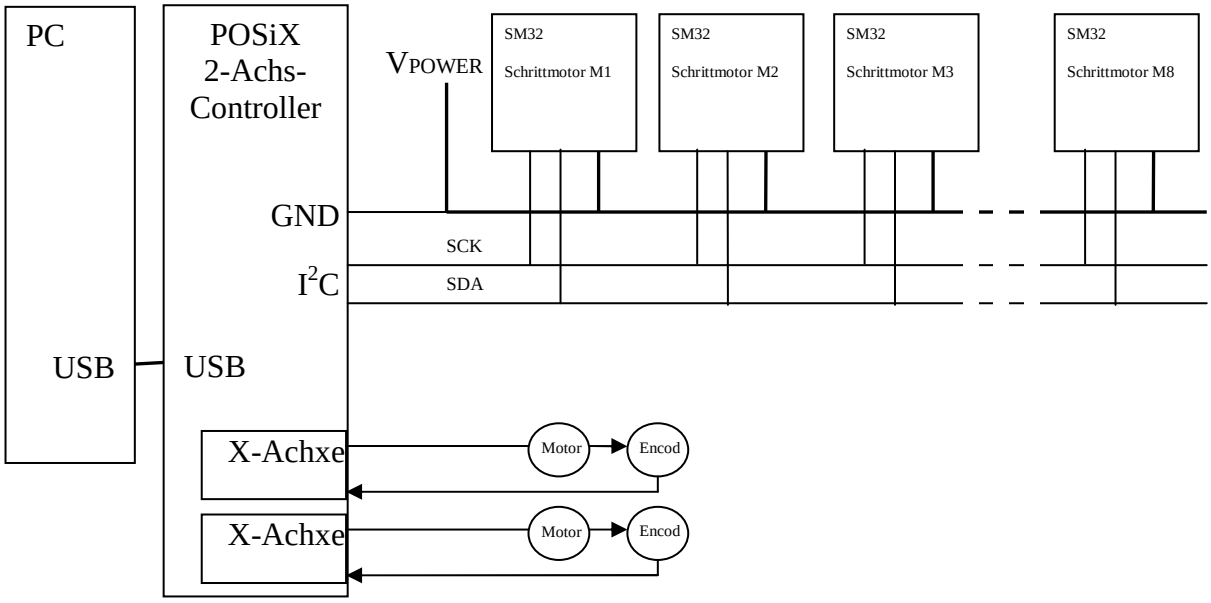
Das POSiX 2 Achsen Positioniersystem

Für anspruchsvolle Positionieraufgaben dienen DC-Antriebe. Bei POSiX handelt es sich um ein vollständiges 2 Achsen Positioniersystem für Bürstenmotoren mit einer Stromaufnahme von bis zu 2.8 Ampere pro Antrieb. Positionierung der 4 Quadranten und unterlagerter Drehzahlregelkreis mit PID Algorytmus erfolgt über einen Encoder. Da der Steuerungsprint zusätzlich über einen I2C Busanschluss verfügt, ist eine beliebige Kombination des DC-Posititioniersystems mit den Schrittmotoren möglich.



Technische Daten:

Grösse der Leiterplatte:	90 mm x 50 mm
Befestigunsbohrungen:	4, d=2.5mm
Schnittstelle:	USB, virtueller Com Port *mit TxD, GND, RxD, b=38400 bd , 1 Stop, kein Parity*
Antriebe:	*Je Achse ein DC-Motor mit 35V/2.8A*
Versorgungsspannung: Stromaufnahme (ohne Peripherie)	7VDC … 35VDC typ. 0.21A bei 7VDC typ. 0.15A bei 10VDC typ. 0.13A bei 12VDC typ. 0.10A bei 18VDC

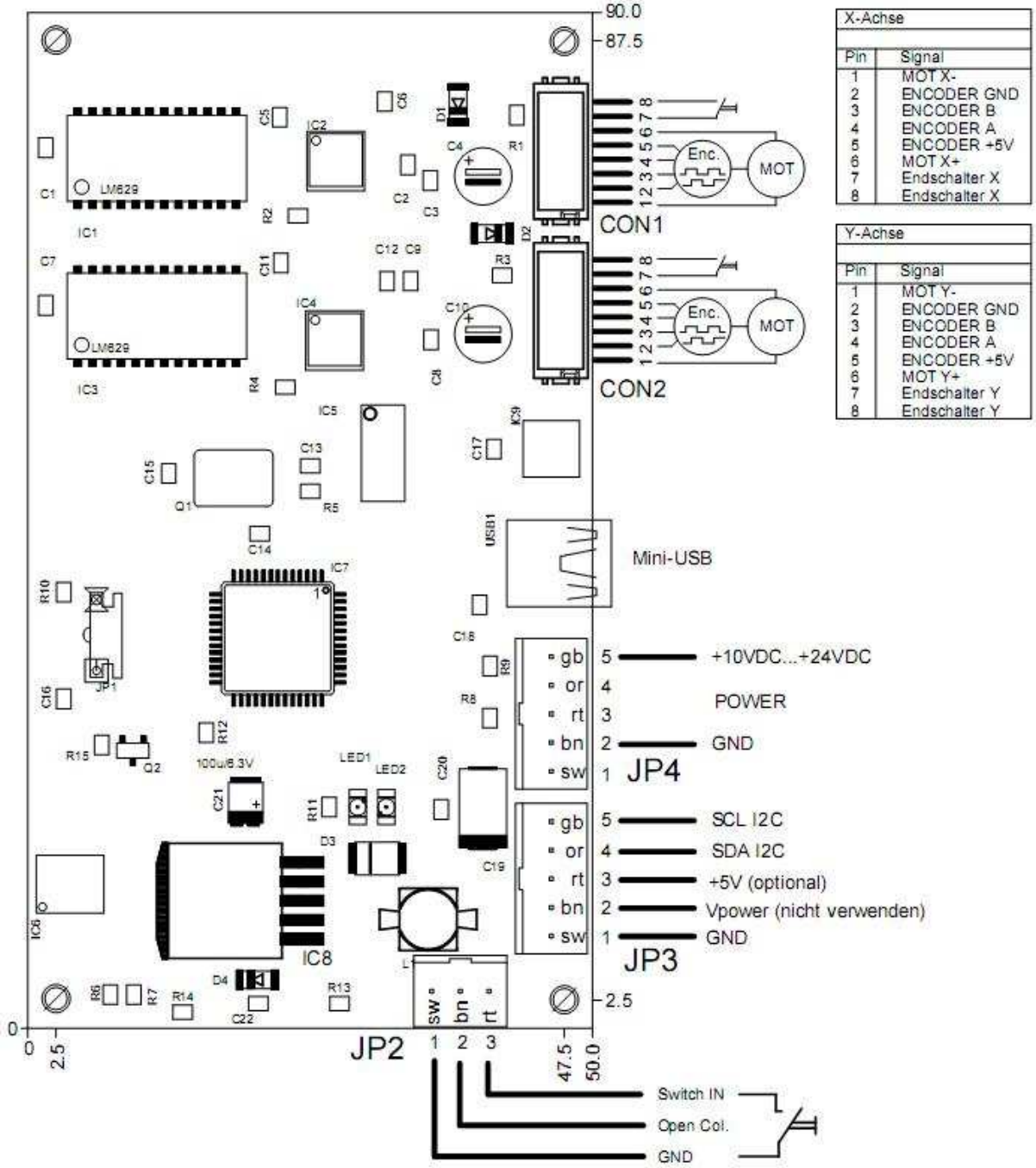
Speisespannung Antriebe: *10V ... 35VDC*

