

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 SLBL V1.12 (Betriebssystemebene)

Gültig ab Hardware-Version 2.18

Allgemeines.....	3
Anforderungen an den Motor	3
Inbetriebnahme	3
LED-Betriebsanzeige	4
Befehlstabelle	5
pm	6
vm	6
st	6
ma n	6
mr n	6
sv n	6
rv	6
sa n	6
ra	7
kp n / ki n / kd n	7
qp / qi / qd	7
sp n	7
rp	7
ca n	7
scv n	9
rcv	9
sca n	9
rca	9
pe	9
ssyscon n	9
rsyscon	10
rrsyscon	10
sipw n	10
ripw	10
sipt n	10
ript	10
ss	10
Auswertung der Flags	11
rin	12
sout n	12
id	12
pg	12
sl	12
rl	12
sil	13
ril	13
scv	13
rcv	13
sca n	13
rca	13
scl n	14
rcl	14
spwm	14
sla n	14
se n	14

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>

Lage der Anschlüsse	16
ANHANG	18
Allgemeines, Zusammenhänge und Hintergründe	18
Endschalteranschluss	18
Spannungsversorgung	19
Winkelenkoder	19
Welche Einstellungen werden dauerhaft gespeichert ?	20
Anmerkungen zum Positionierfehler-Fenster	20
Kommunikationsprotokoll	21
PID-Parameter	23
Parameter-Einstellhilfe	23
Verdrahtungshinweise	24

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

Merkmale SLBL

- Betrieb von bürstenlosen und bürstenbehafteten DC-Motoren
- Eine Versorgungsspannung im Bereich von 12..30V=
- Digitaler PID-Regler
- Positionierbereich +-33554431 (+-2²⁵)
- Maximaler Motorstrom ca. 5A (interne Strombegrenzung)
- Eingänge für differenzielle und nicht-differenzielle Winkelenkoder mit und ohne Null-Spur
- Zwei optogekoppelte Endschaltereingänge
- Sechs optogekoppelte Universal-Eingänge
- Zwei optogekoppelte Universal-Ausgänge
- Anschlüsse für LC-Display, 4 Taster und digitalem Poti (Optischer Drehgeber)
- Host-Anschluss über 9pol-SubD-Verbinder, alle anderen Anschlüsse über Schraubklemmen
- Platinenformat: ca. 100*140*25mm³

Die Baugruppe dient zur exakten Positionierung und Drehzahlregelung kleiner DC-Motoren bis zu einer Leistung von ca. 100W. Integriert ist ein digitaler PID-Regler dessen Parameter jederzeit anpassbar sind. Die Sollwert- und Parametervorgaben erfolgen über eine serielle RS232-Schnittstelle. Darüber erfolgen alle Vorgaben für den Motorbetrieb:

Positionier- oder Drehzahlregelbetrieb

Beschleunigung/Verzögerung

Anzufahrende Position

Aktivierung/Deaktivierung der Endschalter

Achskalibrierung (Auf Endschalter und/oder Nullspur)

Abfrage der Ist-Position des Motors usw.

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 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.

Inbetriebnahme

Um mit der Baugruppe arbeiten zu können muss sie zunächst korrekt angeschlossen und konfiguriert werden.

Folgen Sie dabei den Anweisungen die in unserem Dokument "InbetriebSLBL0100.rtf" gegeben werden um die Baugruppe korrekt zu konfigurieren.

Fahren Sie nach der Konfiguration fort:

Schalten Sie die Stromversorgung ab und schließen Sie den Motor an.

Sorgen Sie dafür, dass der Motor frei drehen kann. Koppeln Sie notfalls die Mechanik ab. Das ist sehr wichtig ! Sie könnten sonst die Mechanik zerstören !

Schalten Sie die Spannungsversorgung wieder ein. Wieder müssen Sie die Einschaltmeldung lesen können (grüne LED leuchtet).

Geben Sie zunächst den Befehl "pm" ein.

