergibt sich eine vierfach höhere Auflösung. Das bedeutet, dass ein Enkoder mit 16 Linien eine Auflösung von 64 Positionen (Counts) pro Umdrehung hat. Das wiederum bedeutet, dass die Elektronik auf $(360^\circ/64=) 5,62$ Winkelgrad genau positionieren kann. Diese Vervielfachung hat auch Einfluss auf die gewünschte Drehzahl und Beschleunigung, die angegebenen Formeln berücksichtigen diesen Umstand.

Endschalteranschlüsse

Bei rotatorischen Anwendungen gibt es i. A. keine Endschalter. Dann bleiben die Anschlüsse unbeschaltet jedoch muss der SLCAN mitgeteilt werden, dass keine Endschalter vorhanden sind.

Hierfür gibt es einen entsprechenden Befehl:

s/0 schaltet beide Endschalter ab.

s/1 schaltet Endschalter 1 ein, Endschalter 2 ab.

s/2 schaltet Endschalter 2 ein, Endschalter 1 ab.

s/3 schaltet beide Endschalter ein.

Die Endschaltereingänge führen über einen Vorwiderstand auf die LED's in Optokopplern. Da beide Anschlüsse der LED's herausgeführt sind, gibt es die Anschlüsse

END1A und END1K (A=Anode K=Kathode) und

END2A und END2K.

Der eingebaute Vorwiderstand für die LED's beträgt je 2400Ohm und ist damit für 24V ausgelegt.

Extrem wichtig ist der korrekte Anschluss der Endschalter im Bezug auf räumliche Anordnung und Drehsinn des Motors. Endschalter 1 wird der Schalter, der betätigt werden soll wenn der Motor in negative Richtung verfahren wird. Endschalter 2 wird der Schalter, der betätigt wird, wenn der Motor in positive Richtung verfahren wird.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Stimmt diese Zuordnung nicht, lässt sich der Motor nicht verfahren oder fährt auf den mechanischen Anschlag. Ferner werden die internen Kalibrierroutinen („ca“) nicht ordnungsgemäß funktionieren.

Anschluss der RS232

Der Anschluss RXD ist der Empfangskanal des Moduls. Der Anschluss TXD der Sendekanal. Wird die SLCAN mit dem Adapter 2 betrieben kann der Anschluss auch über die darauf befindliche 9pol-SubD-Buchse erfolgen. Dazu ist ein Kabel notwendig bei dem keine Vertauschungen stattfinden, d.h. die Anschlüsse 2 und 3 des Kabels dürfen nicht getauscht sein. Die Signale der RS232 entsprechen annähernd der Norm (+8V) und sind ESD geschützt.

Prinzipiell werden nur ASCII-Zeichen zur Befehlsübermittlung übertragen. Da keinerlei Hardware-Handshake benutzt wird, ist ein einfaches Software-Handshake von Nöten das aber recht einfach programmiert werden kann. Zunächst bleibt festzuhalten, dass die Kommunikation für die ersten Gehversuche mit unserem angepassten Terminal GrTermW erfolgen sollte. Es ist aber auch die Benutzung jedes anderen Terminalprogramms möglich, wenn auch mit gewissen Komforteinbußen. Wird nicht GrTermW eingesetzt muss das betreffende Programm konfiguriert werden:

19200Bd, 8 Datenbit, No Parity, 1 Stopbit, alle Hardware-Handshake-Modi aus, auch kein XON/XOFF-Protokoll.

Digitale Ausgänge

Die Ausgänge haben TTL-Pegel und sind ungeschützt. Der max. Ausgangsstrom (Sink und Source) beträgt 5mA.

Die Ausgänge können mit dem Befehl

sout n

gesetzt werden. *n* steht dabei für eine Zahl zwischen 0 und 255.

Beispiel:

Soll nur Ausgang LOUT0 auf High gesetzt werden alle anderen Ausgänge auf Low so wird das Kommando

sout 1

gegeben.

Soll nur Ausgang LOUT1 auf High gesetzt werden alle anderen Ausgänge auf Low so wird das Kommando

sout 2

gegeben.

Sollen die Ausgänge LOUT0 und SOUT1 auf High gesetzt werden wird das Kommando

sout 3

gegeben.

Anschluss	Dezimaler Wert
LOUT0	1
LOUT1	2
LOUT 2	4
LOUT 3	8
LOUT 4	16
LOUT 5	32
LOUT 6	64
LOUT 7	128

Digitale/analoge Eingänge

Die Eingänge haben generell einen Eingangsspannungsbereich von 0..5V und sind ungeschützt. Jeder der Eingänge LIN0 bis LIN7 ist sowohl mit einem TTL-Eingangsgatter als auch mit einem Analogmultiplexer verbunden. Man kann also jeden Eingang entweder als TTL-Pegel einlesen oder die anliegende Spannung mit einer Auflösung von 10Bit wandeln lassen.

Zum digitalen Einlesen dient der Befehl "rin". Er liefert einen Wert zwischen 0 und 255.

Beispiel:

Anschluss	Dezimaler Wert
LIN0	1
LIN1	2
LIN 2	4
LIN 3	8
LIN 4	16
LIN 5	32
LIN 6	64
LIN 7	128

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Wird der Wert 85 zurückgeliefert bedeutet das, dass LIN0=High ist, LIN1=Low, LIN2=High, LIN3=Low, LIN4=High, LIN5=Low, LIN6=High und LIN7=Low war.

Der gelieferte Wert ist also ein binär kodierter dezimaler

Wert:

$85 = 1 + 4 + 16 + 64$

Wie bereits erwähnt, kann die Spannung an jedem einzelnen Eingang digital gewandelt werden. Der Befehl dazu lautet "rad n" wobei das n durch eine Ziffer zwischen 0 und 7 zu ersetzen ist. "rad 5" wandelt die an LIN5 anliegende Spannung in einen digitalen Wert. Der gelieferte Wert liegt im Bereich zwischen 0 und 1023, wobei 0=0V Eingangsspannung, 511=2,5V und 1023=5,0V entsprechen.

$U = \text{Wert} * 0,00488V$

Intern sind generell die TTL-Eingangsschaltkreise auf die Pin's des Prozessors geschaltet. Mit dem Empfang des "rin n"-Befehls werden die Eingänge digital ausgelesen und der Wert gesendet.

Trifft ein "rad n"-Befehl ein, wird der TTL-Eingangsschaltkreis abgehängt und der analoge Eingangsmultiplexer auf den Pin geschaltet und anschließend der AD-Wandler gestartet. Nach dessen Arbeitsende wird der ermittelte Wert gesendet, der Multiplexer abgehängt und der TTL-Eingangsschaltkreis wieder aufgeschaltet.

Befehlssatz

Zunächst muss zwischen den beiden möglichen Betriebsmodi unterschieden werden:

- 1) Drehzahlregelung
- 2) Positioniermodus

Drehzahlregelmodus

Der Modus wird mit dem Befehl "vm" eingeschaltet. Der Motor wird nun mit der mit "sa n" eingestellten Beschleunigung auf die mit "sv n" vorgegebene Drehzahl beschleunigen und laufen, bis der Befehl "st" empfangen wird. Mit "st" wird der Modus abgeschaltet und der Motor stromlos.

sv n

Set Velocity

Setzen der Motordrehzahl. Beispiel:

sv 500

Berechnung der Drehzahl:

ve = der Wert der mit sv übergeben wird

UpM = die gewünschte Drehzahl in Umdrehungen pro Minute

Linien = Linien des Encoders

$ve = (UpM * \text{Linien}) / 234,37$

$UpM = (ve * 234,37) / \text{Linien}$

Beispiel:

Der benutzte Encoder ist mit 512 Linien angegeben. Um den Motor nun mit einer Drehzahl von 2500UpM drehen zu lassen ist der Wert 5461 zu übergeben:

$(2500 * 512) / 234,37 = 5461,449$

Mit "sv" kann ein positiver oder negativer Wert übergeben werden, entsprechend ändert sich die Drehrichtung. Zulässiger Wertebereich $\pm 2^{15}$.

Positioniermodus

Der Modus wird mit dem Befehl PM eingeschaltet. Der Motor wird nun lagegeregelt (bestromt) bis der Befehl "st" empfangen wird. Mit "st" wird der Motor stromlos geschaltet.

Der Motor kann nun relativ ("mr n") oder absolut ("ma n") zur augenblicklichen Position verfahren werden.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

ma n

Move Absolute

Der Motor wird auf die angegebene Position verfahren. Beispiel:

ma -10000

wird den Motor auf die Position –10000 verfahren.

Das Ende der Bewegung kann über die Flag's "move" und/oder "inpos" des Status ermittelt werden.

Voraussetzung: Der Positioniermodus ("pm") muss eingeschaltet und die Endschalter müssen korrekt konfiguriert sein.

Ab Software-Version 1.28:

1) Liegt der übergebene Wert unter dem Wert der mit *sneglimit* übergeben wurde erfolgt eine Fehlermeldung und die Bewegung wird nicht ausgeführt.

2) Liegt der übergebene Wert über dem Wert der mit *sposlimit* übergeben wurde erfolgt eine Fehlermeldung und die Bewegung wird nicht ausgeführt.

mr n

Move Relative

Der Motor wird um die angegebenen Positionen verfahren. Beispiel:

mr 2000

wird den Motor um 2000 Positionen verfahren.

Das Ende der Bewegung kann über die Flag's "move" und/oder "inpos" des Status ermittelt werden.

Voraussetzung: Der Positioniermodus (*pm*) muss eingeschaltet und die Endschalter müssen korrekt konfiguriert sein.

Die Beschleunigung des Motors ist von dem Wert *n* abhängig, der mit dem Befehl

sa n

Übermittelt wurde. Die Verfahrgeschwindigkeit (Motordrehzahl) ist von dem Wert *n* abhängig, der mit dem Befehl

sv n

übermittelt wurde. Während einer Bewegung kann weder die Beschleunigung noch die Verfahrgeschwindigkeit geändert werden. Eine Bewegung kann jederzeit mit *st* abgebrochen werden.

Befehlstabelle

Zu einigen Befehlen gehört ein entsprechender Parameter. Er ist zusammen mit dem Befehl einzugeben und wird in der folgenden Liste durch *n* repräsentiert.

Es spielt keine Rolle ob Befehle In Groß- oder Kleinschreibung übermittelt werden.

Leerzeichen zwischen Befehl und Parameter können, brauchen jedoch nicht eingefügt zu werden.

