

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## Handbuch SLBZ

Handbuch Version V1.00

Hardware Version 2211

Software Version 1.00

Allgemeines.....	2
Anforderungen an den Motor.....	2
Inbetriebnahme.....	2
LED-Betriebsanzeige.....	3
Was Sie machen MÜSSEN.....	3
pm.....	5
vm.....	5
st.....	5
auto.....	5
sdz n.....	5
rdz.....	5
sau n.....	5
rau.....	5
sv n.....	5
rv.....	6
sa n.....	6
ra.....	6
rad, sadfs, smdz, so und sr.....	6
kp n / ki n / kd n.....	7
qp / qi / qd.....	7
sp n.....	7
rp.....	7
ssyscon n.....	7
rsyscon.....	7
rrsyscon.....	7
ss.....	8
Auswertung der Flags.....	8
rss.....	8
id.....	8
pg.....	8
scl n.....	9
rcl.....	9
sdd n.....	9
rdd.....	9
smo n.....	9
rmo.....	9
spwm.....	9
Lage der Anschlüsse.....	10
<b>ANHANG.....</b>	<b>11</b>
Spannungsversorgung.....	12
Winkelencoder.....	12
Welche Einstellungen werden dauerhaft gespeichert ?.....	12
Kommunikationsprotokoll.....	13
PID-Parameter.....	15
Verdrahtungshinweise.....	15
Technische Daten SLBZ.....	15

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## Allgemeines

### Merkmale SLBZ

- Betrieb von bürstenlosen und bürstenbehafteten DC-Motoren
- Eine Versorgungsspannung im Bereich von 12..30V=
- Digitaler PID-Regler
- Maximaler Motorstrom ca. 2A (interne Strombegrenzung)
- Eingänge für differenzielle und nicht-differenzielle Winkelencoder mit und ohne Null-Spur
- Analoger Eingang 0..5V zur Drehzahlvorgabe
- Optokoppelter Enable-Eingang
- Optokoppelter Drehrichtungseingang
- Open Collector – Ausgang zur Fehleranzeige
- Alle Anschlüsse über Käfigzugfederklemmen
- Eingebaut im Gehäuse: ca. 100\*90\*25mm<sup>3</sup>
- Optional mit zum Aufschnappen auf T35-Schienen

Die Baugruppe dient zur exakten Drehzahlregelung kleiner DC-Motoren bis zu einer Leistung von ca. 50W. 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:

Motorconfiguration

Encoderconfiguration

Parametrierung

Beschleunigung/Verzögerung

Abfrage der Ist-Position des Motors usw.

## Anforderungen an den Motor

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

Anschließbar sind bürstenlose und bürstenbehaftete Motoren mit inkrementalem Winkelencoder. 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.

Beherrigen 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 "InbetriebSLBZ0100.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 und die grüne LED leuchtet.

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

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 Encoder-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 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 zwischen dem Gehäuse und den Klemmen 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 2000mA eingestellt. Zur Einstellung und Abfrage der Motorstrombegrenzung dienen die Befehle "scl n" und "rcl".

## Was Sie machen MÜSSEN

Einstellung

Motortyp einstellen

Encodertyp einstellen

Enable-/Disable Konfigurieren

Encoderauflösung einstellen

Max. Motordrehzahl festlegen

Höhe der min. Eingangsspannung feststellen u. einstellen

Höhe der max. Eingangsspannung feststellen u. einstellen

Motorbeschleunigung festlegen

PID-Parameter ermitteln

Siehe Befehl

ssyscon Bit 0

ssyscon Bit 1

ssyscon Bit 2

sr

smdz

so

sadfs

sau oder sa

kp, ki, kd, siehe auch Anhang

Auslieferungszustand

Einstellung

Bürstenbehafteter Motor

Encodertyp single endet

Enabled wenn stromlos od. unbeschaltet

Encoderauflösung 512 Linien

Drehzahl 6000UpM

Max. Eingangsspannung 930 ADC-Counts (~5V)

Analoger Offset 20 ADC-Counts (~20mV)