Jetzt kann es passieren, dass der Motor mit hoher Drehzahl plötzlich losläuft nachdem Sie den Rotor mit der Hand verdreht haben. Das ist nicht weiter schlimm (wenn Sie die Mechanik abgekoppelt haben). Schalten Sie die Spannungsversorgung ab und vertauschen Sie die Enkoder-Anschlüsse für Spur A und Spur B. Schalten Sie wieder ein und versuchen Sie nochmals "pm" und das Verdrehen des Rotors. Nun müsste der Motor ruhig stehen und sich gegen die Kraft wehren mit der Sie den

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>

Rotor verdrehen, jedenfalls darf er nicht mit hoher Drehzahl drehen. Steht der Motor war's das. Sehen Sie sich die verfügbaren Befehle an und probieren Sie sie aus.

Dreht der Motor immer noch mit hoher Drehzahl, überprüfen Sie alle Motoranschlüsse und wiederholen Sie die gesamte Prozedur ab Motoranschluss.

Hintergrundinformation zur Fehleranalyse:

Nach "pm" wird der Regler eingeschaltet und der Motor bestromt wenn der Rotor verdreht wird. Wenn der Drehsinn des Motors nicht mit dem Drehsinn des Encoders übereinstimmt kommt es dazu, dass der Motor mit hoher Drehzahl dreht.

Ein anderes Phänomen tritt auf, wenn eines der Encodersignale (A oder B) fehlt. Dann kann die Elektronik nicht erkennen, dass sich der Rotor bzw. der Encoder gedreht hat. Damit bleibt der Motor stromlos und man kann den Rotor verdrehen obwohl der Regler mit "pm" eingeschaltet wurde.

Das gleiche Phänomen tritt auf, wenn Sie einen bürstenbehafteten Motor besitzen, in der Konfiguration aber "bürstenlos" verwendet haben.

Sie können jederzeit überprüfen, ob der Encoder angeschlossen ist. Das geht einfach mit dem Befehl "rp". Er gibt die aktuelle Encoderposition zurück. Der interne Positionszähler läuft, solange die Spannungsversorgung eingeschaltet ist, vollkommen unabhängig vom Betriebsmodus.

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 "rcl".

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>

Befehlstabelle

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

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 AUF Position <i>n</i> verfahren
mr <i>n</i>	Move Relative	Motor UM <i>n</i> Positionen verfahren
sv <i>n</i>	Set Velocity	Drehzahl setzen
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
qp	Query P	Auslesen des Reglerparameters P
qi	Query I	Auslesen des Reglerparameters I
qd	Query D	Auslesen des Reglerparameters D
sp <i>n</i>	Set Position	Setzen der Motorposition
rp	Read Position	Auslesen der Motorposition
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
pe	Position Error	Schleppfehler auslesen
ssyscon <i>n</i>	Set Configuration	Hard-/Stftware konfigurieren
rsyscon	Read Configuration	Auslesen der Konfiguration
rrsyscon	Read Configuration	Auslesen der Konfiguration in Klartext
sipw <i>n</i>	Set InPos Window	inpos-Fenster setzen
ripw	Read InPos Window	Gesetzten Wert auslesen
sipt <i>n</i>	Set InPos Timer	Zeitgeber für inpos setzen
ript	Read InPos Timer	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 auslesen
rin	Read Inputs	Eingänge abfragen
sout <i>n</i>	Set OUTputs	Ausgänge setzen
id	IDentification	Versionsmeldung abfragen
pg	ProGram	Einstellungen im EEPROM speichern
sla <i>n</i>	SerialLink Address	Setzen der Baugruppen-Adresse (0..15)
se <i>n</i>	SElect	Baugruppe mit Adresse <i>n</i> selektieren
scl <i>n</i>	Set Current Limit	Motorstrombegrenzung (<i>n</i> =mA)
rcl	Read Current Limit	Einstellung der Motorstrombegrenzung auslesen
spwm <i>n</i>	Set PWM	Ungeregelten Motorstrom erzeugen (<i>n</i> =-255..+255)

Hinweis:

Die Ausführung eines Befehls kann verhindert werden wenn SLBL VOR dem Return-Zeichen ein Strg-X (dezimal 24, hex \$18) empfängt. Mit diesem Kontrollzeichen wird der interne Empfangspuffer geleert.