Befehl	Erklärung
pm	Position Mode Positioniermodus einschalten
vm	Velocity Mode. Drehzahlregelmodus einschalten
st	STop Eingestellten Modus beenden, Motor stromlos schalten
ma <i>n</i>	Move Absolute Motor <u>AUF</u> Position <i>n</i> verfahren
mr <i>n</i>	Move Relative Motor <u>UM</u> <i>n</i> Positionen verfahren
rp	Read Position Auslesen der Motorposition
sp <i>n</i>	Set Position Setzen der Motorposition
sv <i>n</i>	Set Velocity Setzen der (max.) Drehzahl
rv	Read Velocity Eingestellte Drehzahl auslesen
sa <i>n</i>	Set Acceleration Beschleunigung einstellen
ra	Read Acceleration Eingestellte Beschleunigung auslesen
kp <i>n</i>	Koeffizient P Setzen des Reglerparameters P
ki <i>n</i>	Koeffizient I Setzen des Reglerparameters I
kd <i>n</i>	Koeffizient D Setzen des Reglerparameters D

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

qp	Query P	Auslesen des Reglerparameters P
qi	Query I	Auslesen des Reglerparameters I
qd	Query D	Auslesen des Reglerparameters D
ca <i>n</i>	CALibrate	Achse kalibrieren
scv <i>n</i>	Set Cal. Velocity	Setzen der Kalibrier-Drehzahl
sac <i>n</i>	Set Cal. Acceleration	Setzen der Kalibrier-Beschleunigung
rcv	Read Cal. Velocity	Auslesen der Kalibrier-Drehzahl
rca	Read Cal. Acceleration	Auslesen der Kalibrier-Beschleunigung
re	Position Error	Schleppfehler auslesen
ssyscon <i>n</i>	Set Configuration	Hard-/Stftware konfigurieren
rsyscon	Read Configuration	Auslesen der Konfiguration
rrsyscon	Read Configuration	Nur für Terminalbetrieb. Siehe weiter unten.
sipw <i>n</i>	Set InPos Window	inpos-Fenster setzen
ripw	Read InPos Window	Gesetzten Wert auslesen
sipt <i>n</i>	Set InPos Tlmer	Zeitgeber für inpos setzen
ript	Read InPos Timer	Gesetzten Wert auslesen
scl <i>n</i>	Set Current Limit	Motorstrombegrenzung setzen
rcl	Read Current Limit	Gesetzten Wert auslesen
sl <i>n</i>	Set Limits on/off	Ein-/Ausschalten der Endschalter-Funktion
rl	Read Limits on/off	Abfrage der Endschalter-Funktion
sil <i>n</i>	Set Invert Limits	Endschalter logisch invertieren
ril	Read Invert Limits	Auslesen der Einstellung von sil
ss	StatuS	Status auslesen
rss	Read StatuS	Status in Klartext auflisten(mehrzeilige Ausgabe!!)
rin	Read Inputs	Eingänge digital abfragen
rad <i>n</i>	Read AD-Converter	Spannung an LIN <i>n</i> wandeln
sout <i>n</i>	Set OUTputs	Ausgänge setzen
id	IDentification	Versionsmeldung abfragen
pg	ProGram	Einstellungen im EEPROM speichern
saddr <i>n</i>	Seriallink ADDRess	Setzen der Baugruppen-Adresse (0..15)
se <i>n</i>	Select	Baugruppe mit Adresse <i>n</i> selektieren
rcal	Read Calibration bit	Auslesen des "Kalibration erfolgt"-Bit
shex <i>n</i>	Set HEXadecimal format Low=Dezimaler / High Hexadezimaler Ausgabemodus	
spwm <i>n</i>	Set Pwm	Erzeugung eines Motorstroms zu Testzwecken

Die nachfolgenden Befehle sind mit Software-Version 1.28 eingeführt

sneglimit <i>n</i>	Set NEGativ LIMIT	Softlimit Negativ setzen
rneglimit <i>n</i>	Read NEGativ LIMIT	Softlimit Negativ auslesen
sposlimit <i>n</i>	Set POSitiv LIMIT	Softlimit Positiv setzen
rposlimit <i>n</i>	Read POSitiv LIMIT	Softlimit Positiv auslesen
sparama <i>n</i>	Set PARAMeter A	Kunden-Parameter A setzen
rparama <i>n</i>	Read PARAMeter A	Kunden-Parameter A auslesen
sparamb <i>n</i>	Set PARAMeter B	Kunden-Parameter B setzen
rparamb <i>n</i>	Read PARAMeter B	Kunden-Parameter B auslesen
sparamc <i>n</i>	Set PARAMeter C	Kunden-Parameter C setzen
rparamc <i>n</i>	Read PARAMeter C	Kunden-Parameter C auslesen

Die nachfolgenden Befehle beziehen sich nur auf eine angeschlossene SLLCD

slcd <i>n</i>	Set LCD type	Display-Typ setzen
rlcd	Read LCD type	Einstellung des LCD-Typs auslesen

Die nachfolgenden Befehle dienen zur Konfiguration des CAN-Anschlusses

scbr <i>n</i>	Set Can BaudRate	Datenübertragungsrate des CAN-Bus setzen
rcbr	Read Can BaudRate	Kontrolle der eingestellten CAN-Baudrate
sii <i>n</i>	Set Input Identifier	Setzen der Adresse für eingehende Meldungen
rii	Read Input Identifier	Kontrolle der Adresse für eingehende Meldungen
soi <i>n</i>	Set Output Identifier	Setzen der Adresse für ausgehende Meldungen
roi	Read Output Identifier	Kontrolle der Adresse für ausgehende Meldungen

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Hinweis: CAN 2.0A / CAN 2.0B wird über "ssyscon" eingestellt.

shex n

Set HEXadecimal output format

Wird mit

shex1

das hexadezimale Zahlenformat eingestellt, werden alle Werte die zurückgegeben werden als 8-stellige (32Bit) hexadezimale Zahl mit vorangestellter Kennzeichnung gesendet.

Beispiel:

Nach dem Senden des Befehls "rp" wird die Motorposition zurückgeliefert:

0x000003e8

"0x" ist die Hexadezimal-Kennung. "000003e8" ist die Position, dezimal 1000.

Übrigens können nun auch Werte in hexadezimaler Schreibweise von Ihrem Rechner an die SLCAN übermittelt werden:

Werte ohne vorangestelltes "0x" werden als dezimale Zahl interpretiert, Werte mit "0x" als hexadezimale.

Mit

shex 0

wird der hexadezimale Modus abgeschaltet.

rv

Read Velocity

Auslesen der eingestellten (maximalen) Verfahrensgeschwindigkeit bzw. Drehzahl.

sa n

Set Acceleration

Setzen der Beschleunigung. Beispiel:

sa 50

Berechnung der Beschleunigung:

ac = der Wert der mit sa übergeben wird

UpM/M = die gewünschte Beschleunigung in Umdrehungen pro Minute in der Minute

Linien = Linien des Encoders

$ac = (UpM/M * Linien) / 225000$

Beispiel:

Der benutzte Encoder ist mit 512 Linien angegeben. Um den Motor nun mit 5000UpM/M zu beschleunigen ist der Wert 11 zu übergeben:

$(5000 * 512) / 225000 = 11.37$

ra

Read Acceleration

Auslesen der eingestellten Beschleunigung.

kp n / ki n / kd n

Einstellen der Reglerparameter P, I und D. Beispiel:

kp 40

ki 40

kd 80

qp / qi / qd

Auslesen der eingestellten Reglerparameter P, I und D

sp n

Set Position

Der Befehl "sp..." ist nur wirksam, wenn weder "pm" noch "vm" eingeschaltet sind – anders ausgedrückt:

"sp..." funktioniert nur nach "st".

Mit dem Befehl kann der interne Motorpositionsähler auf jeden Wert zwischen -33554431 und +33554431

($\pm 2^{25}$) gesetzt werden. Beispiel:

sp 0

sp 5000

rp

Read Position

Die augenblickliche Motorposition wird ausgelesen. Der Befehl funktioniert immer, egal, in welchem Modus ("pm" oder "vm" oder auch "st") sich der Regler befindet. Zahlenbereich $\pm 2^{25}$.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

ca n

CAlibrate

Nach dem Einschalten hat der interne Positionszähler den Wert 0 (Null). Das hat jedoch nichts mit der mechanischen, tatsächlichen Position des Motors zu tun. Daher ist es auch hier, wie bei allen inkrementalen Systemen, notwendig, die tatsächliche mechanische Position zu ermitteln. Das geschieht, indem der Motor solange in eine bestimmte Richtung verfahren wird bis ein Schalter durch den Motor betätigt wird. Dieser Schalter ist natürlich in geeigneter Weise mit dem Regler verbunden und liefert daher die erforderliche Rückmeldung.

An die SLCAN lassen sich maximal zwei Endschalter und ggf. der Index-Puls des Winkelenkoders anschließen.

Der mit "ca" übergebene Parameter *n* bestimmt nun, welcher der drei Eingänge als Referenzschalter dienen soll:

Ca 0	Endschalter 1 dient als Referenzschalter (negative Drehrichtung)
Ca 1	Endschalter 2 dient als Referenzschalter (positive Drehrichtung)
Ca 2	Endschalter 1 und der Index-Puls dienen als Referenzschalter (negative Drehrichtung)
Ca 3	Endschalter 2 und der Index-Puls dienen als Referenzschalter (positive Drehrichtung)
Ca 4	Nur der Index-Puls dient als Referenzschalter (negative Drehrichtung)
Ca 5	Nur der Index-Puls dient als Referenzschalter (positive Drehrichtung)

Negative Drehrichtung: Der Motor dreht in Richtung negative Positionen

Positive Drehrichtung: Der Motor dreht in Richtung positive Positionen

Nach erfolgter Kalibrierung wird intern ein Flag gesetzt, das diesen Umstand anzeigt. Siehe *rca/*

ca 0

Der Motor wird mit der mit *scv* eingestellten Drehzahl und der durch *sca* vorgegebenen Beschleunigung in Richtung Endschalter 1 verfahren. Wird der Endschalter betätigt, wird der Motor sofort gestoppt. Danach wird der Motor mit 1/16 der mit *scv* eingestellten Drehzahl und 1/16 der mit *sca* eingestellten Beschleunigung in Richtung Endschalter 2 verfahren bis der Endschalter 1 wieder gelöst ist. Der Motor stoppt dann sofort und die Kalibrierfahrt ist beendet. Die Motorposition kann nun mittels "rp" ausgelesen oder mit "sp.." auf den gewünschten Wert gesetzt werden.

Die Vorgaben von *scv* und *sca* sind nur während des Kalibrierens wirksam. Danach sind automatisch wieder die Werte von *sv* und *sa* wirksam.

ca 1

Der Motor wird mit der mit *scv* eingestellten Drehzahl und der durch *sca* vorgegebenen Beschleunigung in Richtung Endschalter 2 verfahren. Wird der Endschalter betätigt, wird der Motor sofort gestoppt. Danach wird der Motor mit 1/16 der mit *scv* eingestellten Drehzahl und 1/16 der mit *sca* eingestellten Beschleunigung in Richtung Endschalter 1 verfahren bis der Endschalter 2 wieder gelöst ist.