Eingänge für Achsüberwachung: *Je Achse ein Kontakt* bzw. Initialisierungsschalter

Initialisierung auch über Schleppfehlermessung kontaktfrei möglich (Antrieb geregelt in einen Anschlag fahren).

Ausgänge programmierbar:	*Ein Open-Collectort, maximale Schaltspannung 40VDC bei 0,1ADC.*
Eingänge programmierbar:	*Ein Schliess-Kontakt, potentialfrei*
Programmbetrieb:	*Befehlsliste siehe unten*
Max. Verfahrgeschwindigkeit:	*VMAX = 1000000 Inkremente / Sekunde = 1MHz (Messsystem-Signal)*
Parameterspeicher:	nichtflüchtig, für alle Achsen
Versorgung für Encoder:	+5VDC, 200mA pro Achse
Anschluss X-Achse:	8 poliger Micro-Match Anschluss
Anschluss Y-Achse:	8 poliger Micro-Match Anschluss
Anschluss I^2C-Bus:	5 polige Stiftleiste für SM32
X- und Y- Achse:	unterlagerte Drehzahlregelkreise mit PID
Programmspeicher:	10, nichtflüchtig, PROG0 …. PROG9 à 490 ASCII Zeichen.

Anschluss POSiX 2 Achsen

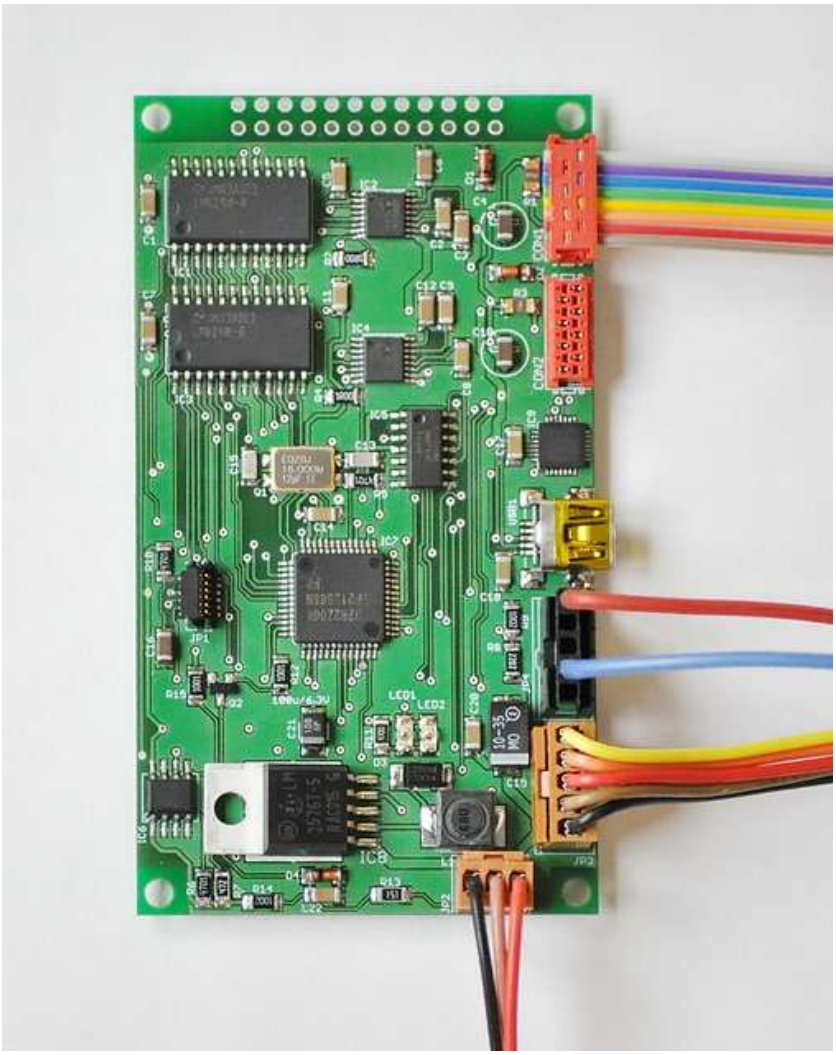


X-Achse	
Pin	Signal
1	MOT X-
2	ENCODER GND
3	ENCODER B
4	ENCODER A
5	ENCODER +5V
6	MOT X+
7	Endschalter X
8	Endschalter X