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>

pm

Position Mode

Damit wird der Regler eingeschaltet und versuchen die augenblickliche Motorposition zu halten.

vm

Velocity Mode

Der Regler wird eingeschaltet und der Motor sofort mit der eingestellten Beschleunigung auf die eingestellte Drehzahl beschleunigt. Währenddessen kann jederzeit die Drehzahl verändert werden.

st

STop

Der Regler wird abgeschaltet, der Motor wird stromlos. Der eingestellte Modus (pm oder vm) wird verlassen/beendet.

ma n

Move Absolute

Der Motor wird auf die mitübergebene 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 sein.

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 sein.

sv n

Set Velocity

Setzen der (maximalen) Verfahrgeschwindigkeit/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 Enkoders

$ve = (UpM * Linien) / 234,37$

$UpM = (ve * 234,37) / Linien$

Beispiel:

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

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

rv

Read Velocity

Auslesen der eingestellten (maximalen) Verfahrgeschwindigkeit bzw. Drehzahl.

sa n

Set Acceleration

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>

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 Enkoders

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

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 $\pm 2^{25}$.

ca n

CAlibrate

Nach dem Einschalten der Spannungsversorgung hat der interne Positionsähler 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 SLBL lassen sich maximal zwei Endschalter und ggf. der Index-Puls des Winkelenkoders anschließen.

Hinweise:

- 1) Nach erfolgreicher Kalibration wird ein Flag im Statusregister gesetzt. Siehe „ss“, Flag **cal**.

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>

- 2) Ein mit „ca x“ eingeleiteter Kalibriervorgang wird abgebrochen wenn das Zeichen Strg-K (dezimal 11, Hex \$0b) empfangen wurde. In diesem Fall bleibt das Flag **cal** im Statusregister auf 0 (Null).

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

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.

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.

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 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.

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 übrigens 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	BPgmStart	High=Basic Pgm Start nach Power On

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	BlntStart	High=Interpreter Start nach Power On
2	4	LCDOn	High=LC-Display eingeschaltet
3	8	RotOn	High=Rot. Encoder eingeschaltet
4	16	ADCOOn	High=ADC eingeschaltet
5	32	BLOn	Motor: High=bürstenlos Low=bürstenbehaftet
6	64	EncType	Enkoder: High=differentiell (RS422) Low=Single ended
7	128	SignOn	High=Drehrichtung an OUT1
8	256	ADCRange	ADC Bereich High=0..10V Low=-5...+5V
9	512	L1On	High=Endschalter 1 wird berücksichtigt
10	1024	L2On	High=Endschalter 2 wird berücksichtigt
11	2048	L1Inv	High=Eingang Endschalter 1 wird invertiert
12	4096	L2Inv	High=Eingang Endschalter 2 wird invertiert
13	8192	HexOn	High=Alle Zahlenausgaben in hexadezimaler Schreibweise

rsyscon

Read SYStem CONfiguration

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

rssystemcon

Read SYStem CONfiguration

Achtung: Dieser Befehl darf nur benutzt werden, wenn die Kommunikation mit der SLBL über ein Terminalprogramm (z. B. GrTermW) erfolgt !

Der Befehl listet die Konfiguration der SLBL in Klartext auf. Die Ausgabe ist mehrzeilig !!!!

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 "sipw n" gesetzt wurde.

sipt n

Set InPos Tlmer

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 Tlmer

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

ss

Status

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

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

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 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	overtemp	High=Übertemperatur der Endstufe detektiert
Bit 7	128	Cal	High=Achse kalibriert / Low=Nicht kalibriert

limit1 / limit 2

Durch Auswertung der beiden Flag' 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.

overtemp

Low=Endstufentemperatur unter 80°C

High=Endstufentemperatur über 80°C. Sobald während des Drehzahlregel- oder Positioniermodus auf Übertemperatur erkannt wird, wird die Endstufe und der entsprechende Modus (pm oder vm) abgeschaltet. Für einen externen Rechner ist das mit diesem Bit prüfbar. Sobald die Endstufe auf unter 80°C abgekühlt ist, kann der Motor wieder mittels "pm" oder "vm" eingeschaltet werden.

cal

Das Flag wird bei Power-On auf 0 (Null) gesetzt. Es wird nur dann 1 (High) wenn die Kalibrierung mittels „cal x“ erfolgreich durchgeführt wurde.