Der Motor stoppt dann sofort und die Kalibrierfahrt ist beendet. Die Motorposition kann nun mittels "rp" ausgelesen oder mit "sp.." auf den gewünschten Wert gesetzt werden.

Die Vorgaben von *scv* und *sca* sind nur während des Kalibrierens wirksam. Danach sind automatisch wieder die Werte von *sv* und *sa* wirksam.

ca 2

Der Motor wird mit der mit *scv* eingestellten Drehzahl und der durch *sca* vorgegebenen Beschleunigung in Richtung Endschalter 1 verfahren. Wird der Endschalter betätigt, wird der Motor sofort gestoppt. Danach wird der Motor mit 1/16 der mit *scv* eingestellten Drehzahl und 1/16 der mit *sca* eingestellten Beschleunigung in Richtung Endschalter 2 verfahren bis der Endschalter 1 wieder gelöst ist.

Der Motor dreht nun solange weiter bis der Index-Impuls des Winkelenkoders erkannt wird.

Der Motor stoppt dann sofort und die Kalibrierfahrt ist beendet. Die Motorposition kann nun mittels "rp" ausgelesen oder mit "sp.." auf den gewünschten Wert gesetzt werden.

Die Vorgaben von *scv* und *sca* sind nur während des Kalibrierens wirksam. Danach sind automatisch wieder die Werte von *sv* und *sa* wirksam.

ca 3

Der Motor wird mit der mit *scv* eingestellten Drehzahl und der durch *sca* vorgegebenen Beschleunigung in Richtung Endschalter 2 verfahren. Wird der Endschalter betätigt, wird der Motor sofort gestoppt. Danach wird der Motor mit 1/16 der mit *scv* eingestellten Drehzahl und 1/16 der mit *sca* eingestellten Beschleunigung in Richtung Endschalter 1 verfahren bis der Endschalter 2 wieder gelöst ist.

Der Motor dreht nun solange weiter bis der Index-Impuls des Winkelenkoders erkannt wird.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Der Motor stoppt dann sofort und die Kalibrierfahrt ist beendet. Die Motorposition kann nun mittels "rp" ausgelesen oder mit "sp.." auf den gewünschten Wert gesetzt werden.

Die Vorgaben von *scv* und *sca* sind nur während des Kalibrierens wirksam. Danach sind automatisch wieder die Werte von *sv* und *sa* wirksam.

ca 4

Der Motor wird mit der Drehzahl von *scv* und Beschleunigung von *sca* in Richtung Endschalter 1 bzw. in negative Richtung verfahren bis der Index-Impuls des Winkelenkoder erkannt wird.

Der Motor stoppt dann sofort und die Kalibrierfahrt ist beendet. Die Motorposition kann nun mittels "rp" ausgelesen oder mit "sp.." auf den gewünschten Wert gesetzt werden.

Die Vorgaben von *scv* und *sca* sind nur während des Kalibrierens wirksam. Danach sind automatisch wieder die Werte von *sv* und *sa* wirksam.

ca 5

Der Motor wird mit der Drehzahl von *scv* und Beschleunigung von *sca* in Richtung Endschalter 2 bzw. in positive Richtung verfahren bis der Index-Impuls des Winkelenkoder erkannt wird.

Der Motor stoppt dann sofort und die Kalibrierfahrt ist beendet. Die Motorposition kann nun mittels "rp" ausgelesen oder mit "sp.." auf den gewünschten Wert gesetzt werden.

Die Vorgaben von *scv* und *sca* sind nur während des Kalibrierens wirksam. Danach sind automatisch wieder die Werte von *sv* und *sa* wirksam.

rcal

Read CALibration bit

Dieses Flag wird nach Power-On stets auf Low gesetzt. Es wird nur von einer erfolgreich ausgeführten Kalibrierung (siehe *ca n*) auf High gesetzt.

Damit kann der externe Rechner feststellen, ob die angeschlossene Achse bereits kalibriert wurde oder nicht.

scv n

Set Calibration Velocity

Setzen der (maximalen) Verfahrensgeschwindigkeit/Motordrehzahl während des Kalibrierens. Beispiel:

scv 500

Zur Berechnungen siehe *sv*.

rcv

Read Calibration Velocity

Auslesen der eingestellten Kalibrier-Drehzahl.

sca n

Set Calibration Acceleration

Setzen der Beschleunigung während des Kalibrierens. Beispiel:

sca 50

rca

Read Calibration Acceleration

Auslesen der eingestellten Kalibrier-Beschleunigung.

pe

Position Error

Dieser Befehl liefert die Differenz zwischen der vom Bewegungsprofil-Generator (Rampen-Generator) erzeugten Motorsollposition und der tatsächlichen Motor-Istposition. Diese Differenz wird oft auch als Schleppfehler bezeichnet.

Der Schleppfehler gibt Auskunft über das gesamte mechanische und elektrische System. Daran kann erkannt werden, ob eine Störung des Gesamtsystems vorliegt.

Wenn beispielsweise die Mechanik klemmt wird sich das sehr schnell als großer Schleppfehler bemerkbar machen. Es ist eine gute Idee, den Schleppfehler während einer Bewegung zyklisch abzufragen. Wie groß der Schleppfehler normalerweise ist, hängt ausschließlich vom Motor, den eingestellten Reglerparametern und der angeschlossenen Mechanik ab und kann hier nicht quantifiziert werden. Ermitteln Sie doch einfach den Schleppfehler in Ihrem einwandfrei arbeitenden System und legen Sie damit einen Grenzwert fest. Wird der Grenzwert überschritten können Sie dann geeignete Maßnahmen einleiten.

ssyscon n

Set SYStem CONfiguration

Mit diesem Befehl (und dem zugehörigen Parameter) wird die Baugruppe konfiguriert.

Bit-Nr.	Dezimaler Wert	Flag-Name	Bedeutung
0	1	LCDOn	High= LC-Display eingeschaltet (SLLCD)

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

1	2	RotOn	High= Rot. Encoder eingeschaltet (SLLCD)
2	4	BLOn	Motor: High= bürstenlos Low= bürstenbehaftet
3	8	EncType	Enkoder: High= differentiell (RS422) Low= Single ended
4	16	L1On	High= Endschalter 1 wird berücksichtigt
5	32	L2On	High= Endschalter 2 wird berücksichtigt
6	64	L1Inv	High= Endschalter 1 wird invertiert
7	128	L2Inv	High= Endschalter 2 wird invertiert
8	256	HexOn	High= Alle Zahlenausgaben in hexadezimaler Schreibweise
9	512	CANB	High=CAN 2.0B (29Bit) Low=CAN 2.0A (11Bit)
10	1024	autostart	High=Nach Power On wird sofort grForth gestartet

Zu Bit 10 autostart:

Diese Bit sorgt für den sofortigen Start von grForth und wird (sofern im Speicher) das Wort "autostart" sofort ausführen. Siehe dazu das Dokument "grForthxxx.rtf".

rsyscon

Read SYStem CONfiguration

Die eingestellte Konfiguration wird ausgelesen und angezeigt. Die Bedeutung der verschiedenen Flags, siehe "ssyscon"

rrsyscon

Gibt die mittels "ssyscon" eingestellte Konfiguration in Klartext mehrzeilig aus. Damit wird das übliche Protokoll: Ein Befehl -> Eine Antwort(zeile) durchbrochen ! Dieser Befehl sollte nur benutzt werden, wenn die Baugruppe an ein Terminal(programm) z.B. GrTermW angeschlossen ist. Dieser Befehl wird bei CAN-Bus-Betrieb nicht ausgeführt und erzeugt eine CAN-Fehlermeldung.

sipw n

Set InPos Window

Fenstergröße für das Flag "inpos" setzen. Beispiel

sipw 5

Nähere Erläuterungen finden Sie beim Befehl "ss" und dessen Flag "inpos" sowie im Anhang.

ripw

Read InPos Window

Gesetzte Fenstergröße auslesen. Es wird der Wert geliefert der mittels "sw n" gesetzt wurde.

sipt n

Set InPos Timer

Zeit für den Inpos-Timer in Millisekunden vorgeben. Beispiel

sipt 50

Setzt die inpos-Verweildauer auf 50ms. Nähere Erläuterungen finden Sie beim Befehl "ss" und dessen Flag "inpos" sowie im Anhang.

ript

Read InPos Timer

Gesetzten Timer-Wert auslesen. Es wird der Wert geliefert der mittels "sipt n" gesetzt wurde.

scl n

Set Current Limit

Mit diesem Befehl wird der maximal zulässige Motorstrom eingestellt. Der mitzuübergabende Parameter *n* darf zwischen 0 und 2000 liegen, das entspricht etwa 0 bis 2000mA. Anzumerken ist, dass der Einsatz der Strombegrenzung nicht so genau ist wie es scheint, zu viele Faktoren (Motorinduktivität, -innenwiderstand usw.) beeinflussen die Strommessung und die daraus abgeleitete Begrenzung. Eine Messung des Motorstroms und Nachjustage der Strombegrenzung ist dann angebracht, wenn ein bestimmter zulässiger Motorstrom auf keinen Fall überschritten werden darf.

Die auf dem Board befindliche grüne LED dient (unmittelbar nach dem Einschalten) zur Betriebsbereitanzeige. Sobald einer der Modi "pm", "vm" oder "spwm" eingeschaltet werden, zeigt ihr verlöschen an, dass der Motorstrom von der internen Elektronik begrenzt wird.

rcl

Read Current Limit

Der mit "scl n" gesetzte Wert wird ausgelesen.

ss

Status

Der von "ss" gelieferte Wert repräsentiert, binär kodiert, den Zustand der System-Flag's.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Bit-Nr.	Dezimaler Wert	Flag-Name	Bedeutung
Bit 0	1	limit1	Low=Endschalter 1 unbetätigt / High=Endschalter 1 betätigt
Bit 1	2	limit2	Low=Endschalter 2 unbetätigt / High=Endschalter 2 betätigt
Bit 2	4	pmode	High=Positioniermodus eingeschaltet
Bit 3	8	vmode	High=Drehzahlregelmodus eingeschaltet
Bit 4	16	move	Low=Motor steht / High=Motor wird verfahren
Bit 5	32	inpos	Low=Motor nicht in Sollposition / High=Motor in Sollposition
Bit 6	64	cal	Low="ca" nicht oder nicht erfolgreich ausgeführt

limit1 / limit 2

Durch Auswertung der beiden Flag's kann festgestellt werden, ob einer der Schalter durch die Mechanik betätigt ist oder nicht.

pmode

Das Flag ist High, wenn der Positioniermodus ("pm") eingeschaltet wurde. Es wird nach "st" Low.

vmode

Das Flag ist High, wenn der Drehzahlregelmodus ("vm") eingeschaltet wurde. Es wird nach "st" Low.

move

Das Flag wird High nachdem im Positioniermodus ("pm") ein Befehl zum Verfahren des Motors ("ma" oder "mr") empfangen wurde. Mit dem Empfang von "ma" oder "mr" startet der Rampengenerator mit seinen Berechnungen und das Flag "move" wird gesetzt. Das Flag bleibt High solange der Rampengenerator arbeitet, d.h. bis die vorgegebene Sollposition erreicht ist. Die Sollposition entspricht aber nicht unbedingt der Motor-Istposition (siehe "pe" – Schleppfehler) zeigt aber, ob die nächste Bewegung gestartet werden kann.

inpos

Das Flag wird mit dem Starten einer Bewegung auf alle Fälle auf Low gesetzt. Nachdem der Rampengenerator fertig ist (Move-Flag=Low) wird die augenblickliche Motorposition mit der gewünschten Sollposition verglichen. Ist die Differenz zwischen Soll- und Istposition kleiner als der eingestellte Wert von "sipw", wird ein Zeitgeber gestartet nach dessen Ablauf das Flag "inpos" auf High gesetzt wird. Die Länge des Timers wird von "sipt" gesteuert. Gerät der Motor während der ablaufenden Zeit aus dem Positionsfenster, wird der Timer zurückgesetzt und beginnt automatisch von vorne, "inpos" bleibt dadurch Low bis die gewünschten Bedingungen erfüllt sind.

