

TMC200



Fraunhofer Institut
Autonome Intelligente
Systeme

Triple Motorcontroller für DC-Motoren



March 15, 2004

Bedienungs- und Einstellanleitung

1 Allgemeine Beschreibung

In Europakartengröße ist der *TMC200* in vielfältigen Anwendungen einsetzbar. Die flexiblen Anbindungsmöglichkeiten ermöglichen ein breites Einsatzgebiet in allen Bereichen, von der mobilen Robotik bis hin zu industriellen Maschinen.

Der *TMC200* ist für bis zu drei Gleichstrommotoren mit einer Dauerleistung von bis zu 200 Watt konzipiert. In Verbindung mit Standard-Impulsgebern lassen sich Drehzahlregelungen mit hohen Anforderungen bewältigen. Der Motor Controller basiert auf einem leistungsstarken 16-Bit Mikrocontroller der eine hohe Regelgüte ermöglicht.

1-1 Fakten

Mit diesem Motor Controller können folgende Antriebsaufgaben gelöst werden:

- ⇒ Drehzahlregelung mit hohen Anforderungen an Gleichlauf und geringen Drehmoment-Schwankungen. Der PID-Regler sorgt für Erhaltung der Drehzahl.
- ⇒ Betrieb als Drehmomentenregler durch einstellbare Strombegrenzung
- ⇒ Thermischer Motorschutz durch I²t- Strombegrenzung
- ⇒ Positions-Berechnung mit Hilfe der Odometrie
- ⇒ 10 Bit frei nutzbarer I/O Port für eigene Sens- oder Aktuatorik
- ⇒ Betriebsspannungsüberwachung

1-2 Anbindung

Zur Umsetzung dieser Aufgaben stehen zwei verschiedenen Kommunikations-Schnittstellen zur Verfügung.

- ⇒ CAN-Bus (opt. Ausgekoppelt)
- ⇒ RS232

Sämtliche Parameter, wie Reglerkonstanten, Motorgrößen und Arbeitsmodi können über die jeweilige Kommunikations-Schnittstellen eingestellt werden.

2 Anschluss der Hardware



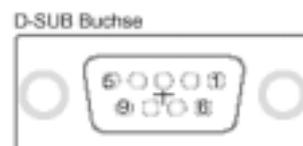
2-1 Anschlussbelegung

Versorgung:

Nr.	Bezeichnung
V1	+24V
V2	GND

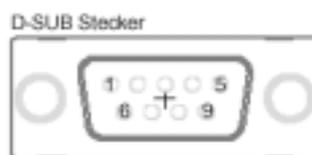
RS-232 SUB-D Buchse:

Nr.	Bezeichnung
2	RS-232 RXD
3	RS-232 TXD
5	GND



CAN SUB-D Stecker:

Nr.	Bezeichnung
-----	-------------



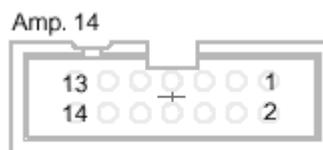
2	CAN-Low
3	CAN-GND
6	CAN-GND
7	CAN-High
9	CAN-Vcc

Motoren-Anschluß:

Nr.	Bezeichnung
M1	Motor1 +
M2	Motor1 -
M3	Motor2 +
M4	Motor2 -
M5	Motor3 +
M6	Motor3 -

Encoder- Stecker (Amp. 14pol):

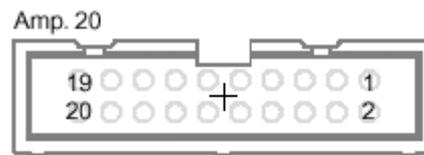
Nr.	Bezeichnung
-----	-------------



1	Motor1 Kanal B
2	VCC (+5V)
3	Motor1 Kanal A
4	Signal GND
5	Motor2 Kanal B
6	VCC (+5V)
7	Motor2 Kanal A
8	Signal GND
9	Motor3 Kanal B
10	VCC (+5V)
11	Motor3 Kanal A
12	Signal GND
14	VCC (+5V)