Y-Achse	
Pin	Signal
1	MOT Y-
2	ENCODER GND
3	ENCODER B
4	ENCODER A
5	ENCODER +5V
6	MOT Y+
7	Endschalter Y
8	Endschalter Y

Lieferumfang:

1	Stück	Leiterplatte POSiX 2 Achsen
2	Stück	Steckverbinder Mikromatch 8-polig mit vorkonfektioniertem Flachbandkabel für CON1 und CON2
1	Stück	Mini-USB Kabel
1	Stück	Power Anschlusskabel für JP4
1	Stück	I^2C-Anschlusskabel für JP3
1	Stück	Peripherie-Anschlusskabel für JP2
1	Stück	Gebrauchsanweisung
1	Stück	CD mit USB Treibern für Silabs CP210x virtual Comport



Hinweise zur ersten Inbetriebnahme

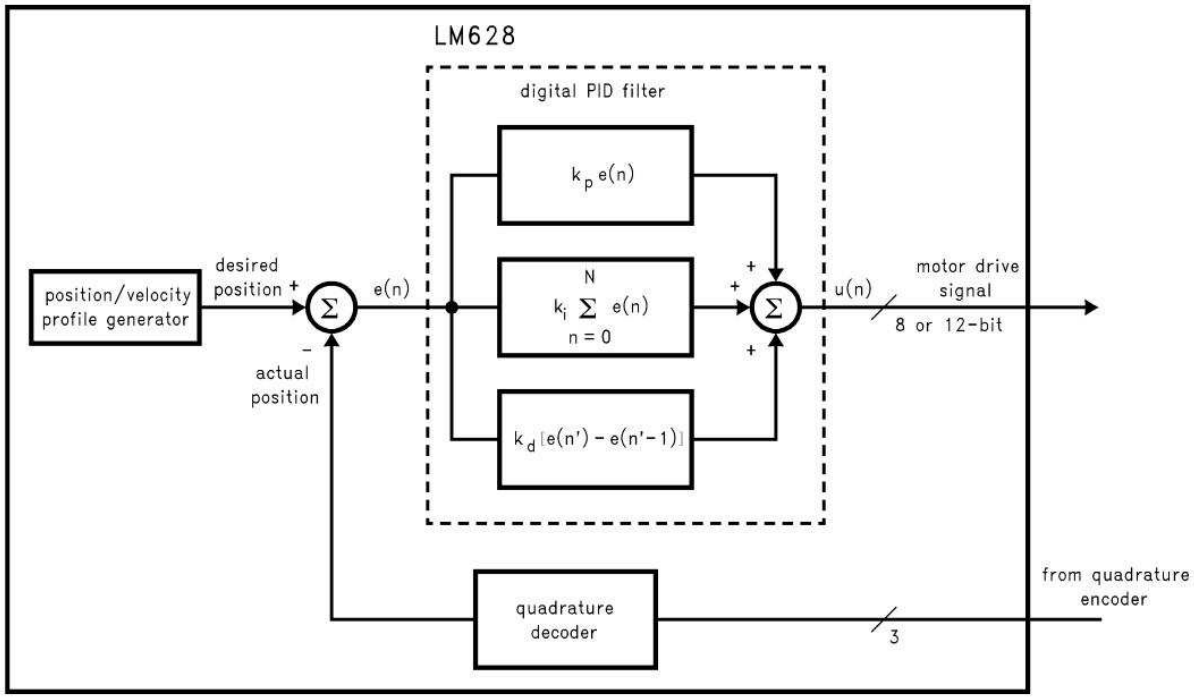
Die Versorgungsspannung für die Motoren MUSS über JP4 geführt werden. Wenn zusätzlich SM32 Schrittmotor-Controller angeschlossen werden, so müssen diese zwingend die Motorenspannung über JP3 erhalten. Keinesfalls die SM32 direkt aus einem Netzteil versorgen.

- Es wird empfohlen, zunächst eine DC-Motoren-Achse zu verdrahten.
- Wenn eine DC-Motoren-Achse angeschlossen ist, dann sollte die Versorgungsspannung angeschlossen werden. Mindestens eine LED auf der Leiterplatte muss blinken.
- Wenn der DC-Motor scheinbar unkontrolliert losläuft, dann ist der Richtungssinn von MOT + und MOT – Anschluss nicht in Übereinstimmung mit den Encoder-Signalen. Sofort ausschalten. Es muss dann entweder der Motor umgepolt werden oder aber die A- und B- Signale des Encoder vertauscht werden.
- Wenn der DC-Motor korrekt regelt, kann die Versorgungsspannung abgeschaltet werden und es können die anderen Verbindungen z.B. zu einem zweiten DC-Motor und USB erfolgen.
- Hyperterminal Programm starten und Verbindung herstellen.
- Die Eingabe der <ENTER> Taste wird von POSiX mit ‚SYNTAXERROR;' quittiert. Wenn das funktioniert, dann kann mit der Steuerung gemäss Befehlssatz kommuniziert werden.