Beschleunigung 2000 Counts

P=40 I=40 D=80

Setzen mit

ssyscon Bit 0=0

ssyscon Bit 1=0

ssyscon Bit 2=0

sr

smdz

sadfs

so

sa

kp, ki, kd

Auslesen mit

rsyscon od. rrsyscon

rsyscon od. rrsyscon

rsyscon od. rrsyscon

rr

rmdz

radfs

ro

ra

qp, qi, qd

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • 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
auto	AUTOmatic	Eine der Eingangsspannung am analogen Eingang proportionale Drehzahl wird automatisch eingestellt. Das ist auch der Modus nach Power on.
sau <i>n</i>	Set Accel. in upm/m	Setzen der Beschleunigung in upm/m
rau	Read Accel. In upm/m	Auslesen der gesetzten Beschleunigung in upm/m
sdz <i>n</i>	Set DrehZahl in upm	Setzen der Soll Drehzahl in upm
rdz	Read DrehZahl	Die intern gemessene Drehzahl wird ausgegeben in upm
sv <i>n</i>	Set Velocity	Drehzahl setzen in Counts
rv	Read Velocity	Eingestellte Drehzahl auslesen in Counts
sa <i>n</i>	Set Acceleration	Beschleunigung einstellen in Counts
ra	Read Acceleration	Eingestellte Beschleunigung auslesen in Counts
rad	Read AD-Conv.	Die am Analog Inp. Anliegende Spannung wird gewandelt und in Counts und mV angezeigt.
sadfs <i>n</i>	Set AD Full Scale	Damit wird festgelegt, welcher AD-Wandler-Wert der max. Drehzahl entsprechen soll
radfs	Read AD Full Scale	Auslesen des eingestellten Wertes
smdz	Set MAX DrehZahl	Hiermit wird die Motordrehzahl festgelegt die erreicht werden soll, wenn der AD-Wandler den Wert von „sadfs“ liest.
rmdz	Read MAX DrehZahl	Auslesen des eingestellten Wertes.
So <i>n</i>	Set analog Offset	Einstellen eines am Analog Inp. Liegenden Offsets (in Counts)
ro	Read analog Offset	Auslesen des eingestellten Wertes
Sr <i>n</i>	Set Resolution	Auflösung des Winkelencoders in Linien einstellen
rr	Read Resolution	Auslesen des eingestellten Wertes
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
rp	Read Position	Auslesen der Motorposition
ssyscon <i>n</i>	Set Configuration	Hard-/Software konfigurieren
rsyscon	Read Configuration	Auslesen der Konfiguration
rrsyscon	Read Configuration	Auslesen der Konfiguration in Klartext
ss	StatuS	Status auslesen
rss	Read StatuS	Status in Klartext auslesen
rin	Read Inputs	Eingänge digital oder analog abfragen
rad	AD-converter	Auslesen des AD-Wandlerkanals vom Analog Inp. Angaben in AD-Wandler-Counts und in mV
ld	IDentification	Versionsmeldung abfragen
pg	ProGram	Einstellungen im EEPROM speichern
scl <i>n</i>	Set Current Limit	Motorstrombegrenzung ( <i>n</i> =mA)
rcl	Read Current Limit	Einstellung der Motorstrombegrenzung auslesen
sdd <i>n</i>	Set Display Delay	Verzögerung bis LED erlischt
rdd	Read Display Delay	Einstellung auslesen
smo <i>n</i>	Set delay Motor Off	Verzögerung bis Motor abschaltet
rmo	Read delay Motor Off	Einstellung auslesen
spwm <i>n</i>	Set PWM	Unregelmäßigen Motorstrom erzeugen ( <i>n</i> =-255..+255)

# **gräbner-elektronik gmbh**

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## **pm**

Position Mode

Damit wird der Regler eingeschaltet und wird 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 (siehe „sdz“ oder „sv“) 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.

## **auto**

AUTOMatic

Der Regler wird eingeschaltet, der Motor wird entsprechend der am analogen Eingang liegenden Spannung drehen. Dieser Modus wird automatisch nach Power On eingestellt. Der Befehl dient zu Testzwecken nach „st“.

## **sdz n**

Set DrehZahl in upm

Die mit „sdz“ übergebene Drehzahl wird abhängig vom Encoder umgerechnet (in Counts) und eingestellt.

## **rdz**

Read DrehZahl

Intern wird die Drehzahl des Motors laufend berechnet, völlig unabhängig vom eingestellten Modus. Der Wert aus dieser Berechnung wird mit „rdz“ geliefert.

Beachten Sie, dass dieser Wert aus der momentanen tatsächlichen Drehzahl des Motors abgeleitet wird.