I/O Stecker (Amp. 20pol):

Nr. Bezeichnung



1	Port1.6
3	Port1.7
4	VCC (+5V)
5	Port1.8
7	Port1.9
9	Port1.10
11	Port1.11
13	Port1.12
15	Port1.13
17	Port1.14
19	Port1.15
20	Signal GND

3 Technische Daten

3-1 Allgemeine Spezifikationen

Beschreibung	Bezeichner	Bereich	Einheit
Versorgungsspannung	U_B	12 ... 30	V DC
PWM - Schaltfrequenz	f_{PWM}	50	kHz
Max. Dauer-Ausgangsstrom	I_{dauer}	8	A
mit zusätzlichem externen Kühlkörper	I_{dauer}	10	A
Max. Spitzen-Ausgangsstrom	I_{max}	20	A
Stromaufnahme der Elektronik	I_{el}	0,05	A
Ausgangsspannung für externen Gebrauch	VCC	5	V DC
Max. Laststrom	I_{CC}	0,5	A
I/O Pin als Eingang	Low Pegel	0 ... 0,5	V
	High Pegel	4,5 ... 5V	V
I/O Pin als Ausgang	Low Pegel	0 ... 0,5	V
	High Pegel	4,5 ... 5V	V
Max. Ausgangsstrom	$I_{I/Omax}$	0,02	A
RS 232 Max. Datendurchsatz		57,6	kBaud
CAN Max. Datendurchsatz		1	MBaud
Betriebstemperaturbereich		0 ... 70	°C
Lagertemperaturbereich		-20 ... 80	°C
Gewicht		300	g

4 Anschlüsse

4-1 Stromversorgungsanschlüsse (+24 V, GND)

Das angeschlossene Netzgerät/Akkumulator sollte mindestens die Summe der Nennströme der angeschlossenen Motoren als Dauerlast liefern.

Polaritätsanschluss beachten.

5 allgemeine Bemerkung

Die Dokumentation und Beschreibung der CAN- Bus Schnittstelle ist in dieser Version nicht vorhanden.

6 RS-232 Schnittstelle und ASCII-Befehle

6-1 Einstellungen der RS-232 Schnittstelle

Das RS-232 Interface ist mit den nachstehenden Einstellungen vorkonfiguriert.

Hinweis: nur durch gleiche Einstellung des Personal Computers etc. kann eine korrekte Funktion sichergestellt werden.

Einstellungen:

⇒ 57600 Baud(default)

⇒ 8 Datenbits

⇒ 1 Stoppbit

⇒ No Parity

6-2 Aufbau der ASCII-Befehle

Die Kommunikation mit dem PC geschieht üblicherweise über ein einfaches ASCII – Terminalprogramm, wie es im Windows Betriebssystem standardmäßig vorhanden ist. (Hyperterminal)

Die ASCII-Befehle sind nach folgendem Schema aufgebaut:

Befehlswort:

ein oder zwei Worte, eine Zeichenfolge aus Buchstaben getrennt durch Leerzeichen

Leerzeichen

Zahlenwert:

bei manchen Befehlen folgt nach dem Befehlswort eine oder mehrere Zahlen die durch Leerzeichen getrennt sind

Den Abschluss bildet immer ein „Cartridge Return“ - Zeichen, oder ein „Line Feed“ Zeichen.

Beispiele:

„SV 0 1024 0\n“ Setzen der Soll Drehzahl des Motors2 auf 100%.

„SMODE 2\n“ Setzen des Controller Modus auf PID Geschwindigkeitsregelung mit Strombegrenzung .