POSiX Parameter

POSiX verwendet für den korrekten Betrieb der Kraft- und Drehzahlregelung sogenannte PID-Werte. Dabei steht P für den Proportionalteil, D für den Differentialteil und I für den Integralteil sowie IL (Integral limit). Weiterhin müssen Werte für Beschleunigung, Geschwindigkeit und Zielposition definiert werden.

Mit Vorsicht sind die PID Parameter und IL zu behandeln. Diese folgen mathematischen Beziehungen. Bei falschen Kombinationen von Werten kann es zu Überschwingen oder Maschinenproblemen kommen. Da die Berechnung sehr komplex ist, wird empfohlen, die Standardwerte zu verwenden und an die jeweilige Maschine durch herantasten anzupassen – dieses Verfahren wird auch vom Hersteller des LM629 (motion controll IC) empfohlen.



Block Diagramm des LM628/LM629. In POSiX wird der LM629 verwendet, der einen PWM (pulse width modulation) Signal an die Motorenendstufe abgibt.

Befehlssatz

POSiX verwendet für die Steuerung von Schrittmotoren feste I^2C Adressen. Diese sind hier dokumentiert. Ausserdem wird ein komplexer Klartext Befehlssatz verwendet, der ausschliesslich ASCII Zeichen verwendet. Damit kann ein so betriebenes System auch mit einfachen Terminalprogrammen auskommen, um den I^2C Bus anzusprechen.

Grundsätzlich gilt:

- POSiX verwendet einen Befehlsbuffer in der Grösse von 500 Zeichen.
- Als Befehlstrennzeichen (delimiter) werden die Zeichen <;> oder char[13] = 0x0d verwendet. In dieser Beschreibung wird als Delimiter ausschliesslich das <;>-Zeichen verwendet.
- ein unbekannter Befehl wird mit 'SYNTAXERROR' quittiert.
- Alle Befehle unterscheiden Gross- und Kleinschreibung.
- Alle Quittierungen bzw. Ausgaben enden mit dem <;>-Zeichen.
- Wird in der unten stehenden Befehlsbeschreibung als Antwort "z.B. " aufgeführt, so ist die Antwort abhängig vom Parameter "confirm on".
- Der virtuelle COM-Port muss folgendermassen eingestellt sein:
  - 38400 baud
  - 8 Datenbits
  - 1 Stopbit
  - kein Hardwarehandshake, kein Xon/Xoff
  - Verwendet wird der USB Treiber von Silabs für die Familie der CP210x USB-UART-Bridges.
- Wenn nicht explizit anders erwähnt, dann handelt es sich bei den I2C Busteilnehmern um TMC222 Schrittmotorcontroller bzw. SM32 Baugruppen von TB-ELECTRONICS. Zu Fragen und Betrieb der TMC222 muss die technische Dokumentation von Trinamic verwendet werden.
- Geänderte Parameter werden in POSiX permanent solange gespeichert, bis sie überschrieben werden.

Befehle in alfabetischer Reihenfolge

Systembefehle

Befehl	Bedeutung
buf;	Zeigt den aktuellen Stand des Empfangs-buffers. Antwort: BUFFER=15;
confirm on;	Jeder eingegebene Befehl wird mit der Antwort quittiert (default). Antwort: CONFIRM=ON;
confirm off;	Die Quittierung von Befehlen wird unterdrückt. Antwort: CONFIRM=OFF;
echo on;	Jedes eingegebene Zeichen wird quittiert. Antwort: ECHO=ON;
echo off;	Ein eingegebens Zeichen wird nicht quittiert (default). Antwort: ECHO=FF;
new;	Rücksetzen des Eingabebuffers auf 0. Antwort: BUFFER=0;
nl;	Neue Zeile (new line). <cr> <lf>
delay vvvvv;	Verzögert den Befehlsablauf. vvvvv=0 ... 65535. v=1 entspricht 1 Millisekunde. Antwort: delay...ready; „ready“ wird nach Ablauf der Delayzeit angezeigt.

whois;

Zeigt alle Schrittmotor-Busteilnehmer an und initialisiert die Schrittmotoren mit Default Werten. Es wird für alle Motoren s=3 eingestellt. Das entspricht der kleinsten Schrittauflösung der TMC222.

Ein korrekt angeschlossener Motor wird mit seiner Adresse angezeigt. Ansonsten erfolgt.

Antwort z.B. wenn nur MOT3 und MOT7 installiert sind:

MOT=3=0xCC:
MOT=7=0xEC;

version;

Zeigt die aktuell installierte Firmware

Antwort:
VERSION=V1.00B01;SN=11001;STARTONSWITCH=0;

Dabei bedeuten V1.00B01 die installierte Firmware-Version,
SN=11001 die Seriennummer,
STARTONSWITCH=0 der Parameter für den Programmstart ist ausgeschaltet.

set default;

Setzen der Default-Parameter für angeschlossenen Schrittmotoren. Diese sind:
IRUN=10
IHOLD=2
VMAX=10
VMIN=2
SHAFT=0
ACCEL=2

xydefault;

Setzen der Default Parameter der X- und Y Achsen. Diese sind:

XAchse:	XAchse:
XKP=100	YKP=100
XKI=100	YKI=100
XKD=100	YDP=100
XIL=1000	YIL=100
XACCEL=100	YACCEL=100
XPRECI=10	YPRECI=10
XVELO=1000	YVELO=1000

Vorschlag Parameter, wenn 24VDC Motoren verwendet werden und ein Encoder mit 500 Inkrementen pro Umdrehung an der Motorenachse verbunden ist:

XAchse:	XAchse:
XKP=1000	YKP=100
XKI=100	YKI=100
XKD=10000	YDP=100
XIL=10000	YIL=100
XACCEL=300	YACCEL=100
XPRECI=10	YPRECI=10
XVELO=1000	YVELO=1000

Programm-Speicherbefehle

Das System verfügt über 10 nichtflüchtige Programmspeicherplätze. Die Nummern sind 0 … 9. Programmspeicherplätze können gelöscht, über die Schnittstelle ausgegeben werden und erzeugt werden. Der Start eines Programmes erfolgt über einen Befehl oder über den Eingang, wenn der Parameter STARTONSWITCH auf 1 gesetzt wird.