Das Flag dient dem Rechner der die Achse steuert dazu, zu erkennen ob die Achse bereits kalibriert ist oder nicht.

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.

rss

Read StatuS

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>

Achtung: Dieser Befehl darf nur benutzt werden, wenn die Kommunikation mit der SLBL über ein Terminalprogramm (z. B. GrTermW) erfolgt !

Der Befehl listet die Status-Flag's der SLBL in Klartext auf. Die Ausgabe ist mehrzeilig !!!!

rin

Read INputs

Auch dieser gelieferte Wert ist binär kodiert:

Bit-Nr.	Dezimaler Wert	Flag-Name	Bedeutung
Bit 0	1	Eingang 0	High=Eingang offen / Low=Eingang gegen GND geschaltet
Bit 1	2	Eingang 1	High=Eingang offen / Low=Eingang gegen GND geschaltet
Bit 2	4	Eingang 2	High=Eingang offen / Low=Eingang gegen GND geschaltet
Bit 3	8	Eingang 3	High=Eingang offen / Low=Eingang gegen GND geschaltet
Bit 4	16	Eingang 4	High=Eingang offen / Low=Eingang gegen GND geschaltet
Bit 5	32	Eingang 5	High=Eingang offen / Low=Eingang gegen GND geschaltet

sout n

Es stehen zwei Ausgänge zur Verfügung. Die Ausgangsstufe ist als "Open Colector" ausgeführt, d.h. der Kollektor des Ausgangstransistors liegt auf der Klemmleiste, der Emitter auf GND.

Bit-Nr.	Dezimaler Wert	Flag-Name	Bedeutung
Bit 0	1	Ausgang 0	High=Ausgang hochohmig / Low=Ausgang gegen GND geschaltet
Bit 1	2	Ausgang 1	High=Ausgang hochohmig / Low=Ausgang gegen GND geschaltet

Um Ausgang 0 auf GND-Potential und Ausgang 1 hochohmig zu schalten: *sout 2*

Um Ausgang 1 auf GND-Potential und Ausgang 0 hochohmig zu schalten: *sout 1*

Um Ausgang 0 und Ausgang 1 auf GND-Potential zu schalten: *sout 0*

Um beide Ausgänge hochohmig zu schalten: *sout 3*

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

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

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 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

Siehe dazu auch **Endschalteranschluss** im Anhang!

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, nach "sil 0" wird erwartet, dass die Endschalter als Öffner zwischen dem betreffenden Eingang und GND geschaltet sind. 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. Auch dieser Kontakt muss zwischen dem betreffenden Eingang und GND geschaltet werden, nur liefert er 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 sie 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.

scv

Set Calibration Velocity

Während der Kalibrierung ("ca n") wird diese besondere Kalibriergeschwindigkeit benutzt. Sie sollte im Interesse der Kalibrierengenauigkeit deutlich unter der sonst benutzten Verfahrensgeschwindigkeit 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 der betreffende Endschalter betätigt wird.

rca

Read Calibration Acceleration

Zum Überprüfen des mit "sca" eingestellten Wertes.

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>

scl n

Set Current Limit

Ab Hardware-Version 2.17 ist eine elektronische Motorstrombegrenzung implementiert. Der mit „scl“ zu übergebende Wert *n* darf zwischen 50 und 4000 liegen und gibt an, bei welchem Motorstrom (in Milliampere) die Begrenzung einsetzen soll. Auch dieser Wert wird im EEPROM gespeichert und nach Power Off/Power On wieder hergestellt.

Beispiel für eine Motorstrombegrenzung von 1500mA:

scl 1500

Hinweis: Die (grüne) LED auf der Leiterplatte signalisiert durch ihr Verlöschen oder Flackern, dass die Strombegrenzung wirksam ist.

rcl

Read Current Limit

Zum Überprüfen des mit “scl” eingestellten Wertes.

spwm

Set PWM

Mit diesem Befehl (Wertebereich –255 bis +255) kann ein unregelmäßiger Motorstrom erzeugt werden. Bei bürstenbehafteten Motoren genügt der Anschluss der Motorzuleitungen. Bei bürstenlosen Motoren sind die Motorwicklungen und die Hallgeber mit ihrer Stromversorgung anzuschließen. Der Anschluss der Winkelenkodern ist nicht zwingend notwendig.