6-3 Grenzen der Kommunikation

Die minimale Zykluszeit des Controllers ist auf 10ms festgelegt, in dieser Zeit, kann der Controller maximal einen Befehl entgegennehmen, bearbeiten und anschließend antworten. Dies gilt für alle Befehle.

Der Befehl wird erst ausgeführt wenn er vollständig übertragen wurde. (Caridge Return- oder Line Feed Zeichen)

Wird ein Befehl nicht erkannt, so antwortet der Controller mit „ERR \n“.

Sind die Werte oder der Wert eines Befehls außerhalb des Definierten-Wertebereichs so werden sie auf das Maxi oder Minimum reduziert.

Befehle werden als „Setter“ oder „Getter“ kategorisiert, die meisten Befehle sind Tupel.

Auf einen „Getter“ antwortet der Controller immer mit den aktuellen Controller Größen.

Ein „Setter“ wird mit einem „OK \n“ beantwortet, wenn er eindeutig identifiziert wurde, die Ausnahme bilden die Zyklischen Befehle wie „SV 0 0 0\n“ ihre Antworten können frei konfiguriert werden.

6-4 Auslesen der Softwareversion

Der Revisions-String wird nach einem Reset oder beim Einschalten der TMC's gesendet, alternativ kann er auch wie folgt ausgelesen werden.

Befehl	Funktion	Beschreibung	Beispiel
GVERS	Get Version String	liest die aktuelle SW Versionsnummer aus	„VERS \n“ -> „TMC200 SW: \$Revision: 1.8 \$ \n“

6-5 Einstellen der Arbeitsmodi

Der Controller kann in drei verschiedenen Modi betrieben werden:

- ⇒ Direct Output, reine Steuerung (MODE 0)
- ⇒ PID Geschwindigkeitsregelung (MODE 1)
- ⇒ PID Geschwindigkeitsregelung mit Strombegrenzung (MODE 2) (default)

Befehl	Funktion	Beschreibung	Beispiel
SMODE <i>x</i>	Set Mode <i>x</i>	<p><i>x</i> = 0 Setzen des Betriebsmodus des Controllers auf Direct Output, reine Steuerung</p> <p><i>x</i> >= 1 Setzen des Betriebsmodus des Controllers auf PID Geschwindigkeitsregelung</p> <p><i>x</i> = 2 Setzen des Betriebsmodus des Controllers auf Geschwindigkeitsregelung mit Strombegrenzung</p>	„SMODE 0\n“
GMODE	Get Mode	Aktuellen Betriebsmodus abfragen.	„GMODE \n“ -> „MODE 0\n“

6-6 Der Geschwindigkeits- Befehl

Der wichtigste Befehl ist das Setzen der Sollgeschwindigkeiten für alle drei Motoren.

Befehl	Funktion	Beschreibung	Beispiel
SV Wert1 Wert2 Wert3	Set Velocity Wert1 Wert2 Wert3	Setzen der Sollgeschwindigkeit für alle drei Motoren	„SV -512 1024 0“

Wert1 = Sollgeschwindigkeit Motor1
(Bereich ± 1024)

Wert2 = Sollgeschwindigkeit Motor2
(Bereich ± 1024)

Wert3 = Sollgeschwindigkeit Motor3
(Bereich ± 1024)

z.B.: Der Befehl „SV -512 1024 0“ veranlasst im Arbeitsmodus 0 und 1, das Motor1 mit halber negativer Batteriespannung (-50%), Motor2 mit voller positiver (+100%) und Motor3 mit null Batteriespannung (0%) versorgt wird.

Im Arbeitsmodus 2, dagegen wird mit Hilfe des PID Reglers der Motor1 auf halber negativer Drehzahl (-50%) geregelt Motor2 auf volle positive (+100%) und Motor3 wird auf null Drehzahl (0%) gehalten.