Auswertung der Flags

In jeder Programmiersprache können die einzelnen Flag's recht leicht ausgewertet werden. Im folgenden Beispiel wird "C" verwendet:

Angenommen, der mit "ss" abgefragte Wert steht numerischer Wert in der Integer-Variablen status, dann kann mit

```
if( (status&4)==0 printf("Positioniermodus nicht eingestellt");  
else printf("Positioniermodus eingeschaltet");
```

die entsprechende Meldung erzeugt werden.

Mit

```
if ( (status&3)!=0) printf("Ein Endschalter ist betätigt!");
```

wird die Meldung ausgegeben wenn einer der beiden (oder beide) Endschalter betätigt ist.

rin

Read digital INputs

Alle acht Eingänge werden digital ausgelesen. Auch dieser gelieferte Wert ist binär kodiert.

Bit-Nr.	Dezimaler Wert	Anschluss
Bit 0	1	LIN0
Bit 1	2	LIN1
Bit 2	4	LIN2
Bit 3	8	LIN3
Bit 4	16	LIN4
Bit 5	32	LIN5
Bit 6	64	LIN6
Bit 7	128	LIN7

sout n

Es stehen acht Ausgänge zur Verfügung. Die Ausgänge sind TTL-Push-Pull-Ausgänge.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Bit-Nr.	Dezimaler Wert	Ausgang
Bit 0	1	LOUT0
Bit 1	2	LOUT1
Bit 2	4	LOUT2
Bit 3	8	LOUT3
Bit 4	16	LOUT4
Bit 5	32	LOUT5
Bit 6	64	LOUT6
Bit 7	128	LOUT7

Um nur LOUT3 auf High zu schalten: *sout 8*

Um nur die beiden Ausgänge LOUT0 und LOUT3 auf High zu schalten: *sout 9*

Um Alle Ausgänge auf Low zu schalten: *sout 0*

rad n

Der mit *n* angegebene Eingang wird auf den AD-Wandler geschaltet und eine Wandlung gestartet.

rad 0 liefert den Wert von LIN0,

rad 1 den Wert von LIN1 usw.

Der gelieferte Wert liegt zwischen 0 (0V) und 1023 (+5V).

id

Versionsmeldung der Baugruppe mit Angabe der Geräteseriennummer.

pg

Die augenblicklich eingestellten Parameter werden dauerhaft im EEPROM gespeichert. Nach dem Abschalten und dem späteren Einschalten werden die gespeicherten Werte gelesen und automatisch wieder eingestellt.

Welche Werte gespeichert werden entnehmen Sie bitte dem Anhang.

sl n

Set Limits on/off

Bit-Nr.	Dezimaler Wert	Wirkung auf	Bedeutung
Bit 0	1	Endschalter 1	Low=Endschalter 1 abgeschaltet / High=eingeschaltet
Bit 1	2	Endschalter 2	Low=Endschalter 2 abgeschaltet / High=eingeschaltet

Hier geht es darum, die Software anzuweisen die/den Endschalter zu berücksichtigen oder nicht zu berücksichtigen. Wenn Sie eine Mechanik benutzen an die nur ein oder kein Endschalter angeschlossen ist, ist diese Möglichkeit des Abschaltens recht sinnvoll.

Wichtig ist diese Funktion auch, wenn zum Referenzieren ein Schalter in der Mitte der Strecke benutzt wird.

Dazu aktiviert man vor der Kalibrierung den betreffenden Endschalter. Danach wird er deaktiviert und erlaubt somit ein Verfahren auf "beliebige" Positionen ohne dass der Endschalter die Bewegung abbricht.

rl

Read Limits on/off

Der Wert zeigt an, wie die Endschalter mit "sl" konfiguriert wurden.

Der tatsächliche Status (die "Stellung") der Endschalter spiegelt sich in "ss" Bit 0 und Bit 1 wieder.

sil n

Set Invert Limits

Bit-Nr.	Dezimaler Wert	Wirkung auf	Bedeutung
Bit 0	1	Endschalter 1	Low=Endschalter 1 nicht invertiert / High=invertiert
Bit 1	2	Endschalter 2	Low=Endschalter 2 nicht invertiert / High=invertiert

Im Normalfall, also bei "sil 0", wird erwartet, dass die Endschalter als Öffner arbeiten wenn sie betätigt werden. Dann liefern sie im Statuswort ("ss") jeweils 0 solange sie nicht betätigt sind. Aus vielen Gründen kann es notwendig sein, andere Endschalter zu benutzen. Z.B. Endschalter mit einem Schließer, d.h. beim Betätigen wird ein Kontakt geschlossen. Er liefert, solange er nicht betätigt ist, ein High (eine "1"). Im Normalfall (nach "sil 0") würde der Motor sich nicht rühren, da er beide Endschalter als betätigt erkennt.

Nach "sil 3" werden beide Endschalter logisch invertiert und sie liefern jeweils ein Low solange die Schließer nicht betätigt werden. Damit ist das Verfahren des Motors wieder möglich.

ril

Read Invert Limit

Nun ja, es wird geliefert, was mit "sil" eingestellt wurde.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

scv n

Set Calibration Velocity

Während der Kalibrierung ("ca n") wird diese besondere Kalibriergeschwindigkeit benutzt. Sie sollte im Interesse der Kalibrierengenauigkeit deutlich unter der sonst benutzten Verfahrgeschwindigkeit liegen – auch wenn das natürlich lange dauert. Aber dagegen ist auch ein Kraut gewachsen: Angenommen Sie möchten auf Endschalter 1 kalibrieren, dann können Sie zunächst mit einer mittleren Geschwindigkeit gegen den Endschalter fahren und dann erst den Befehl "ca.." benutzen. Das ergibt eine gewisse Zeitersparnis.

Programmzeilen dazu:

sa 400

sv mittlere Geschwindigkeit

pm

ma -8000000

Nun warten bis das move-Flag des Statuswortes=Low ist und dann

ca 0

senden.

Hintergrundinfo: Sobald ein Endschalter betätigt wird, wird der Motor abrupt angehalten, die Bewegung abgebrochen und das move-Flag des Statuswortes auf Low gesetzt.

rcv

Read Calibration Velocity

Zum Überprüfen des mit "scv" eingestellten Wertes.

sca n

Set Calibration Acceleration

Um ein sanftes Beschleunigen auch bei dem Befehl "ca n" zu gewährleisten. Allerdings wird der Motor gnadenlos gestoppt, wenn ein Endschalter betätigt wird.

rca

Read Calibration Acceleration

Zum Überprüfen des mit "sca" eingestellten Wertes.

saddr n

SI-ADDResse

Durch besondere Maßnahmen auf Seiten unserer SLCAN ist es möglich bis zu 16 SLCAN-Baugruppen an einer seriellen Schnittstelle zu betreiben. Dazu werden alle RXD-Eingänge der SLCAN's mit dem TxD-Ausgang des steuernden Rechners verbunden und alle TxD-Ausgänge werden mit dem RxD-Eingang des Rechners verbunden.

Um die einzelnen SLCAN dennoch unterscheiden zu können ist eine Adressierung der einzelnen Baugruppen nötig. Damit das funktionieren kann erhält zunächst jede einzelne Baugruppe ihre individuelle Adresse.

Schließen Sie zunächst die neu gelieferten Baugruppen einzeln an Ihren Rechner an und geben Sie ihr mit dem Befehl "saddr x" ihre eigene Adresse. Z.B.:

saddr 1

Damit erhält die angeschlossene Baugruppe die Adresse 1. Speichern Sie diese Einstellung mittels

pg

Die neue Einstellung wird erst nach dem nächsten "Power-On" wirksam !

Verfahren Sie so mit allen Baugruppen und notieren Sie die zugeteilte Adresse auf der jeweiligen Baugruppe mit einem kleinen Aufkleber.

Sind alle Baugruppen derart vorbereitet können sie wie oben beschrieben verdrahtet und angeschlossen werden.

Übrigens ist die Baugruppe mit der Adresse 0 die einzige, die sich nach Power-On automatisch mit ihrer Version meldet.

Um nun eine bestimmte Baugruppe auszuwählen verwenden Sie den Befehl:

se n

Dieser Befehl selektiert die Baugruppe mit der Nummer "n". Z.B.:

se 2

Nach dem Senden des Befehls kommuniziert Ihr Rechner mit der Baugruppe 2 solange bis eine andere Baugruppe mittels "se n" angewählt wird.

Es ist also nicht nötig und nicht sinnvoll vor jedem Befehl eine Adressierung vorzunehmen. Eine einmal selektierte Baugruppe bleibt solange selektiert bis eine andere Baugruppe gewählt wird.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Ein Hinweis noch:

Eine Baugruppe im Verbund sollte die Adresse 0 haben, denn nur diese Baugruppe meldet sich automatisch mit ihrer Kennung nach Power-On.

Baugruppen mit anderen Adressen melden sich nicht automatisch.

spwm n

Set PWM

Ab und an ist es nützlich, einen (ungeregelten) Motorstrom erzeugen zu können. Das ist mit diesem Befehl möglich. Voraussetzungen: Es darf weder *pm* noch *vm* eingeschaltet sein. Der Wert *n* hat einen Zahlenbereich von -255 bis +255 und stellt einen "negativen" oder "positiven" Motorstrom ein (entsprechend ändert sich die Drehrichtung des Motors). Bei Nutzung des Befehls ist der PID-Regler nicht eingeschaltet, es erfolgt also weder eine Positions- noch eine Drehzahlkontrolle und es wird keinerlei Rücksicht auf evt. angeschlossene Endschalter genommen. Er kann dazu dienen die Funktion der Endstufen zu überprüfen und kommt bei einer ganz besonderen Anwendung zum Tragen: Dem Straffhalten eines Fadens oder Ähnlichem in einer Wickelvorrichtung.