Der Befehl „rdz“ kann also nur bedingt zur Verifikation des mittels „sdz“ eingestellten Wertes dienen.

## **sau n**

Set Acceleration in Upm/m

Einstellen der Beschleunigung in Umdrehungen pro Minute in der Minute.

## **rau**

Read Acceleration in Upm/m

Der mittels „sau“ oder „sa“ eingestellte Wert wird (ggf.) umgerechnet und in UpM/M ausgegeben.

## **sv n**

Set Velocity

Setzen der (maximalen) Verfahrensgeschwindigkeit/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

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

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

# **gräbner-elektronik gmbh**

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

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

## **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.

## **rad, sadfs, smdz, so und sr**

Diese vier Befehle sollen gemeinsam behandelt werden, da sie in einem engen inneren Zusammenhang stehen.

Damit überhaupt Angaben in UpM erfolgen können, muss der Baugruppe bekannt sein, welche Auflösung der Winkelencoder hat. Übergeben Sie die Auflösung in Linien mittels des Befehls „sr“ an die Baugruppe. Beispiel:

*sr 512*

Stellt einen Winkelencoder mit 512 Linien ein.

Nun legen Sie zunächst fest, welche maximale Drehzahl Sie verlangen, wenn am Analog Inp. die maximale Spannung anliegt. Beispiel:

*smdz 400*

Legt fest, dass die Drehzahl 400 UpM sein wird wenn die maximale Spannung am Analog Inp. anliegt.

Nun stellen Sie den analogen Offset fest. Das geschieht, indem Sie die minimalste Spannung an den Analog Inp. anlegen und mehrmals mittels „rad“ den Wandler auslesen. Bilden Sie aus den ermittelten Werten den Mittelwert und übergeben Sie diesen Wert mittels „so“. Dabei interessieren nicht die Werte in „mV“, sondern die linken, dimensionslosen Werte in den Ausgaben von „rad“. Beispiel:

*so 30*

legt fest, dass der Offset am Analog Inp. (Eingang des AD-Wandlers) 30 Counts ist.

Nun legen Sie fest, bei welcher Spannung am Analog Inp. die höchste Drehzahl erreicht werden soll.

Legen Sie dazu diese hohe Spannung (Achtung: max. 5V!) an den Analog Inp. und lesen Sie den AD-Wandler mittels „rad“ mehrfach aus. Bilden Sie wieder den Mittelwert aus den dimensionslosen Zahlenwerten links in der Ausgabe. Stellen Sie nun diesen Wert mittels „sadfs“ ein. Beispiel:

*Sadfs 500*

stellt knapp 2,5V als max. Eingangsspannung ein.

Intern wurde durch die beschriebenen Massnahmen festgelegt:

Eingangsspannungsbereich zw. 30 und 500 Counts

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

Drehzahlbereich zw. 0 und 400UpM

Daraus ergibt sich ein wirksamer AD-Wandler-Bereich 470 Counts und eine Einstellgenauigkeit von  $400\text{Upm}/470 = 0,851\text{ UpM}$  pro AD-Wandler-Count.

Benutzen Sie als Drehzahl-Sollwert-Vorgabe einen einfachen Potentiometer müssen Sie davon ausgehen, dass diese oftmals über einen recht erheblichen Restwiderstand verfügen wenn der Schleifer am Anschlag steht. Ermitteln Sie daher diesen Offset unbedingt und geben Sie ein paar Counts dazu, damit ein Drehzahl von 0UpM sicher eingestellt werden kann.

## **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  $\pm 2^{25}$ .

## **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	bl	Motor: High=bürstenlos Low=bürstenbehaftet
1	2	enc	Encoder: High=differentiell (RS422) Low=Single ended
2	4	ce	Low=Baugruppe enabeld wenn der Enable-Eingang stromlos oder unbeschaltet ist. High=Baugruppe enabled, wenn der Enable-Eingang bestromt wird.
6	8	hex	High=Alle Zahlenausgaben in hexadezimaler Schreibweise

## **rsyscon**

Read SYStem CONfiguration

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

## **rrsyscon**

Read SYStem CONfiguration

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

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

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## 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	vmode	High=Drehzahlregelmodus eingeschaltet
Bit 1	2	pmode	High=Positioniermodus eingeschaltet
Bit 2	4	move	Low=Motor steht / High=Motor dreht
Bit 3	8	auto-vm	High=Drehzahlregelmodus. Sollwert über Analog Inp.
Bit 4	16	überstrom	High= Überstrom wurde detektiert
Bit 5	32	stillstand	High=Motorstillstand wurde detektiert
Bit 6	64	motor off	High=Motorabschaltung wegen Überstrom od. Stillstand

### vmode

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

### pmode

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

### move

Das Flag wird High nachdem „vm“ oder „auto“ empfangen wurde oder direkt nach Power On. Das Flag wird nach „st“ Low.