Die prozentuale Angabe der Drehzahl bezieht sich immer auf die gesetzte Leerlauf-Drehzahl (siehe Punkt „*Einstellen der angeschlossenen Impulsgeber und Getriebekonstanten*“)

6-7 Konfiguration der Rückmeldung

Die Rückmeldung auf einen Geschwindigkeits-Befehl mit aktuellen Daten kann wie folgt konfiguriert werden.

Befehl	Funktion	Beschreibung	Beispiel
SSMODE Wert1	Set Send Mode Wert1	Konfigurieren der Rückmeldung auf ein Geschwindigkeits- Kommando Wert1 = 0 keine Antwort Wert1 >= 1 aktuelle Geschwindigkeiten (Bereich ± 1024) Wert1 >= 2 +aktuelle Ströme (Bereich ± 1024) Wert1 >= 3 +Nachrichten-Zähler, aktuelle Zykluszeit (Bereich 0 bis 255, 10[ms]) Wert1 >= 4 +Distance (Bereich ± 23768) Wert1 >= 5 +aktueller PWM-Output (Bereich ± 1024)	„SSEND 5\n“ „SV -10 -20 0\n“ “ ->„V -5 -8 0 \t C 39 60 0 \t x 255 \t dT 10 \t D -2216 39 0 \t O -10 -20 0 \t \n“

Die Rückmeldung stellt eine Anreihung durch Tabulatoren getrennter Vektoren dar, abgeschlossen mit einem „Line Feed“ Zeichen.

Die Vektoren sind immer nach dem selben Prinzip aufgebaut:

Kennung Wert1 Wert2 Wert3 \t	Kennung : „V“ für Velocity	Wert0 = allgemein
oder	„C“ für Current	Wert1 = act. Motor1 Größe
	„D“ für Distance	
Kennung Wert0 \t	„O“ für Output (PWM)	Wert2 = act. Motor2 Größe
	„x“ für xCounter	
	„dT“ für delta Time	Wert3 = act. Motor3 Größe

6-8 Einstellen des Time-Outs

Befehl	Funktion	Beschreibung	Beispiel
SMZNM Wert1	Set Maxi Cycle of No Message Wert1	Setzen der Maximalen Anzahl (Wert1), von Durchläufen ohne neues Kommando, bis die Motoren gestoppt werden (Bereich 0 bis 5000)	„SMZNM 20\n“
GMZNM	Get Maxi Cycle of No Message	Abfragen der Anzahl	„GMZNM \n“ -> „MZNM 20\n“

z.B.: Bei einer Zykluszeit von 10ms werden mit „SMZNM 20\n“ nach 200ms ohne neues Kommando, alle drei Motor-Ausgänge abgeschaltet.

6-9 Einstellen der angeschlossenen Impulsgeber und Getriebekonstanten

Für die Geschwindigkeitsermittlung benötigt der Controller die Encoderauflösung und die Maximale Drehzahl (Leerlauf) des Motors (Wert1, Wert2).

Des Weiteren kann über das Summieren der Encoderpulse die zurückgelegte Distanz jedes einzelnen Motors berechnet werden (Wert3).

z.B.:

Befehl	Funktion	Beschreibung	Beispiel
SENCO Mx Wert1 Wert2 Wert3	Set Encoder Constants, Motor x, Wert1 , Wert2 , Wert3	Setzen der Encoder- Konstanten für Motor x = 1 bis 3 Wert1 = 2-fache Encoderauflösung (Bereich 0 bis 5000) Wert2 = Max.Motordrehzahl in [U/min] (Bereich 0 bis 20000) Wert3 = Distanzänderung pro Encoderschritt [$\mu\text{m}/\frac{1}{2}\text{schritt}$] (Bereich 0 bis 1000)	"SENCO M1 1000 11000 22\n"
GENCO Mx	Get Encoder Constants Motor x	Abfrage der Encoder- Konstanten für Motor x = 1 bis 3	"GENCO M1\n" ->"ENCO M1 1000 11000 22\n"

Der Befehl „SENCO M1 1000 10300 22“ setzt die Konstanten für Motor1.