Der Befehl kann dazu dienen Motoren zu überprüfen, oder in speziellen Fällen ein (unregelmäßiges) Drehmoment zu erzeugen – nützlich um zu verhindern, dass sich eine Spule abwickelt.

sla n

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

Um die einzelnen SLBL 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 “sla x” ihre eigene Adresse. Z.B.:

sla 1

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

Die neue Einstellung ist 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 gezielt gewählt wird.

Ein Hinweis noch:

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>

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.

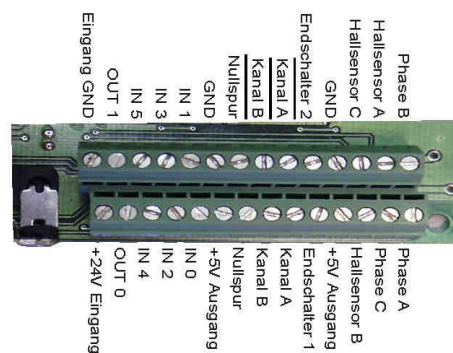
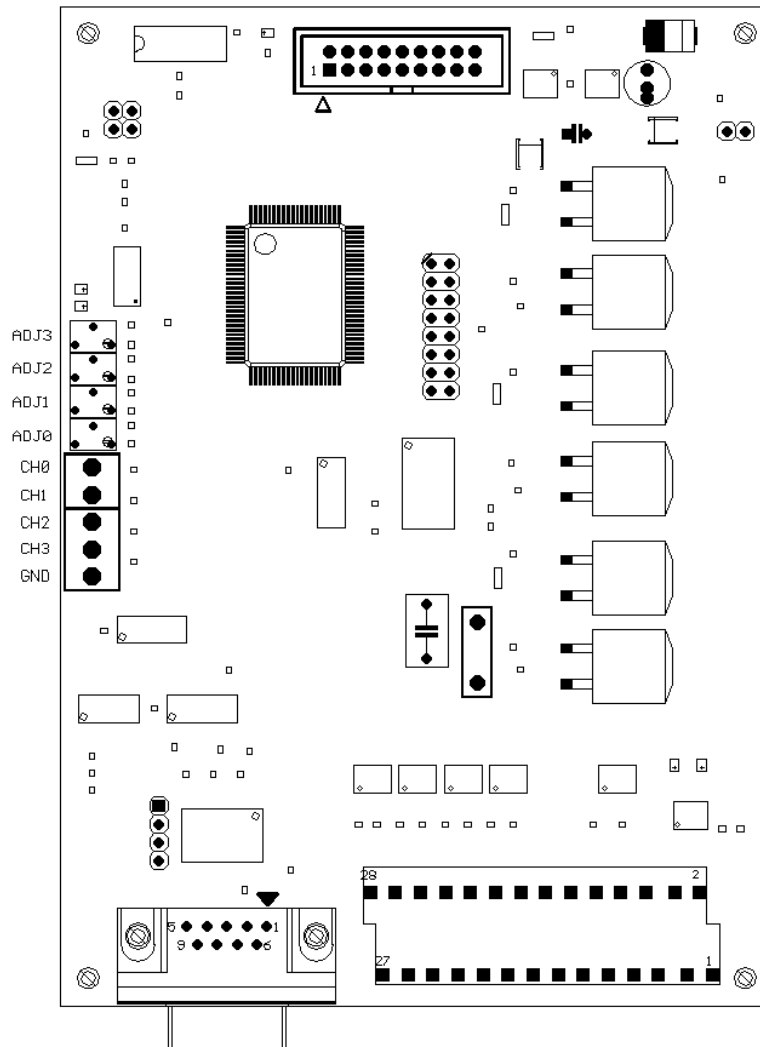
gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