Vorsicht bei der Wahl des übergebenen Wertes: Kleine, unbelastete Motoren drehen schon bei kleinen Werten um 10! Auch in diesem Modus ist die elektronische Strombegrenzung wirksam (siehe "scl n").

Kleine Motore (speziell bürstenlose) drehen ohne weiteres > 10000UpM. Dann können auf die Achse montierte Bauteile durchaus durch die Luft fliegen und Verletzungen oder Schäden anrichten!

Die Eingabe von *spwm0* oder *st* macht den Motor wieder stromlos. Die Modi *pm* oder *vm* können jederzeit eingeschaltet werden, auch wenn vorher nicht *spwm0* oder *st* gegeben wurde denn dann übernimmt der Regler sofort die Kontrolle und überschreibt den mit *spwm n* eingestellten Wert.

sneglimit n

Dieser Befehl dient dazu das negative Softlimit zu setzen. Der Zahlenwert, der mit *ma* übergeben wird, wird mit dem Wert von *sneglimit* verglichen. Liegt *ma* unter *sneglimit* wird *ma* nicht ausgeführt und es erfolgt eine Fehlermeldung. Nach Power On hat *neglimit* den Wert -33554431 (0xFE000000).

rneglimit

Der Wert der mit *sneglimit* übergeben wurde wird ausgelesen.

sposlimit n

Dieser Befehl dient dazu das negative Softlimit zu setzen. Der Zahlenwert, der mit *ma* übergeben wird, wird mit dem Wert von *sposlimit* verglichen. Liegt *ma* über *sposlimit* wird *ma* nicht ausgeführt und es erfolgt eine Fehlermeldung. Nach Power On hat *poslimit* den Wert 33554431 (0x01FFFFFF).

rposlimit

Der Wert der mit *sposlimit* übergeben wurde wird ausgelesen.

sparama n

Mit diesem Befehl wird ein kundenspezifischer Wert ins interne RAM geschrieben. Der übergebene Wert kann einen Zahlenbereich von $\pm 2^{31}$ haben und wird bei Power On stets auf 0(Null) gesetzt.

Der Zahlenwert wird intern von der SLCAN nicht benutzt und kann als Ablage für einen Wert dienen.

rparama

Auslesen des mit *sparama* gespeicherten Werts.

sparamb n

Siehe *sparama*.

rparamb

Auslesen des mit *sparamb* gespeicherten Werts.

sparamc n

Siehe *sparama*.

rparamc

Auslesen des mit *sparamc* gespeicherten Werts.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Befehle für SLLCD

slcd n

Set LCD type

Der Software wird mit diesem Befehl mitgeteilt, welches LC-Display angeschlossen ist. Standard ist die Lieferung mit einem zweizeiligen Display mit 16 Zeichen pro Zeile (n=216). Bei entsprechender Bestellung ist jedoch auch ein einzeiliges Display mit 8 Zeichen (n=108) und ein vierzeiliges mit je 20 Zeichen (n=420) erhältlich. Um das Display betreiben zu können, muss das Display eingeschaltet werden (siehe "ssyscon"). Gleiches gilt für den frontseitigen Rotationsdekoder.

rlcd

Auslesen des eingestellten LCD-Typs.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Fehlermeldungen

Fehlermeldungen über RS232

Etwaige Fehler die beim Verarbeiten des gegebenen Kommandos auftreten werden prinzipiell als Text auf der RS232 ausgegeben und sind somit unmittelbar lesbar. Zusätzlich zu der Textmeldung wird der String „-1UC“ angehängt. In Anwendungen, in denen die SLCAN über die RS232 betrieben wird sollte das auf dem PC laufende Programm jeden Antwortstring der SLCAN auf das Vorkommen des „-1UC“ untersuchen um festzustellen, ob ein Fehler aufgetreten ist. Zusätzlich ermöglicht der Befehl „rerrno“ festzustellen welcher Fehler auftrat. „rerrno“ liefert folgende Fehlernummern zurück:

Befehl	Fehlerbeschreibung	Fehlernummer
pm	System not in Stopp mode	1
vm	System not in Stopp mode	2
sp	System not in Stopp mode	3
spwm	System not in Stopp mode	4
ca	System not in position mode	5
slcd	Unknown LCD-Type	6
rad	Wrong ad channel(<0/>7)	7
saddr	Addr out of range(<0/>15)	8
	Common Error (Unknown command)	9
ma	Value lower then neglimit	10
ma	Value higher then poslimit	11
ma, mr	Only in position mode	12

Fehlermeldungen über CAN

Die Fehlermeldungen werden im Datenbyte 1 übergeben. Zusätzlich zu den Fehlernummern ist das Bit 7 gesetzt. Beispiel:

Tritt ein Fehler beim Befehl „pm“ auf steht in Datenbyte 1 der Wert 129 (128=Bit7+Fehlernr. 1)

RS232 und CAN

Liegt kein Fehler vor, liefert „rerrno“ 0(Null). Ebenso ist steht in Datenbyte 1 eine 0(Null).

Die Fehlernummer wird auf 0(Null) gesetzt wenn der Befehl „rerrno“ über die RS232 empfangen wurde.

Wird über CAN kommuniziert erfolgt das Löschen der Fehlernummer sofort nachdem die CAN-Antwort auf den „fehlerhaften“ Befehl gesendet wurde.

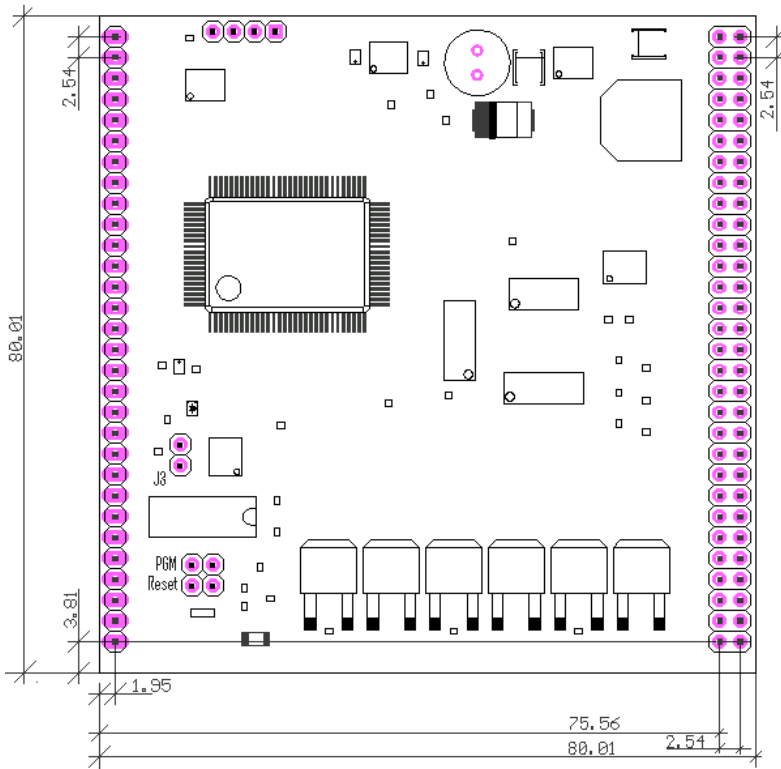
gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Abmaße und Anschlüsse



Jumper

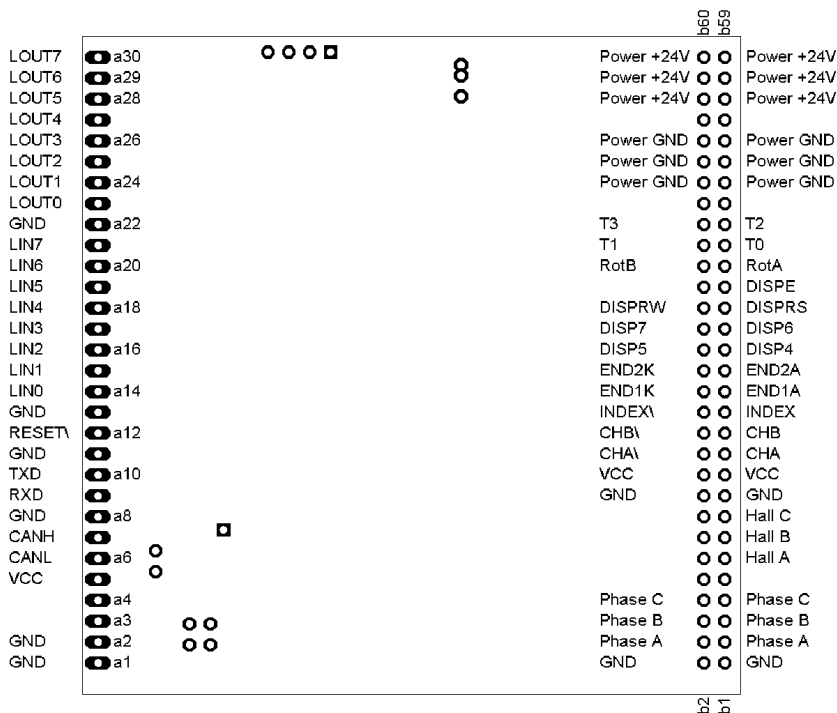
Reset: Das kurzzeitige Brücken der beiden Kontakte löst einen Reset aus.

PGM: Diese beiden Kontakte dienen zum Software-Update.

J3: Mit dem Überbrücken der beiden Kontakte wird der 100 Ohm CAN-Bus Abschlusswiderstand zugeschaltet.

Option LC-Display

Optional lässt sich das Zubehörmodul SLLCD anschließen. Es verfügt neben dem LC-Display (1x8, 2x16 (=Standart) **oder 4x20. Bei Bestellung angeben!**) über 4 Tasten und einen optischen Drehgeber. Zum Anschließen dieses Moduls dienen die Anschlüsse DISP4..DISP7, DISPRS, DISPRW, DISPE, RotA, RotB und T0..T3. Ist die Verwendung des SLLCD nicht vorgesehen, müssen die betreffenden Anschlüsse offen bleiben.