- ⇒ Encoderauflösung 1000 entspricht einem Encoder mit 500 Schritten / Umdrehung (2-fach Imp/U).
- ⇒ Maximale Drehzahl 10300 entspricht 10300U/min Leerlauf Drehzahl des Motor1.
- ⇒ 22 entspricht 22 μm Distanzänderung pro $\frac{1}{2}$ Encoderschritt des Motor1 (unter Berücksichtigung der Getriebeuntersetzung und des Rad-Radius).

6-10 Einstellen der Odometrie

Auf der Basis der Encoder-Information (Motor1 und Motor2) rechnet der Controller die Odometrie des Systems in Encoder Tickauflösung. Als Hilfsgrößen werden die Konstanten des beschriebenen „SENCO Mx ... Wert3“ Kommandos mit einer zusätzliche Rotationswinkel-Konstanten verrechnet.

Die Rotationswinkel-Konstante definiert den Rotations Winkel (in 10^{-6}° oder μ°) des Systems pro Encoder-Tickänderung eines Rades.

Befehl	Funktion	Beschreibung	Beispiel
SMDPT Wert1	Set Micro Degree per Tick (Odometrie Rotation-Angle-Constant) Wert1	Setzen der Odometrie Rotations Winkel Variable Wert1 = X (Bereich 1 bis 20000)	„SMDPT 3599n“
GMDPT	Get Odometrie Rotation-Angle-Constant	Fragt die gesetzte Winkel Konstante ab	„GMDPT n“, -> „MDPT 3599“

Hilfsformel:

$$\alpha_{MDPT} = \frac{180^\circ \Delta s[m]}{\pi \cdot rw[m]}$$

$\Delta s = \text{Dis tan zänderung pro Encodertick}$

$rw = \text{Radanbs tan } d$

z.B.:

$ds = 22,428\mu\text{m}$

$rw = 357\text{mm}$

$\alpha = 68,29\mu\text{m}/357\text{mm} \cdot 180^\circ/\pi = 3599,61 \cdot 10^{-6}^\circ$

$\alpha = 3599\mu^\circ$

6-11 Rücksetzen der Odometrie

Befehl	Funktion	Beschreibung	Beispiel
SRODO	Set Reset Odometrie	Rücksetzen der Odometrie Variablen (X, Y, alpha)	„SRODO n“

6-12 Konfiguration der Drehzahlregelung

Im Arbeitsmodus 2 wird die Drehzahl der angeschlossenen Motoren mit jeweils einem PID Regler geregelt. Der Regler wird durch drei Konstanten parametrisiert.

Wert1 entspricht dem der Proportionalanteil,

Wert2 dem Integralanteil und

Wert3 dem Differentialanteil des Reglers.

Befehl	Funktion	Beschreibung	Beispiel
SVREG Mx Wert1 Wert2 Wert3	Set Velocity Regulator Constants Motor x Wert1 Wert2 Wert3	Setzen der PID-Regel- konstanten für Motor x = 1 bis 3 Wert1 = Regelverstär- kung (Bereich 0 bis 1000) Wert2 = Integralanteil (Bereich 0 bis 1000) Wert3 = Differentialanteil (Bereich 0 bis 1000)	„SVREG M1 10 100 0\n“
GVREG Mx	Get Velocity Regulator Constants Motor x	Abfrage der PID- Regel- konstanten für Motor x = 1 bis 3	„GVREG M1“ ->“VREG M1 10 100 0\n“

Zur Verbesserung der dynamischen Eigenschaften können die Regelparameter verändert werden. Die Anpassung sollte unbedingt im System erfolgen, da dieses erheblichen Einfluss auf die Regelung ausübt.

Hinweis: In der klassischen DC-Drehzahlregelung wird auf den Differentialanteil des Reglers verzichtet, um den Einfluss von Strom- bedingten Momentstößen auf die Mechanik zu vermeiden.