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

Lage der Anschlüsse



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>

Signale am Klemmenblock

Klemme	Signal
1	Motor-Phase A
2	Motor-Phase B (Anschluss nur für bürstenlose Motoren)
3	Motor-Phase C
4	Hallsensor A (Anschluss nur für bürstenlose Motoren)
5	Hallsensor B (Anschluss nur für bürstenlose Motoren)
6	Hallsensor C (Anschluss nur für bürstenlose Motoren)
7	+5V Encoder (Ausgang!)
8	GND Encoder
9	Eingang Endschalter 1
10	Eingang Endschalter 2
11	Encoder Kanal A
12	Encoder Kanal A invertiert (Anschluss nur bei differenziellem Encoder)
13	Encoder Kanal B
14	Encoder Kanal B invertiert (Anschluss nur bei differenziellem Encoder)
15	Encoder Nullspur
16	Encoder Nullspur invertiert (Anschluss nur bei differenziellem Encoder)
17	+5V für Hallsensoren (Ausgang!)
18	GND Hallsensoren
19	Digitaler Eingang 0
20	Digitaler Eingang 1
21	Digitaler Eingang 2
22	Digitaler Eingang 3
23	Digitaler Eingang 4
24	Digitaler Eingang 5
25	Digitaler Ausgang 0
26	Digitaler Ausgang 1
27	+12..30V Spannungsversorgung
28	GND Spannungsversorgung

Bürstenbehaftete Motoren

werden an Motor-Phase A (Klemme 1) und Motor-Phase C (Klemme 3) angeschlossen. Klemme 2 (Motor-Phase B) bleibt ohne Beschaltung.

Die Hallsensoranschlüsse

werden nur bei bürstenlosen Motoren benötigt. Bei Verwendung eines bürstenbehafteten Motors dürfen die Anschlüsse nicht beschaltet werden !

Nicht-differenzielle Winkelenkoder

werden an die Klemmen 12 und 14 angeschlossen. Falls der Encoder über eine Nullspur verfügt kann diese an Klemme 16 angeschlossen werden.

Differenzielle Winkelenkoder

werden an die Klemmen 12/11 und 14/13 angeschlossen. Falls der Encoder über eine Nullspur verfügt kann diese an Klemme 16/15 angeschlossen werden.

Die digitalen Eingänge

führen auf die Kathoden der LED's in den Optokopplern. Deren Anoden liegen über 2,4kOhm an den internen +5V.

Wenn diesen Eingängen eine externe Spannung von >5V zugeführt wird, werden die Optokoppler unweigerlich zerstört !

Bei Verwendung der Display-/Tastenplatine dürfen die Eingänge 0 und 1 nicht extern beschaltet werden ! Auf diesen beiden Eingängen liegt dann der digitale "Poti" !

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>

Die digitalen Ausgänge führen jeweils auf den Kollektor des NPN-Transistors in den Optokopplern. Deren Emittoren sind mit GND verbunden. Wird ein Ausgang via Software aktiviert, wird der interne Transistor durchgeschaltet, der Kollektor also auf GND gezogen. Maximaler Kollektorstrom ca. 5mA, Spannungsfestigkeit 30V.

Die Spannungsversorgung

für die Baugruppe wird nur über die Klemmen 27 und 28 hergestellt.

*Das ist die einzige zuzuführende Spannung und sie ist aus technischen Gründen **nicht verpolsicher!***

Aus dieser Versorgung wird die Motorendstufe gespeist und die intern benötigten +5V mittels Schaltregler erzeugt.