### auto-vm

Das Flag ist unmittelbar nach Power On gesetzt und wird erst nach „st“ low.

### überstrom

Das Flag wird sofort gesetzt, wenn ein Überstrom erkannt wurde. Der max. zulässige Motorstrom wird mit „scl“ eingestellt.

### stillstand

Dieses Flag wird High wenn der interne Schleppfehler (Sollposition des Rampengenerators–Istposition des Motors) das Maß überschreitet das mit „sse“ eingestellt wurde.

### motor off

Das Flag wird high wenn Motorüberstrom oder Motorstillstand erkannt wurden und die Zeiten „sdd“ und „smo“ abgelaufen 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&8)==0 printf ("Positioniermodus nicht eingestellt");  
else printf ("Positioniermodus eingeschaltet");
```

die entsprechende Meldung erzeugt werden.

## rss

Read StatuS

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

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

## 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.



# **gräbner-elektronik gmbh**

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## **scl n**

Set Current Limit

Der mit „scl“ zu übergebende Wert *n* darf zwischen 0 und 2000 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 zwischen Gehäuse und Klemmen signalisiert durch ihr Verlöschen oder Flackern, dass die Strombegrenzung wirksam ist.

## **rcl**

Read Current Limit

Zum Überprüfen des mit „scl“ eingestellten Wertes.

## **sdd n**

Set Display Delay

Tritt ein interner Fehler auf (Überstrom oder Motorstillstand) wird die mit „sdd“ eingestellte Zeit vergehen, bis die LED abgeschaltet wird und so den Fehler anzeigt.

Wird während des Ablaufs der Zeit der Fehler behoben, erfolgt kein Abschalten der LED.

Die Fehlerursache kann aus den Bits von „ss“ bzw. „rss“ ermittelt werden.

Hinweise zur Fehlerursache liefert auch die LED: Blinkt sie, ist die Fehlerursache ein Schleppfehler bzw. ein Motorstillstand. Ist sie dauerhaft abgeschaltet war die Fehlerursache ein Motorüberstrom.

## **rdd**

Read Display Delay

Zum Überprüfen des mit „sdd“ eingestellten Wertes.

## **smo n**

Set delay Motor Off

Tritt ein interner Fehler auf (Überstrom oder Motorstillstand) und ist die mit „sdd“ eingestellte Zeit abgelaufen, wird die mit „smo“ eingestellte Zeit vergehen, bis der Motor abgeschaltet wird.

Wird während des Ablaufs der Zeit der Fehler behoben, erfolgt kein Abschalten des Motors und die LED wird wieder eingeschaltet.

Der Motor bleibt abgeschaltet bis über die ser. Schnittstelle der Befehl „auto“ gegeben wird oder die Versorgungsspannung aus- und wieder eingeschaltet wurde (Power On Reset).

Die Fehlerursache kann aus den Bits von „ss“ bzw. „rss“ ermittelt werden.

Hinweise zur Fehlerursache liefert auch die LED: Blinkt sie, ist die Fehlerursache ein Schleppfehler bzw. ein Motorstillstand. Ist sie dauerhaft abgeschaltet war die Fehlerursache ein Motorüberstrom.

## **rmo**

Read delay Motor Off

Zum Überprüfen des mit „smo“ 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 Winkelencoders 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.

# gräbner-elektronik gmbh

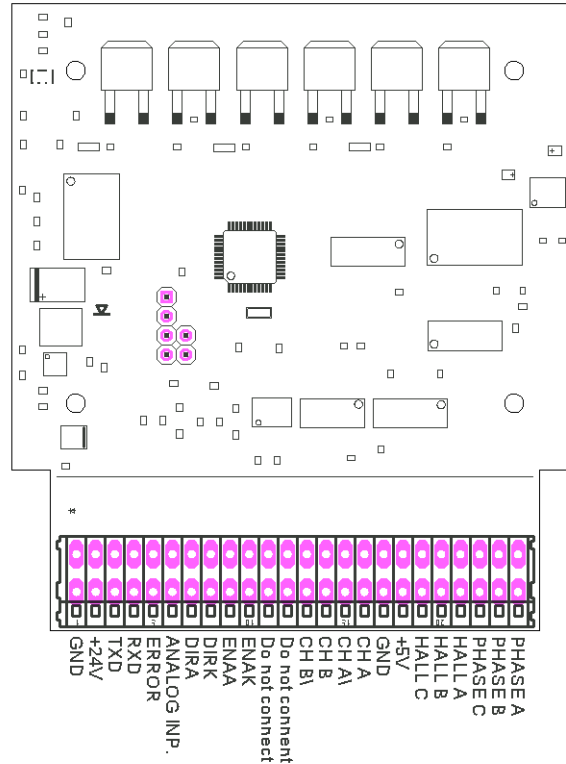
Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## Lage der Anschlüsse

Bild: Draufsicht



## Signale am Klemmenblock

Klemme	Signal	
1	GND	Spannungsversorgung
2	+24V	Zulässig sind +12..30V zur Spannungsversorgung
3	TxD	RS232-Sendekanal der SLBZ
4	RxD	RS232 Empfangskanal der SLBZ
5	Error	Error Fehlerausgang
6	Analog Inp	Analoger Solldrehzahleingang. Spannungsbereich 0..5V