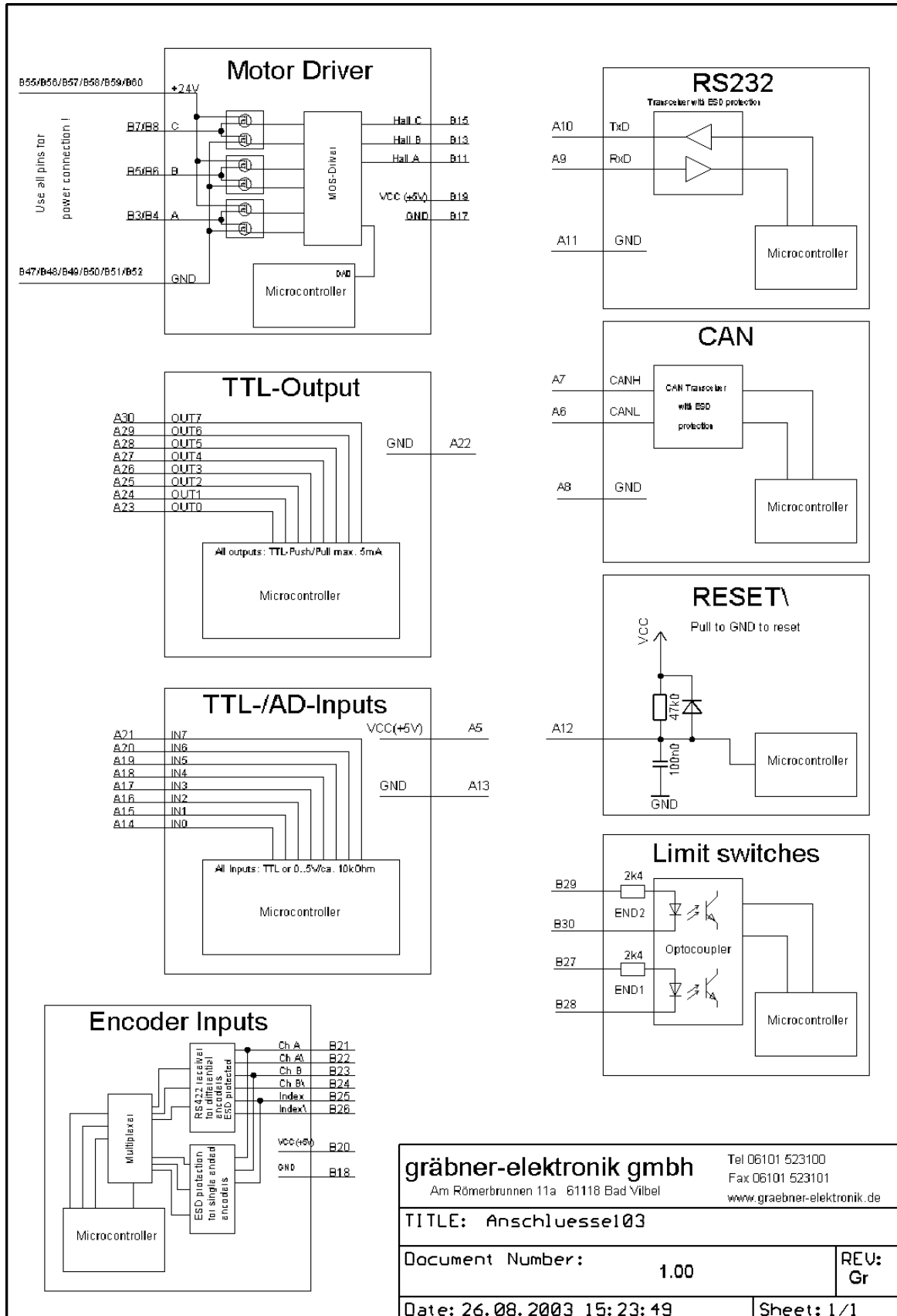
gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Schematics



gräbner-elektronik gmbh Am Römerbrunnen 11a 61118 Bad Vilbel		Tel 06101 523100 Fax 06101 523101 www.graebner-elektronik.de
TITLE: Anschlusse103		
Document Number:	1.00	REV: Gr
Date: 26.08.2003 15:23:49		Sheet: 1/1

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Anhang

Spannungsversorgung

Die SLCAN benötigt nur eine Spannungsversorgung im Bereich 12..24V.

Aus dieser Versorgungsspannung werden alle anderen Hilfsspannungen gewonnen, die auf dem Board benötigt werden. Auch die Motoren werden aus dieser Spannung gespeist. Es ist also darauf zu achten, dass genügend Strom zur Verfügung steht, das Netzteil also entsprechend leistungsfähig ist. Die SLCAN nimmt im Leerlauf ca. 100mA auf. Zu diesem Leerlaufstrom addieren Sie den zulässigen maximalen Strom des Motors und geben eine Reserve hinzu. Diese Reserve richtet sich danach, ob der Motor in Ihrer Applikation mit hohen Beschleunigungswerten verfahren werden muss. Erfahrungsgemäss treten beim Beschleunigen und Verzögern die höchsten Motorströme auf. Wenn Sie sicher gehen wollen, wählen Sie die Reserve so groß wie der maximal zulässige Motorstrom.

Auf dem Board werden zwei Spannungen erzeugt: Die 5V-Logikversorgung und eine 12V-Versorgung zum Ansteuern der Endstufen. Die 5V werden mittels eines DC-DC-Wandlers erzeugt. Der Wandler arbeitet ab einer Versorgungsspannung von ca. 7V. Der Prozessor wird also bereits arbeiten und sich mit seiner Einschaltmeldung bemerkbar machen.

Die Ansteuerstufe für die Endstufen arbeitet jedoch erst bei einer Versorgungsspannung ab ca. 10V. Die Ansteuerstufe überwacht selbständig die Versorgungsspannung und schaltet sich ggf. überhaupt nicht ein. Es kann also vorkommen, dass der Logikteil der SLCAN einwandfrei arbeitet, der Motor jedoch stromlos bleibt. Probleme kann es auch geben, wenn die Versorgungsspannung durch die Belastung des Motors zusammenbricht und unter 10V abfällt. Dann können die verrücktesten Phänomene auftreten: Der PID-Regler (Prozessor) arbeitet munter weiter, der Motor wird jedoch kurzzeitig nicht bestromt, naja, das kann ja nichts werden.

Leider besitzt die Ansteuerstufe keinen Ausgang der anzeigt, ob sie eingeschaltet ist oder nicht. Daher "merkt" der Prozessor auch nichts von dieser "Fehlfunktion" der Ansteuerstufe.

Normalerweise dürfte das Ganze aber relativ unkritisch sein, wenn das Netzteil entsprechend ausgelegt ist. Anders sieht es bei Akku- oder Batteriebetrieb aus. Auch ein 12V-Bleiakku ist irgendwann einmal leer. Daher unser Vorschlag: Benutzen Sie einen der AD-Wandlereingänge (über einen Spannungsteiler) und überwachen Sie die Versorgungsspannung mittels Software. Oder prüfen Sie stets und regelmäßig den Schleppfehler (*pe*). Übersteigt der einen in Ihrer Applikation typischen Wert ist etwas nicht in Ordnung! Motor defekt, Kabel gebrochen oder eben Versorgungsspannung zu niedrig.

Übrigens wird der Motorstrom permanent überwacht und dafür gesorgt, dass er nicht größer als der mit *scl n* werden kann. Dabei handelt es sich um eine rein elektronische Schutzmassnahme die immer nur dann greift, wenn die Bottom-Transistoren (Transistoren nach GND) durchgeschaltet werden. Fließt zu diesem Zeitpunkt ein höherer Strom als von "scl n" vorgegeben, werden diese Transistoren sofort gesperrt. Dieser Vorgang wiederholt sich beim nächsten PWM-Zyklus. Das alles geschieht automatisch durch die Hardware der SLCAN und ist nach außen nur durch das Verlöschen oder Flackern der LED zu erkennen. Außerdem wird der Motor ein merkwürdiges, kratzendes Geräusch erzeugen.

Die mit VCC (+5V) bezeichneten Anschlüsse stellen Ausgänge dar die z.B. zur Versorgung des Winkelenkoders dienen. Wird an diese Klemmen eine externe Spannung angelegt, wird die Baugruppe sofort zerstört und ist nicht mehr reparierbar!

Winkelenkoder

An die SLCAN können fast alle handelsüblichen Enkoder angeschlossen werden. Einzige Einschränkung: Deren Ausgangssignale dürfen 5V nicht überschreiten.

Differenzielle Winkelenkoder arbeiten nach der Norm RS422 und sind direkt anschließbar.

Nichtdifferenzielle Enkoder gibt es in vielen verschiedenen Ausführungen. Wenn sie mit 5V gespeist werden müssen, können sie direkt angeschlossen werden. Ist deren Spannungsversorgung jedoch höher, müssen Sie vor dem Anschluss prüfen, ob sie Signale mit 5V-Pegeln abgeben. Ggf. müssen Sie eine Anpass-Schaltung vorschalten.

Speziell HP liefert einige Enkoder mit Open-Collector-Ausgängen. Auch diese Enkoder können direkt angeschlossen werden. Auf dem Board befinden sich Pull-Up-Widerstände.

Oftmals werden Enkoder geliefert, die mit einer Null-Spur ausgestattet sind. Die Nullspur muss nur dann angeschlossen werden wenn die Funktionen *ca 2*, *ca 3*, *ca 4* oder *ca 5* (siehe dort) verwendet werden sollen. Die Nullspur (auch Index genannt) wird sonst nirgendwo benutzt. Die Positionsinformation wird ausschließlich aus den Spuren A und B gewonnen und niemals mit der Nullspur synchronisiert. Der Anschluss der Nullspur kann also unter den genannten Umständen entfallen.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Welche Einstellungen werden dauerhaft gespeichert ?

Wie aus dem Vorangegangenen ersichtlich, werden verschiedene Einstellungen dauerhaft im internen EEPROM gespeichert und nach einem Power-On wiederhergestellt.

Hier nun eine Liste dieser Einstellungen:

Bezeichnung	Befehl zum Setzen/Auslesen
PID-Parameter P	kp/qp
PID-Parameter I	ki/qi
PID-Parameter D	kd/qd
Positionierfehler-Fenster Grösse	sipw/ripw
Positionierfehler-Fenster Zeit	sipt/ript
Kalibrierdrehzahl	scv/rcv
Kalibrierbeschleunigung	sca/rca
Motorstrombegrenzung	scl/rcf
LC-Display-Typ	slcd/rlcd
Alle Flags/Modi die mit <i>ssyscon</i> eingestellt wurden	ssyscon/rsyscon
CAN Baudrate	scbr/rcbr
CAN Input-Identifizier	sii/rii
CAN Output-Identifizier	soi/roi

Anmerkungen zum Positionierfehler-Fenster (Flag *inpos*)

Sobald der Positioniermodus (*pm*) eingeschaltet wird, wird die Motor-Istposition überwacht. Das Bit *inpos* im Status spiegelt diesen Vorgang wieder.

Zunächst werden intern zwei Grenzwertpositionen gebildet:

- 1) Sollposition – Wert von *sipw* und
- 2) Sollposition + Wert von *sipw*.

Nun wird die Motor-Istposition mit diesen beiden Grenzwerten verglichen. Liegt die Motor-Istposition unter Grenzwert 1 oder über Grenzwert 2 wird das Flag *inpos* zurückgesetzt (auf Null).

Liegt die Motor-Istposition irgendwo zwischen Grenzwert 1 und Grenzwert 2 dann wird ein Zeitzähler gestartet. Dieser Vergleich findet nun in Intervallen von 1ms fortlaufend statt.