Bei der Erzeugung eines Programmes erfolgt keine Syntaxüberprüfung. Diese erfolgt erst beim Ablauf. Es wird empfohlen, ein Programm in einem Texteditor zu entwerfen und dann ins System zu übertragen.

Bei der Syntax ist zu beachten, dass die Trennzeichen zwischen den einzelnen Befehlen hier das , ,' – Zeichen (Komma) ist.

clearprogv;	Löschen eines der 10 Programmspeicher. v=0 bis v=9 Beispiel: clearprog9; löscht Programmspeicher 9. Es darf kein Leerzeichen oder ein anderes Zeichen vor der Ziffer enthalten sein. Antwort: CLEARING PROG 9...FINISHED;
printprogv;	Ausgabe eines der 10 Programmspeicher. v=0 bis v=9 Beispiel: printprog0; gibt Programmspeicher 0 aus. Es darf kein Leerzeichen oder ein anderes Zeichen vor der Ziffer enthalten sein. Antwort: PROG 0: mot0, mabs10000, xvelo1000, yvelo1000, xabs10000, yabs10000, xdesti5000, ydesti, xylin, mabs,

createprogv; Erzeugung eines der 10 Programmspeicher. v=0 bis v=9
Beispiel: createprog0……; speichert solange Befehle ab, bis der Separator ‚;' eingegeben wird. . Es darf
kein Leerzeichen oder ein anderes Steuerzeichen
enthalten sein.

Beispiel:
createprog0mot0,mabs10000,xvelo1000,yvelo1000,xabs10000,yabs10000,xdesti5000,ydesti,xylin,mabs,;

Eingabe in einer Zeile, ohne Zeilenvorschub etc.

goprogv; Start eines der 10 Programmspeicher. v=0 bis v=9
Beispiel: goprog0; startet Programmspeicher 0. Es darf
kein Leerzeichen oder ein anderes Zeichen vor der Ziffer
enthalten sein.
Der Ablauf kann über die Schnittstelle beobachtet werden, wenn AUTOCONFIRM = ON ist,

outv; Setzt den Open Collector Ausgang. v=0 oder 1

switch; Einlesen Eingang SWITCH IN an JP3.

Antwort;: SWITCH=0; = Schalter offen
SWITCH=1; = Schalter geschlossen

startonswitchv; Startet Programm0 automatisch, wenn dieser Parameter auf '1' gesetzt ist UND der Schalter geschlossen wird. Bevor das Programm startet, muss der Kontakt des Schalters wieder geöffnet werden.

autostatusv; Setzt die automatische Ausgabe von X- und Y-Position, wenn v=1. autostatus0; schaltet die automatische Ausgabe aus.

POSiX Achscontroller – Befehle X-Achse

xparas; Gibt die aktuelle eingestellten Parameter der X-Achse aus.

Antwort z.B.

```
XPARAS:XKP=100;XKI=100;XKD=1000;XIL=100;XACCEL=500;XPRECI=20;XVEL
O=10;XDEST=0;XREAL=XABS=+0;
```

Es bedeuten:

`XKP=100;`	P-Anteil des PID-Reglers.
`XKI=100;`	I-Anteil des PID-Reglers.
`XKD=1000;`	D-Anteil des PID-Reglers.
`XIL=100;`	IL-Wert des PID-Reglers (integration limit)
`XACCEL=500;`	Beschleunigung der Achse, kleiner Wert = geringe Beschleunigung, grosser Wert = hohe Beschleunigung.
`XPRECI=20;`	Genauigkeit, mit der der Regler für ABS-Befehle ‚IN POSITION' meldet, damit der folgende Befehl abgearbeitet werden kann.
`XVELO=10;`	Geschwindigkeit der Achse. Kleiner Wert = geringe Geschwindigkeit, grosser Wert = hohe Geschwindigkeit.
`XDEST=0;`	Zuletzt übergebene Zielposition.
`XREAL=XABS=+0;`	Aktuelle Position der Achse.

xoff;	Schaltet die X-Ache aus. Die Achse ist jetzt ungeregelt und kann frei bewegt werden. Man kann die Positionen abfragen. Dieser Modus ist z.B. sinnvoll, um einen Teach-In Betrieb zu ermöglichen oder die Achse manuell in eine bestimmte Position zu bewegen.
xon;	Schaltet die X-Ache ein nachdem der Befehl xoff; übermittelt wurde. Die Achse wird wieder geregelt und es wird die Position übernommen, die im Positionszähler enthalten ist.
xreset;	Die aktuelle Position der Achse wird auf ‚0' gesetzt.
xiniti vvvvvvvv;	Initialisierung der Achse auf Schleppfehler bis zum Anschlag. Mit vvvvvvvv wird der maximale Verfahrweg angegeben, innerhalb der die Initialisierung erfolgen muss. Wird innerhalb des vorgegebenen Weges der mechanische Anschlag angefahren, baut sich ein interner Schleppfehler als Integrationssumme auf und das System meldet XINIT…READY. Wird kein Anschlag angefahren, so meldet das System XINITI…FAILED; In beiden Fällen stoppt der Antrieb und die Position ist auf 0 gesetzt. Der Wertebereich ist -9999999 <= vvvvvvvv <= +9999999.
xinits vvvvvvvv;	Initialisierung der Achse auf Endschalter. Mit vvvvvvvv wird der maximale Verfahrweg angegeben, innerhalb der die Initialisierung erfolgen muss. Wird innerhalb des vorgegebenen Weges der Endschalter angefahren, meldet das System XINIT…READY. Wird kein Anschlag angefahren, so meldet das System XINITI…FAILED; In beiden Fällen stoppt der Antrieb und die Position ist auf 0 gesetzt. Der Wertebereich ist -9999999 <= vvvvvvvv <= +9999999.
xkp vvvvv;	0 <= v <= 10000. Eingabe P-Anteil für die Achse. Kleiner Wert = geringe Verstärkung, grosser Wert = hohe Verstärkung. Default xkp = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
xki vvvvv;	0 <= v <= 10000. Eingabe I-Anteil für die Achse. Kleiner Wert = geringe Integration (Aufschaukeln), grosser Wert = hohe Integration (Vibration). Default xki = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
xkd vvvvv;	0 <= v <= 10000. Eingabe D-Anteil für die Achse. Kleiner Wert = geringer Differentialteil (startet schwer), grosser Wert = hoher Differentialteil (Überkompensation, ggf. Schwingen). Default Ixkd = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !

xil vvvvv;	0 <= v <= 10000. Eingabe Integartionsgrenze für die Achse. Default xil = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
xaccel vvvvv;	0 <= v <= 10000. Eingabe Beschleunigung. Default xaccel = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
xpreci vvvvv;	0 <= v <= 10000. Genauigkeit, mit der der Regler für ABS-Befehle ‚IN POSITION' meldet, damit der folgende Befehl abgearbeitet werden kann. Eingabe in Inkrementen. Default I = 20. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
xvelo vvvvv;	0 <= v <= 10000. Geschwindigkeit, mit der die Achse fährt. Default I = 10.
xgo vvvvvvv;	-9999999 <= v <= +9999999. Position in Inkrementen, auf die die Achse fährt. Hier handelt es sich um einen Positionierbefehl, der ausgeführt wird und der nächste (falls vohanden) im Batch befindliche Befehl sofort zur Ausführung kommt.
xabs vvvvvvv;	-9999999 <= v <= +9999999. Position in Inkrementen, auf die die Achse fährt. Hier handelt es sich um einen Positionierbefehl, der ausgeführt wird und die Achse muss sich erst innerhalb der Grenze [xpreci] befinden, bevor der nächste (falls vohanden) im Batch befindliche Befehl sofort zur Ausführung kommt.

POSiX Achscontroller – Befehle Y-Achse

Yparas; Gibt die aktuelle eingestellten Parameter der Y-Achse aus.

Antwort z.B.

```
YPARAS:YKP=100;YKI=100;YKD=1000;YIL=100;YACCEL=500;YPRECI=20;YVEL
O=10;YDEST=0;YREAL=YABS=+0;
```

Es bedeuten:

`YKP=100;`	P-Anteil des PID-Reglers.
`YKI=100;`	I-Anteil des PID-Reglers.
`YKD=1000;`	D-Anteil des PID-Reglers.
`YIL=100;`	IL-Wert des PID-Reglers (integration limit)
`YACCEL=500;`	Beschleunigung der Achse, kleiner Wert = geringe Beschleunigung, grosser Wert = hohe Beschleunigung.
`YPRECI=20;`	Genauigkeit, mit der der Regler für ABS-Befehle ‚IN POSITION' meldet, damit der folgende Befehl abgearbeitet werden kann.
`YVELO=10;`	Geschwindigkeit der Achse. Kleiner Wert = geringe Geschwindigkeit, grosser Wert = hohe Geschwindigkeit.
`YDEST=0;`	Zuletzt übergebene Zielposition.
`YREAL=YABS=+0;`	Aktuelle Position der Achse.

yoff;	Schaltet die Y-Ache aus. Die Achse ist jetzt ungeregelt und kann frei bewegt werden. Man kann die Positionen abfragen. Dieser Modus ist z.B. sinnvoll, um einen Teach-In Betrieb zu ermöglichen oder die Achse manuell in eine bestimmte Position zu bewegen.
yon;	Schaltet die Y-Ache ein nachdem der Befehl yoff; übermittelt wurde. Die Achse wird wieder geregelt und es wird die Position übernommen, die im Positionszähler enthalten ist.
yreset;	Die aktuelle Position der Achse wird auf ‚0' gesetzt.
yiniti vvvvvvvv;	Initialisierung der Achse auf Schleppfehler bis zum Anschlag. Mit vvvvvvvv wird der maximale Verfahrweg angegeben, innerhalb der die Initialisierung erfolgen muss. Wird innerhalb des vorgegebenen Weges der mechanische Anschlag angefahren, baut sich ein interner Schleppfehler als Integrationssumme auf und das System meldet YINIT…READY. Wird kein Anschlag angefahren, so meldet das System YINITI…FAILED; In beiden Fällen stoppt der Antrieb und die Position ist auf 0 gesetzt. Der Wertebereich ist -9999999 <= vvvvvvvv <= +9999999.
yinits vvvvvvvv;	Initialisierung der Achse auf Endschalter. Mit vvvvvvvv wird der maximale Verfahrweg angegeben, innerhalb der die Initialisierung erfolgen muss. Wird innerhalb des vorgegebenen Weges der Endschalter angefahren, meldet das System YINIT…READY. Wird kein Anschlag angefahren, so meldet das System YINITI…FAILED; In beiden Fällen stoppt der Antrieb und die Position ist auf 0 gesetzt. Der Wertebereich ist -9999999 <= vvvvvvvv <= +9999999.
ykp vvvvv;	0 <= v <= 10000. Eingabe P-Anteil für die Achse. Kleiner Wert = geringe Verstärkung, grosser Wert = hohe Verstärkung. Default ykp = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
yki vvvvv;	0 <= v <= 10000. Eingabe I-Anteil für die Achse. Kleiner Wert = geringe Integration (Aufschaukeln), grosser Wert = hohe Integration (Vibration). Default yki = 100, Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !
ykd vvvvv;	0 <= v <= 10000. Eingabe D-Anteil für die Achse. Kleiner Wert = geringer Differentialteil (startet schwer), grosser Wert = hoher Differentialteil (Überkompensation, ggf. Schwingen). Default Iykd = 100. Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !

yil vvvvv; 0 <= v <= 10000. Eingabe Integartionsgrenze für die Achse.
Default yil = 100.
Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !

yaccel vvvvv; 0 <= v <= 10000. Eingabe Beschleunigung.
Default yaccel = 100.
Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !

ypreci vvvvv; 0 <= v <= 10000. Genauigkeit, mit der der Regler für ABS-Befehle ‚IN POSITION' meldet, damit der folgende Befehl abgearbeitet werden kann. Eingabe in Inkrementen.
Default I = 20.
Achtung: nach Ausführung dieses Befehls ist die aktuelle Position ‚0' !

yvelo vvvvv; 0 <= v <= 10000. Geschwindigkeit, mit der die Achse fährt. Default I = 10.

ygo vvvvvvv; -9999999 <= v <= +9999999. Position in Inkrementen, auf die die Achse fährt. Hier handelt es sich um einen Positionierbefehl, der ausgeführt wird und der nächste (falls vohanden) im Batch befindliche Befehl sofort zur Ausführung kommt.

yabs vvvvvvv; -9999999 <= v <= +9999999. Position in Inkrementen, auf die die Achse fährt. Hier handelt es sich um einen Positionierbefehl, der ausgeführt wird und die Achse muss sich erst innerhalb der Grenze [ypreci] befinden, bevor der nächste (falls vohanden) im Batch befindliche Befehl sofort zur Ausführung kommt.

xyshow; Abfrage der absoluten Position in Inkrementen. Antwort z.B.: XABS=+0; oder XABS=-12345; oder +20000;

Befehle für die lineare Interpolation:

Die lineare Interpolation koordiniert die Achsen derart, dass sie gemeinsam an der Zielposition ankommen. So sind geregelte „Schrägbewegungen“ möglich.

xdesti vvvvvvv; -9999999 <= v <= +9999999. Zielposition X in Inkrementen, auf die die Achse von der momentanen Position fahren soll.

Beispiel: Eingabe xdesti10000;
Antwort: XDESTI=10000;

ydesti vvvvvvv; -9999999 <= v <= +9999999. Zielposition Y in Inkrementen, auf die die Achse von der momentanen Position fahren soll.

Beispiel: Eingabe ydesti-10000;
Antwort: YDESTI=-10000;

xylin; Ausführen der Bewegung mit linearer Interpolation auf XDESTI und YDESTI. Es sind keine Befehle oder Positionsabfragen möglich.

Antwort: BUSY; solange die Achsen sich bewegen
Sobald die Achsen innerhalb der durch PRECI definierten Position sind, erfolgt die Anzeige der aktuellen Positionen.

Befehle für Schrittmotoren

Motorsteuerbefehle - nicht in alfabetischer, sondern in funktionell logischer Reihenfolge.

Befehl	Bedeutung	Antwort
mot v;	Auswahl des anzusteuernden Schrittmotors. Motorennummer v=0...13. Wenn v=0, dann wird Adresse 0xC0 als I^2C device address verwendet. Das entspricht einem nicht programmierten TMC222. Das Leerzeichen zwischen 'MOT' und dem Wert muss nicht vorhanden sein. Alle auf den MOT-Befehl folgende Befehle beziehen sich auf die Motorennummer. Beispiel: mot7; Antwort: MOT=7=0xEC;	z.B. 'MOT=
maccel vv;	Setzen der Beschleunigung des aktuell angewählten Motors. vv=1...15 Kleine Werte = kleine Beschleunigung, grosse Werte = grosse Beschleunigung. Default: vv=2. Beispiel: maccel3; Antwort: Accel=3;	
mihold vv;	Setzen des Haltestroms des aktuell angewählten Motors. vv=0...15 Kleine Werte = kleiner Haltestrom, grosse Werte = grosser Haltestrom. Default: vv=2. Beispiel: mihold1; Antwort: Ihold=1;	

mirun vv;	Setzen des Fahrstroms des aktuell angewählten Motors. vv=0...15 Kleine Werte = kleiner Fahrstrom, grosse Werte = grosser Fahrstrom. Default: vv=10. Beispiel: mirun13; Antwort: Irun=13;
mvmin vv;	Setzen der Minimalgeschwindigkeit des aktuell angewählten Motors. vv=0...15 Kleine Werte = kleine Geschwindigkeit, grosse Werte = grosse Geschwindigkeit. Default: vv=2. Beispiel: mvmin1; Antwort: Vmin=1;
mvmax vv;	Setzen der Maximalgeschwindigkeit des aktuell angewählten Motors. vv=0...15 Kleine Werte = kleine Geschwindigkeit, grosse Werte = grosse Geschwindigkeit. Default: vv=10. Beispiel: mvmax12; Antwort: Vmax=12;
mshaft v;	Setzen der Richtung für positive Positionswerte. v=0...1. Default: v=0. Beispiel: mshaft1; Antwort: Shaft=1;