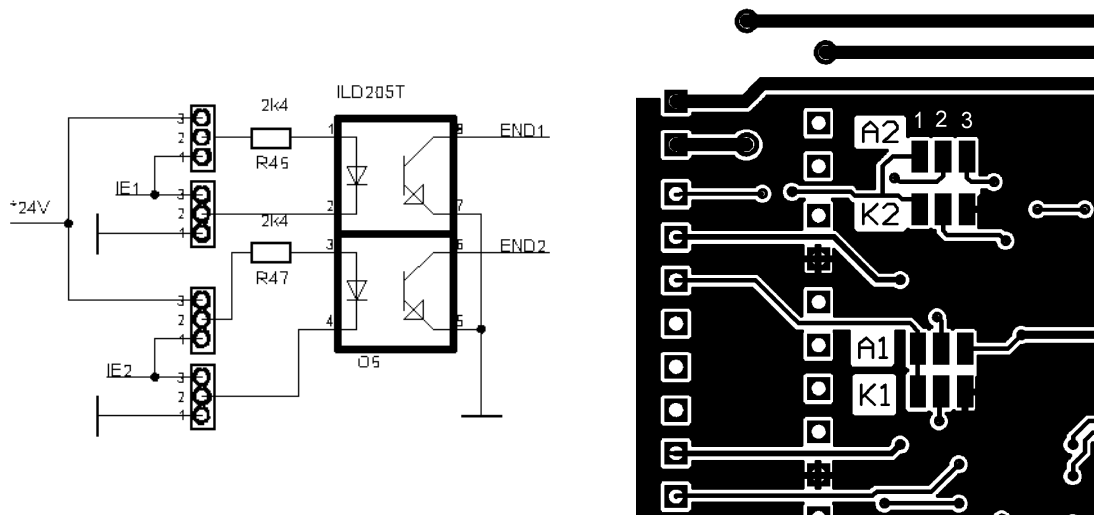
ANHANG

Allgemeines, Zusammenhänge und Hintergründe

Hier werden einige Zusammenhänge innerhalb der Baugruppe beschrieben die zum besseren Verständnis der internen Abläufe dienen soll.

Endschalteranschluss

Mit der Hardware-Version 2.18 ist es möglich, nahezu beliebige Endschalter anzuschließen. Das Schaltbild:



Die Pad's sind rechts sichtbar. Pad 1 der Pad-Gruppen ist jeweils das linke Pad.

Pad 2 der Pad-Gruppen ist jeweils das mittlere Pad und Pad 3 das rechte Pad.

IE1 bzw IE2 sind direkt mit den Klemmen verbunden (Eingänge END1 und END2).

Induktive Näherungsschalter liefern oft 24V wenn sie betätigt werden. Um einen solchen Schalter als Endschalter 1 zu betreiben, muss bei Pad-Gruppe „A1“ der linke und der mittlere Pad gebrückt werden. Ferner muss in der Pad-Gruppe „K1“ der mittlere mit dem rechten Pad verbunden werden. Die Anode der LED wird also (über den Widerstand) mit dem Schalter verbunden und die Kathode an Masse gelegt.

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>

Spannungsversorgung

Die SLBL 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 SLBL nimmt im Leerlauf weniger als 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 SLBL 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 ca. 5A 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, werden diese Transistoren sofort gesperrt. Dieser Vorgang wiederholt sich beim nächsten PWM-Zyklus. Das alles geschieht automatisch durch die Hardware der SLBL und ist nach außen hin nicht sichtbar, äußert sich jedoch mit einem merkwürdigen Kratzen des Motors.

Die mit 5V bezeichneten Klemmen 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 SLBL 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 auch 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 *cal 2*, *cal 3*, *cal 4* oder *cal 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

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>

Nullspur synchronisiert. Der Anschluss der Nullspur kann also unter den genannten Umständen entfallen.

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
Endschalter On/Off	sl/rl bzw. ssyscon/rsyscon
Endschalter invertieren	sil/ril bzw. ssyscon/rsyscon
Motorstrombegrenzung	scl/rcl
LC-Display-Typ	slcd/rlcd
Bürstenloser/Bürstenbehafteter Motor	ssyscon/rsyscon
Differenzieller/nicht-differenzieller Enkoder	ssyscon/rsyscon
AD-Wandler On/Off	ssyscon/rsyscon
Basic-Interpreter-Sart nach Power-On	ssyscon/rsyscon
Basic-Programm-Start nach Power-On	ssyscon/rsyscon
LC-Display On/Off	ssyscon/rsyscon
Rotationsgeber (Display-Modul) On/Off	ssyscon/rsyscon

Anmerkungen zum Positionierfehler-Fenster

Sobald der Positioniermodus(*pm*) eingeschaltet wird, wird die Motoristposition 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 bleibt das Flag *inpos* auf Null.

Liegt die Motoristposition 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 herum befunden haben wenn das Flag *inpos* gesetzt ist.