Ist der Vergleich positiv (Istpos > Grenzwert 1 < Grenzwert2) dann wird der Zeitzähler um 1 erhöht.

Ist der Vergleich negativ, wird der Zeitzähler auf Null gesetzt.

Erreicht der Zeitzähler den Wert, der mittels *sipt* vorgegeben wurde, wird das Flag *inpos* auf High gesetzt.

Die Gründe dieser Vorgehensweise sind schnell beschrieben: Am Ende einer Bewegung kann es durch schlecht eingestellte PID-Parameter zu einem Schwingen des Motors kommen. In diesem Fall würde der Motor über die gewünschte Sollposition hinausfahren, dann die Drehrichtung umkehren und unter die Sollposition drehen, usw. Damit der Anwender "sicher" sein kann, dass der Motor die gewünschte Sollposition erreicht hat, muss nach dem oben beschriebenen Verfahren der Motor für den Zeitraum von *sipt* innerhalb des Fensters sein das mit *sipw* vorgegeben wurde. Nehmen wir an, mit *sipw* wurde 5 übergeben und mit *sipt* der Wert 100, so muss sich der Motor für mindestens 100ms innerhalb des Fensters von +-5 Positionen um die Sollposition befunden haben wenn das Flag *inpos* gesetzt ist.

Kommunikationsprotokoll (RS232)

Die Kommunikation mit einem übergeordneten Rechner kann über die RS232-Schnittstelle stattfinden. Dazu sind lediglich die Signale TxD (Sendeleitung), RxD (Empfangsleitung) und natürlich GND (Masse) notwendig.

Da keinerlei Handshake-Leitungen benutzt werden, muss das im Folgenden beschriebene Kommunikationsprotokoll unter allen Umständen eingehalten werden, soll es nicht zu Daten-/Zeichenverlusten kommen.

Zunächst: Alle Zeichen die über die Schnittstelle laufen sind ASCII-Zeichen. Numerische Werte die übermittelt werden sollen, müssen als dezimale (hexadezimale) Werte im ASCII-Format übermittelt werden. Der Wert 100 wird also als eine Serie von 3 Zeichen übermittelt: Zuerst eine "1" (ASCII 49) und dann zweimal "0" (ASCII 48).

Die beiden grundlegenden Routinen zur Kommunikation mit der SLCAN bestehen aus dem Senden eines einzelnen ASCII-Zeichens und dem Empfang eines einzelnen Zeichens über die von Ihnen gewählte RS232-Schnittstelle Ihres Rechners.

Die Routine zum Empfang eines Zeichens sollte in jedem Fall über einen Timeout verfügen. Wird innerhalb einer bestimmten Zeit (<200ms) kein Zeichen empfangen, sollte diese Routine mit einer Fehlermeldung

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

abbrechen. Sinnvoll ist ein Rückgabewert von -1, denn alle regulär empfangenen ASCII-Zeichen liegen im Bereich von 0 bis 127.

Testen Sie diese beiden Routinen ausgiebig, sie stellen die Basis für die Kommunikation dar und sind damit elementar.

Damit bei den nachfolgenden Erläuterungen klar ist, welche Routine gemeint ist, verbege ich für die Zeichen-Sende-Routine den Namen "schar" und für die Zeichen-Empfangs-Routine den Namen "rchar". Als nächstes ist eine Routine zu schreiben, die die empfangenen Zeichen analysiert und ggf. in einem String sammelt: Diese Routine sollte alle Zeichen > ASCII 31 in den String schreiben und sich beenden, wenn ein ASCII 13 (CR) empfangen wurde. Diese Routine erhält von mir den Namen "rstring".

Die nun zu schreibende vierte Routine erhält den Namen "sstring" und soll einen Übergabestring auf besondere Art über die RS232 senden. Prinzipiell muss diese Routine die Zeichen des Strings einzeln senden und, nachdem ein Zeichen gesendet wurde, dessen Echo abwarten das die SLCA erzeugt. Also: In einer Schleife wird zunächst das erste Zeichen isoliert und "schar" übergeben. Danach wird "rchar" aufgerufen und das Echo des Zeichens eingesammelt (und wenn gewünscht, mit dem gesendeten Zeichen verglichen). Danach wird das zweite Zeichen mit "schar" gesendet und dessen Echo geholt usw. Ist das letzte Zeichen gesendet und dessen Echo abgeholt, muss ein ASCII 13 (CR) gesendet werden. Auch dessen Echo muss mit "rchar" abgeholt werden.

Die fünfte Routine bekommt den Namen "sbefehl". An sie wird ein beliebiger Befehlsstring übergeben und liefert die von der SLCAN erzeugte Antwort zurück. "sbefehl" übergibt ihrerseits also den Befehlsstring an "sstring" und ruft anschließend sofort "rstring" auf. Der von "rstring" zurückgegebene String wird ohne Bearbeitung von "sbefehl" an das aufrufende Programm zurückgegeben.

Das war's. Wenn Sie sich akribisch an diese Beschreibung halten, wird die Kommunikation mit der Baugruppe praktisch auf Anhieb funktionieren und zwar unter allen praktischen Umständen.

Zum weiteren Verständnis nun Folgendes: Jeder an die SLCAN übergebene Befehl erzeugt einen Antwortstring. Der kann aus einer Zeichenfolge plus CR bestehen oder nur aus CR (ASCII 13 (CR)).

1. Beispiel:

Es soll die aktuelle Motorposition ausgelesen werden. Der Befehl lautet *rp* ohne weiteren Parameter. Also wird "sbefehl" aufgerufen, im Übergabestring steht "rp".

"sbefehl" übergibt den String mit dem Inhalt "rp" an "sstring". "sstring" sendet das erste Zeichen, holt dessen Echo, sendet das zweite Zeichen und holt dessen Echo. Nun wird ein ASCII 13 (CR) gesendet und dessen Echo geholt. "sstring" ruft nun "rstring" auf. "rstring" holt alle Zeichen von der seriellen Schnittstelle und packt sie in den Rückgabestring sofern sie > ASCII 31 sind. Empfängt "rstring" ein CR (13=CR) übergibt "rstring" die empfangenen Zeichen an das aufrufende "sbefehl". "sbefehl" liefert diesen String an Ihr Hauptprogramm in dem Sie die empfangenen Zeichen in einen numerischen Wert wandeln und auswerten können.

2. Beispiel:

Es soll der Positioniermodus eingeschaltet werden. Der Befehl lautet *pm* ohne weitere Parameter.

"sbefehl" übergibt den String mit dem Inhalt "pm" an "sstring". "sstring" sendet die Zeichen einzeln, holt jeweils deren Echo, sendet nach dem zweiten, letzten Zeichen ein ASCII 13 (CR) und holt dessen Echo.

"sstring" ruft nun "rstring" auf. "rstring" holt das erste Zeichen von der seriellen Schnittstelle und würde es in den Rückgabestring packen wenn es grösser als ASCII 31 wäre. Das ist es aber nicht denn dieses Zeichen ist ein ASCII 13 (CR). Damit ist die Endbedingung für "rstring" erreicht. An "sbefehl" wird der Antwortstring auf den Befehl *pm* zurückgeliefert der eine Länge von 0 Zeichen hat – also ein Leerstring. Dieser Leerstring wird natürlich auch an Ihr Hauptprogramm geliefert. Da keine Antwort von *pm* zu erwarten war, kann damit die Auswertung des Antwortstrings unterbleiben.

3. Beispiel:

Es soll der Motor im eingeschalteten Positioniermodus auf Position 1234 verfahren werden. Der Befehl lautet *ma1234*.

"sbefehl" übergibt den String mit dem Inhalt "ma1234" an "sstring". "sstring" sendet die Zeichen einzeln, holt jeweils deren Echo, sendet nach dem letzten Zeichen ein ASCII 13 (CR) und holt dessen Echo. "sstring" ruft nun "rstring" auf. "rstring" holt das erste Zeichen von der seriellen Schnittstelle und würde es in den Rückgabestring packen wenn es grösser als ASCII 31 wäre. Das ist es aber nicht, denn dieses Zeichen ist ein ASCII 13 (CR). "rchar" wird ein ASCII 13 (CR) zurückliefern. Damit ist die Endbedingung für "rstring" erreicht. An "sbefehl" wird der Antwortstring auf den Befehl *ma1234* zurückgeliefert der eine Länge von 0

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Zeichen hat – also ein Leerstring. Dieser Leerstring wird natürlich auch an Ihr Hauptprogramm geliefert. Da keine Antwort von *ma1234* zu erwarten war, kann damit die Auswertung des Antwortstrings unterbleiben.

Dieses Protokoll bleibt auch gültig wenn mehrere SLCAN an einer seriellen Schnittstelle betrieben werden: Angenommen die SLCA mit der Adresse 2 ist augenblicklich selektiert. Nun soll die Karte mit der Adresse 1 selektiert werden. Der Befehl dazu lautet *se1*. Folgendes geschieht intern in der SLCAN: Die drei Zeichen werden zusammen mit dem abschließenden ASCII 13 (CR) empfangen (und geechot). SLCA 2 analysiert den String und ermittelt den übergebenen Wert (die Adresse) und stellt fest, dass dieser Wert nicht mit ihrer eigenen, internen Adresse übereinstimmt. Daraufhin wird die SLCAN 2 sofort ihren Sendekanal abschalten. SLCAN 1 hat diesen String auch empfangen und festgestellt, dass der Wert mit ihrer internen Adresse übereinstimmt. Sie wird nun sofort ihren Sendekanal einschalten und anschließend ein ASCII 13 (CR) senden. Somit ist auch in diesem Fall dem Protokoll genüge getan.

Wenn Sie das Ganze einmal überdenken, werden Sie erkennen, dass das Protokoll ein hohes Maß an Sicherheit bietet und etwaige Fehler leicht erkannt und behandelt werden können. Wenn mittels *se* eine Karte adressiert wird, die es gar nicht gibt, wird der Antwortstring ASCII 13 (CR) ausbleiben. Wenn Sie das Timeout in "rchar" richtig implementiert haben und die darüber liegenden Routinen den Fehlercode (mein Vorschlag war -1) korrekt weitergeben, kann Ihr Hauptprogramm entsprechend reagieren. Gleiches gilt logischerweise auch, wenn eine der RS232-Leitungen gebrochen ist – dann bleibt das Echo auf das erste gesendete Zeichen eines Befehls vollständig aus.

In diesem Zusammenhang noch eine wichtige Anmerkung: Die Baugruppe mit der Adresse 0 meldet sich automatisch nach Power-On ! Damit stehen Zeichen in dem Puffer Ihres Rechners, die nicht mittels eines Befehls angefordert wurden. Das bringt das Ganze natürlich durcheinander. Abhilfe zu schaffen ist aber recht einfach: Rufen Sie "rchar" sooft auf, bis "rchar" den Timeout-Fehlercode zurückliefert. Dann ist der interne UART-Puffer Ihres Rechners garantiert leer.