7	DIRA	Direction – Drehrichtung Anode
8	DIRK	Direction – Drehrichtung Kathode
9	ENAA	Enable-Eingang Anode
10	ENAK	Enable-Eingang Kathode
11		Unbeschaltet lassen
12		Unbeschaltet lassen
13	CH B\	Encoder Kanal B invertiert (Anschluss nur bei differenziellem Encoder )
14	CH B	Encoder Kanal B
15	CH A\	Encoder Kanal A invertiert (Anschluss nur bei differenziellem Encoder )
16	CH A	Encoder Kanal A
17	GND	GND Hallensoren/Winkelencoder/RS232
18	+5V	für Hallensoren/Winkelencoder <b>Ausgang!</b>
19	HALL C	Hallsensor C (Anschluss nur für bürstenlose Motoren)
20	HALL B	Hallsensor B (Anschluss nur für bürstenlose Motoren)
21	HALL A	Hallsensor A (Anschluss nur für bürstenlose Motoren)
22	PH C	Motor-Phase C
23	PH B	Motor-Phase B (Anschluss nur für bürstenlose Motoren)
24	PH A	Motor-Phase A

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## Bürstenbehaftete Motoren

werden an Motor-Phase A (Klemme 24) und Motor-Phase C (Klemme 22) angeschlossen. Klemme 23 (Motor-Phase B) bleibt bei bürstenbehafteten Motoren ohne Beschaltung.

## Die Hallsensoranschlüsse

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

## Nicht-differenzielle Winkelencoder

werden an die Klemmen 16 und 14 angeschlossen.

## Differenzielle Winkelencoder

werden an die Klemmen 16/15 und 14/13 angeschlossen.

## ENAA und ENAK

führen auf Optokoppler. Klemme 9 (Enable-Eingang Anode) führt über einen Widerstand von 2,2kOhm auf die Anode der LED im Optokoppler. Klemme 10 (Enable-Eingang Kathode) führt auf die Kathode der LED des Optokopplers. Je nach vorliegenden Verhältnissen sind diese Klemmen entsprechend zu beschalten.

## DIRA und DIRK

führen auf Optokoppler. Klemme 7 (Drehrichtungseingang Anode) führt über einen Widerstand von 2,2kOhm auf die Anode der LED im Optokoppler. Klemme 8 (Drehrichtungseingang Kathode) führt auf die Kathode der LED des Optokopplers. Je nach vorliegenden Verhältnissen sind diese Klemmen entsprechend zu beschalten.

## Analog Inp.

An Klemme 6 wird die Spannung zur Sollwertvorgabe eingespeist. Nur positive Spannungen sind erlaubt. Der höchste zulässige Spannungspegel an diesen Klemmen ist +10V.

## Error

Der Pegel an Klemme 5 zeigt an, ob eine Störung vorliegt. Liegt eine Störung vor, schaltet der interne Transistor durch und es sind nahezu 0V zu messen (Max. Strom: 20mA).

Liegt keine Störung vor, ist der interne Transistor gesperrt und die Klemme wird über einen 10kOhm Widerstand auf Betriebsspannungspegel gezogen.

## Die Spannungsversorgung

für die Baugruppe wird nur über die Klemmen 1(GND) und 2(+24V) 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.

## Klemmen 17 und 18 (GND und +5V)

Die mit +5V bezeichnete Klemme stellt einen Ausgang dar, der zur Versorgung des Winkelencoders und der Hallsensoren dienen soll. Wird an diese Klemme eine externe Spannung angelegt, wird die Baugruppe sofort zerstört und ist nicht mehr reparierbar!

## ANHANG

Allgemeines, Zusammenhänge und Hintergründe

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

# **gräbner-elektronik gmbh**

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## **Spannungsversorgung**

Die SLBZ benötigt nur eine Spannungsversorgung im Bereich 12..30V.

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 SLBZ 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 SLBZ 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, mittels „scl n“ eingestellte Wert 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 SLBZ.

## **Winkelencoder**

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