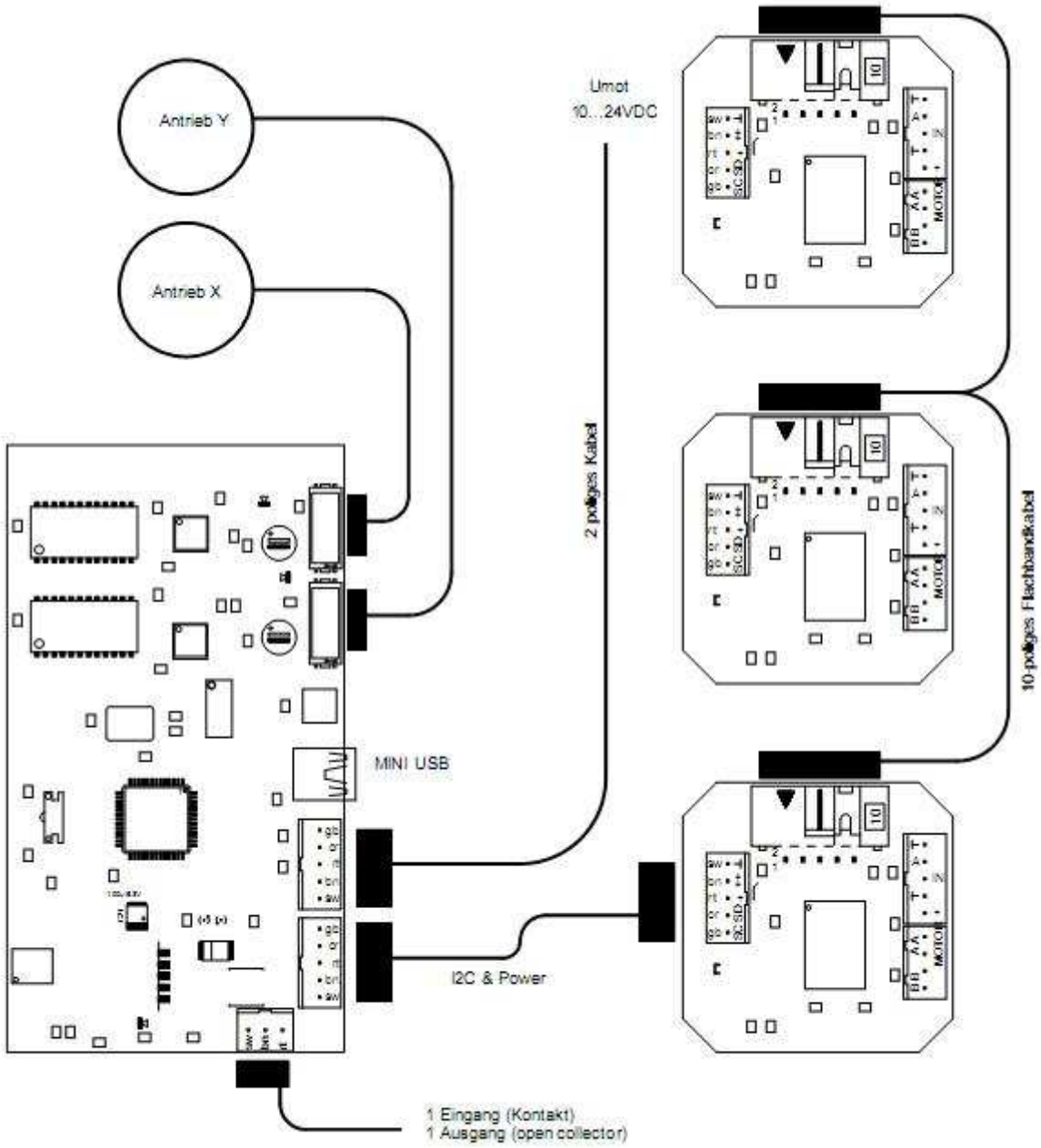
mgo vvvvv;	Setzen der Zielposition des aktuell angewählten Schrittmotors und Start der Fahrbewegung. -32000 < vvvvv < +32000 Inkremente. Beispiel: mgo12000; fährt den Motor auf Position 12000 Schritte.
mpos;	Abfrage der Position des aktuell angewählten Motors. Beispiel: mpos12000; Antwort: MOT7=+12000;
msearch v;	Suche der Referenzposition des aktuell angewählten Motors. v=0...1. v=0 fährt solange bis der angeschlossene Endschalter schliesst. v=1 fährt solange bis der angeschlossene Endschalter öffnet. Wenn die Zeitüberwachung abgelaufen ist, wird der Vorgang beendet.
mstart v;	Betrieb des aktuell angewählten Motors im Dauerlauf - keine Positionierung. v=0...2. v=0 stoppt den Dauerlauf. v=1 startet in Standard-Drehrichtung. v=2 startet in Umkehr-Drehrichtung. Beispiel: mstart1; Antwort: Start=1; Motor dreht solange, bis der Befehl mstart0; ausgelöst wird.

Beispiel für einen Befehlsablauf:

new;mot3;mspeed3;mgo10000;mot^{2};mgo3000;delay1500;mgo0;mot3;mgo0;

- rücksetzen des Befehlsbuffers
- Motor 3 adressieren
- Geschwindigkeit MOT3 auf 3 setzen
- Motor 3 auf 10000 positionieren
- Motor 2 adressieren
- Motor 2 auf -3000 positionieren
- 1500 Millisekunden warten
- Motor 2 auf 0 positionieren
- Motor 3 adressieren
- Motor 3 auf 0 positionieren

Verbindungs-Beispiel



Besipiel Anschluss POSiX 2 Achsen & 3 Schrittmotoren

Obiges Beispiel zeigt, wie ein System bestehend aus DC-Servos und Schrittmotoren aufgebaut sein kann.

TB-ELECTRONICS GmbH

Bahnhofstrasse 3
CH-9443 Widnau

Tel.: 071 722’52’55
Fax: 071 722’52’05