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>

Kommunikationsprotokoll

Die Kommunikation mit einem übergeordneten Rechner findet über die RS232-Schnittstelle statt. 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 Kommunikations-Protokoll 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 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 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 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, verbehalte 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 SLBL 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 SLBL 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 SLBL 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 ASCII 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.

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>

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). "rchar" wird ein ASCII 13 (CR) zurückliefern. Damit ist die Endebedingung 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 Endebedingung für "rstring" erreicht. An "sbefehl" wird der Antwortstring auf den Befehl *ma1234* zurückgeliefert der eine Länge von 0 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 SLBL an einer seriellen Schnittstelle betrieben werden:

Angenommen die SLBL 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 SLBL: Die drei Zeichen werden zusammen mit dem abschließenden ASCII 13 (CR) empfangen (und geechot). SLBL 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 SLBL 2 sofort ihren Sendekanal abschalten.

SLBL 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). 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 aus 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 SLBL keinerlei Bedeutung haben und beim Empfang bereits ausgefiltert werden. Es spielt also keine Rolle, ob Sie z.B. "sv 1000" oder "sv1000" senden.

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>

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

Parameter-Einstellhilfe

Voraussetzungen: SLBL Software PID v1.11
GrTermW Version 1.08

In die Software der SLBL wurde eine kleiner „Datenrecorder“ eingebaut der für 300ms nach Ende der Bewegung das Ergebnis der Subtraktion Istposition – Sollposition aufzeichnet.

Diese Daten können in GrTermW V1.05 eingelesen werden indem die Tastenkombination Strg-D

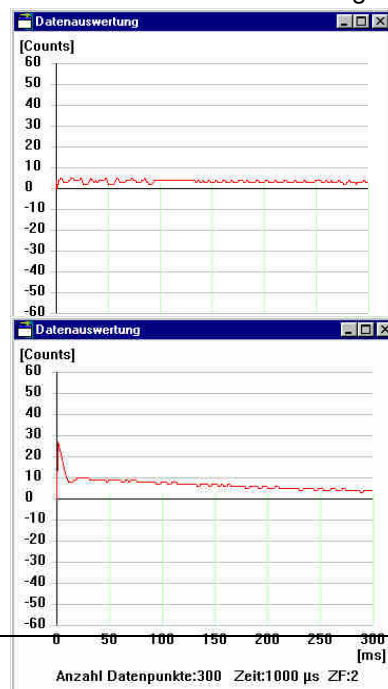
gedrückt wird. Nach Drücken von Strg-D erscheint ein Nachrichtenfenster mit dem Text „Wait..“. Nach einigen Sekunden werden die Daten in einem Fenster angezeigt.

Unter „Counts“ ist die Encoderposition zu verstehen.

Unter „Zeit:1000µs“ ist die Zeit zwischen zwei Datenpunkten zu verstehen. Mit „ZF:2“ wird der augenblickliche Zoom-Faktor angezeigt.

Die (roten) angezeigten Messdaten sagen eindeutig aus, dass der I-Anteil der Reglers etwas zu niedrig eingestellt ist. Der Motor hat eine Abweichung von etwa +4 Counts zum Zielwert. Die Tasten Page-Up und Page-Down ändern den Skalierungsfaktor der Y-Achse.

Vor der Aufnahme neuer Daten muss das Fenster unbedingt geschlossen werden, das geht auch mit der ESC-Taste !



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>

Bei diesem Bild ist zu sehen, dass der Motor nach Ende der Bewegung um etwa 28 Counts über die Sollposition hinaus fährt, dann aber innerhalb von ca. 15ms auf einen Positionierfehler von +10 Counts zurückkommt. Danach wird der (zu kleine) I-Anteil des Reglers wirksam der den Positionierfehler innerhalb des angezeigten Zeitraums auf etwa +4 Counts reduziert. Sinnvoll wäre hier zunächst den I-Anteil abzuschalten und P und D soweit zu erhöhen bis der Überschwinger möglichst klein wird. Danach wird I erhöht bis ein zufriedenstellendes Ergebnis erzielt wird.

Übrigens werden die Daten in dem File „data.txt“ zwischengespeichert. Diese Daten lassen sich mit Strg-Y nochmals anzeigen.

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 SLBL verbunden werden.

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

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