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

Nichtdifferenzielle Encoder 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 die Signalpegel dann angepasst werden.

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

## **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:

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

Bezeichnung	Befehl zum Setzen/Auslesen
PID-Parameter P	kp/qp
PID-Parameter I	ki/qi
PID-Parameter D	kd/qd
Motorstrombegrenzung	scl/rcl
Bürstenloser/Bürstenbehafteter Motor	ssyscon/rsyscon
Differenzieller/nicht-differenzieller Encoder	ssyscon/rsyscon
Max. Drehzahl	smdz/rmdz
Analoger Offset	so/ro
Analoger Full Scale Wert	sadfs/radfs
Winkelencoderauflösung	sr/rr
Beschleunigung	sau/rau
Anzeige-Delay	sdd/rdd
Abschalt-Delay	smo/rmo

## **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, 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 SLBZ 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.

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

Die fünfte Routine bekommt den Namen "sbefehl". An sie wird ein beliebiger Befehlsstring übergeben und liefert die von der SLBZ 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 SLBZ 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.

## 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 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 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.

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.

In diesem Zusammenhang noch eine wichtige Anmerkung: Die Baugruppe 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 SLBZ keinerlei Bedeutung haben und beim Empfang bereits ausgefiltert werden. Es spielt also keine Rolle, ob Sie z.B.

# gräbner-elektronik gmbh

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

“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

Damit sollte sich jeder Motor zumindest drehen.

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.

## **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 SLBZ verbunden werden.

Alle anderen Anschlüsse wie Winkelencoder, 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.

## **Technische Daten SLBZ**

Hardware Version 2211 / Software Version 1.00

### Datenübertragungsrate

19200Bd, 8Bit, 1 Stoppbit, Keine Parität

### Pegel auf den Datenübertragungsleitungen

Es werden Pegel nach RS232 (ca. +12V) erwartet bzw. generiert

### Spannungsversorgung

12..30V Gleichspannung

Das Gerät ist aus technischen Gründen nicht verpolsicher

# **gräbner-elektronik gmbh**

Am Römerbrunnen 11a • 61118 Bad Vilbel

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

eMail: [info@graebner-elektronik.eu](mailto:info@graebner-elektronik.eu) • Internet <http://www.graebner-elektronik.eu>

## Stromaufnahme ohne Motor

<100mA bei 24V Versorgungsspannung

## Maximal zulässiger Motordauerstrom

2000mA

## Strombegrenzung

Auslieferungszustand: 2000mA

## Eingänge Winkelenkoder

TTL-Pegel, Innenwiderstand ca. 3000 Ohm oder

differenziel nach RS422, Innenwiderstand ca. 360Ohm

Es werden zwei(vier), um 90° phasenverschobene Rechtecke erwartet (Quadraturenkoder)

## Maximal zulässige Ausgangsfrequenz des Winkelenkoders

0,5MHz

## Flankensteilheit der Enkodersignale

Kleiner 500ns

## Maximale Eingangsspannung an Analog Inp.

+10V

## Umgebungstemperaturbereich

0..60°C

## CE-Kennzeichnung

Das Gerät verfügt nicht über eine CE-Kennzeichnung da es eigenständig nicht zu betreiben ist