Nebenbei sei noch angemerkt, das Leerzeichen (ASCII 32) in den SLCA keinerlei Bedeutung haben. Es spielt also keine Rolle, ob Sie z.B. "sv 1000" oder "sv1000" senden.

PID-Parameter

Die Parameter P, I und D haben den gleichen Wertumfang: 0..32767. Nur positive Zahlen sind erlaubt. Die Grundeinstellung ist:

P=40, I=40, D=80

Das ist richtig für einen unbelasteten Motor Typ Faulhaber 2224 mit 512er Encoder.

Die Steifigkeit des Motors wird mit P verändert.

Die Schwingneigung wird mit D unterdrückt.

I verschafft Positioniergenauigkeit am Zielort.

Parameter vorsichtig ändern, all zu große Wertänderungen sind meist nicht sinnvoll, besser ist es, sich langsam an das "Optimum" heranzutasten. Folgende Vorgehensweise kann empfohlen werden:

I-Anteil abschalten bzw. stark verringern ("KI0" oder "KI1"). Nun ist die Positioniergenauigkeit sehr schlecht, aber der D- und P-Anteil wird nicht mehr vom I-Anteil beeinflusst. D-Anteil und P-Anteil erhöhen, bis die gewünschte oder machbare Steifigkeit erreicht ist. Um dem Schwingen vorzubeugen ggf. D-Anteil variieren. Ist man mit P und D zufrieden, wird der I-Anteil solange erhöht bis auch die Positioniergenauigkeit stimmt. Um Veränderungen des Regelverhaltens feststellen zu können muss zwischen den einzelnen Schritten der Parameterermittlung der Motor immer wieder verfahren werden, nur dann bemerkt man die vorgenommenen Veränderungen.

Dieser Vorgang ist nicht sonderlich schwierig, ich empfehle aber, sich einen sehr ruhigen Raum zu suchen (damit man den Motor hört) und sich 30 Minuten Zeit zu nehmen. Danach hat man ein Gefühl für den Motor und die Regelung und kennt die Grenzen von Beiden. Weitere Nachbesserungen gehen dann leicht von der Hand. Vorteilhaft ist auch, den Motor in die Hand zu nehmen. Dabei spürt man geringste Schwingungen, denn, egal was man will, der Motor muss ruhig und gleichmäßig laufen, das ist oberste Maxime. Übrigens gibt es Applikationen bei denen es besser ist, die Steifigkeit des Systems zu verringern. Das wirkt sich vorteilhaft auf die Geschwindigkeitsregelung aus, wenn mehr Gleichmäßigkeit als Drehzahlgenauigkeit gefragt ist. Im Positioniermodus bringt das nur dann etwas, wenn sich das System ein wenig federnd verhalten soll. Man sollte jetzt aber nicht auf die Idee kommen mit den Motoren und der Regelung eine

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Feder nachzubilden. Der Regelung ist das egal, aber der Motor macht es nicht lange, wenn die Bürsten immer über ein und dieselbe Stelle kratzen.

Verdrahtungshinweise

Der angeschlossene Motor wird über eine PWM-Endstufe versorgt. Dabei werden z.T. erhebliche Ströme geschaltet die erhebliche Störungen verursachen können.

Aus diesem Grund müssen die Motorzuleitungen über ein von allen anderen Leitungen getrenntes und abgeschirmtes Kabel mit der SLCAN verbunden werden.

Alle anderen Anschlüsse wie Winkelenkoder, Endschalter und ggf. Hallsensoren können gemeinsam in einem Kabel geführt werden das ebenfalls geschirmt sein sollte.

Folgt die Verkabelung nicht diesen Empfehlungen ist ein ordnungsgemäßer Betrieb des Motors nicht zu erwarten.

CAN-Modul Beschreibung V1.00

CAN 2.0A

Der CAN-Identifizier von CAN 2.0A hat eine Länge von 11Bit.

Eingestellt wird dieser Modus vorzugsweise über die RS232. Dazu dient der Befehl `ssyscon`, das Bit `canb` muss auf 0 (Null) gesetzt werden. Diese Änderung wird sofort wirksam.

CAN 2.0B

Der CAN-Identifizier von CAN 2.0B hat eine Länge von 29Bit.

Eingestellt wird dieser Modus vorzugsweise über die RS232. Dazu dient der Befehl `ssyscon`, das Bit `canb` muss auf 1 (Eins=High) gesetzt werden. Diese Änderung wird sofort wirksam.

CAN Baudrate

Die Baudrate lässt sich mit dem Befehl `scbr n` einstellen.

N=0 1Mbit

N=1 500kBit

N=2 250kBit

N=3 125kBit

Die Änderung wird sofort wirksam.

Zur Kontrolle der eingestellte Baudrate dient `rcbr`. Der mit `scbr n` eingestellte Wert wird zurückgeliefert.

CAN Identifizier

Für das Modul SLCAN sind zwei Identifizier zu reservieren. Einer dient dem Modul zum Empfang von CAN-Nachrichten (Input-Identifizier), der andere wird zum Senden verwendet (Output-Identifizier). Beide Identifizier können auf beliebige Werte gesetzt werden.

Der Input-Identifizier wird mit `sii n` gesetzt, der Output-Identifizier mit `soi n`.

Diese Einstellungen werden sofort wirksam.

Die Einstellungen können mit `rii` bzw. `roi` ausgelesen und kontrolliert werden.

Auslieferungszustand

Die CAN-Baudrate ist auf 500kB eingestellt.

Der Input-Identifizier ist auf 0x100, dezimal 256 gesetzt.

Der Output-Identifizier ist auf 0x0101, dezimal 257 gesetzt.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Bitte beachten Sie, dass Ihre vorgenommenen Änderungen lediglich temporär sind, sie gehen mit Power-Off verloren. Möchten Sie Ihre Änderungen dauerhaft speichern, benutzen Sie bitte den Befehl *pg* !

CAN Datenformat

ID (11/29 Bit)	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5
----------------	-----	--------	--------	--------	--------	--------	--------

Schematischer Aufbau einer CAN-Nachricht

Empfang

Alle Befehle bzw. Kommandos werden mit dem Input-Identifizier der SLCAN empfangen und müssen aus sechs Datenbytes bestehen.

Das nullte Datenbyte kennzeichnet den auszuführenden Befehl.

Das erste Datenbyte ist reserviert und sollte stets Null sein.

Das zweite bis fünfte Datenbyte beinhaltet den zum Befehl nötigen Parameter, wobei Datenbyte 2 das hochwertigste und Datenbyte 5 das niederwertigste Byte darstellt.

Senden

Alle vom Modul gesendeten Nachrichten benutzen den Output-Identifizier und bestehen aus sechs Datenbytes. Nachrichten werden nur nach dem Empfang eines Befehls gesendet. Das Modul sendet in keinem Fall selbständig.

Im nullten Byte ist die Kennung des Befehls der diese Nachricht ausgelöst hat enthalten.

Das erste Byte dient zum Anzeigen interner Informationen und ist bit-kodiert.

Z.Z. wird dieses Byte zum Senden von Fehlermeldungen benutzt (siehe Fehlermeldungen). In den Bytes 2 bis 5 steht der angeforderte Wert. MSB ist Byte 2, LSB Byte 5.

CAN Protokoll

Zunächst gilt: Jeder über den Input-Identifizier empfangene Befehl löst das Senden einer Nachricht mit dem Output-Identifizier aus, unabhängig vom benutzten Befehl.

Beispiele:

Der Befehl *pm* (Wert 1) schaltet den PID-Regler im Positioniermodus ein. Daten werden von diesem Befehl nicht geliefert, trotzdem wird eine Nachricht gesendet. Im nullten Byte steht die Kennung von *pm*, im 1. eine Null wie auch in den Bytes 2 bis 5.

Der Befehl *rp* (7) liest die augenblickliche Motorposition aus. Das 0. Byte der Antwort enthält wieder die Kennung von *rp* (7), das 1. Byte ist Null und im 2. bis 5. Byte wird die Motorposition übermittelt.

gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

Tel.: 06101/523100 • Fax: 06101/523101

eMail: info@graebner-elektronik.de • Internet <http://www.graebner-elektronik.eu>

Befehlskennungen

Im ersten Byte ist die Kennung des gewünschten Befehls einzutragen.

Kennung	Befehl	Kommentar	Kennung	Befehl	Kommentar
1	pm	Position Mode	38	ssyscon	Set SYStem CONfiguration
2	vm	Velocity Mode	39	rsyscon	Read SYStem CONfiguration
3	st	Stop	40	rcal	Read CALibration flag
4	spwm	SetPWM	41	rad	Read AD converter channel
5	rpwm	ReadPWM	42	rin	Read INputs digitally
6	ma	Move Absolut	46	scbr	Set Can BaudRate
7	mr	Move Relativ	47	rcbr	Read Can BaudRate
8	rp	Read Position	48	sii	Set Input Identifier
9	sp	Set Position	49	rrii	Read Input Identifier
10	ca	Calibrate	50	soi	Set Output Identifier
11	sv	Set Velocity	51	roi	Read Output Identifier
12	rv	Read Velocity	52	rerrno	Read Error Number
13	scv	Set Calibration Velocity	53	spel	Set Position Error Limit
14	rcv	Read Calibration Velocity	54	rpel	Read Position Error Limit
15	sa	Set Acceleration	55	stp	Set Target Position
16	ra	Read Acceleration	56	rtp	Read Target Position
17	sca	Set Calibration Acceleration	57	go	Go to Target Position
18	rca	Read Calibration Acceleration	58	sneglimit	Set Negativ Position Limit
19	kp	set Parameter P	59	rneqlimit	Read Negativ Position Limit
20	qp	Query Prameter P	60	sposlimit	Set Positiv Position Limit
21	ki	set Parameter I	61	rposlimit	Read Positiv Position Limit
22	qi	Query Prameter I	62	sparama	Set Parameter A
23	kd	set Parameter D	63	rparama	Read Parameter A
24	qd	Query Prameter D	64	sparamb	Set Parameter B
25	pe	read Position Error	65	rparamb	Read Parameter B
26	ss	StatuS	66	sparamc	Set Parameter C
27	id	Identification	67	rparamc	Read Parameter C
28	slcd	Set LCD type			
29	rlcd	Read LCD Type			
30	sipw	Set In Position Window			
31	ripw	Read In Position Window			
32	sipt	Set In Position Time			
33	ript	Read In Position Time			
34	sl	Set Limits			
35	rl	Read Limits			
36	sil	Set Inver Limits			
37	ril	Read Invert Limits			

Die meisten Befehle verhalten sich über CAN-Bus und RS232 gleich. Es gibt jedoch eine Ausnahme:

id IDentification

In Datenbyte 2 Steht die Softwareversion, Byte 3 ist stets Null, in Byte 4 steht das MSB und in Byte 5 das LSB der Seriennummer.