Vorgehensweise:

1. Ausgangskonfiguration setzen:
 Regelverstärkung = 10
 Integralanteil = 300
 Differentialanteil = 0
 „SVREG M1 10 300 0\n“
2. Drehzahlsprung auf 2/3 der Maximaldrehzahl
 „SV 680 0 0\n“
3. Regelverstärkung mit Schrittweite 5 erhöhen
 „SVREG M1 15 300 0\n“
4. Drehzahlsprung auf 2/3 der Maximaldrehzahl
 „SV 680 0 0\n“
5. Drehzahlsprung von 2/3 der Maximaldrehzahl auf Null,
 und das Verhalten beobachten.
 „SV 0 0 0\n“

6. Schritt 3 bis 5 wiederholen bis das System instabil wird. Danach die Regelverstärkung verringern bis sichere Stabilität gegeben ist.
7. Mit dem Integralanteil entsprechend den Schritten 3 bis 5 verfahren. (mit negativer Strittweite von 20)

6-13 Konfiguration der Strombegrenzung

Im Arbeitsmodus 2 wird hinter die Drehzahlregelung jeweils eine Strombegrenzung geschaltet.

Diese Strombegrenzung hat die Aufgabe die angeschlossenen Motoren vor Überlast zu schützen. Dies erfolgt durch hinzunahme des Thermischen Modells (I^2t Strombegrenzung).

Solange das Thermische Modell keine kritische Motorwicklungs-Temperatur berechnet, wird der Motorstrom auf den Maximal zulässigen Motorstrom begrenzt (Wert 2). Da das Antriebsmoment des DC-Motors proportional zum Motorstrom ist, lässt sich somit auch eine Momentregelung realisieren.

Sobald das Thermische Modell eine kritische Wicklung- Temperatur berechnet, wird der Motorstrom auf den Nennstrom begrenzt (Wert1).

Diese Begrenzung erfolgt bis das Thermische Modell eine unkritische Minimal Temperatur berechnet hat.

Befehl	Funktion	Beschreibung	Beispiel
SCLIM Mx Wert1 Wert2	Set Current Limits Motor x Wert1 Wert2	Setzen der Strombegrenzungslimits für Motor x = 1 bis 3 Wert1 = Maxi. zulässige Motorstrom (Bereich 0 bis 1000) Wert2 = Nennstrom (Bereich 0 bis 1000)	„SCLIM M1 306 102\n“
GCLIM Mx	Get Current Limits Motor x	Abfrage der Strombegrenzungslimits für Motor x = 1 bis 3	„GCLIM M1“ ->“CLIM M1 306 102\n“

Auflösung der Strommessung:

Der Messbereich 0 bis 5A wird auf 0 bis 1024 mit 10Bit Auflösung abgebildet. Wobei die Messelektronik bei ca. 850 (83%) in die Sättigung geht.

Anmerkung:

Der Strommessbereich kann durch Einlötlung von Jumpers auf der Platine verdoppelt werden (0 bis 10A).

Es ist darauf zu achten, dass die Strommessung nur bei rein Ohmscher-Last linear ist, sich aber bei Ohmsch-Induktiver-Last quadratisch verhält.

6-14 Abfragen der Versorgungsspannung

Befehl	Funktion	Beschreibung	Beispiel
GADC UB	Get AD-Converter Value U-Battery	Abfrage der Versorgungs- Spannung (Bereich 0 bis 1024)	„GADC UB“ ->„ACD UB 517“

Der Rückgabewert entspricht der Betriebsspannung mit dem Faktor $1/23 [V / \text{ticks}] + 1,5[V]$.

z.B.: Rückgabe Wert = 517[ticks]

$517[\text{ticks}] / 23 [V / \text{ticks}] + 1,5[V] = 24[V]$ Betriebsspannung

6-15 Abfragen der Motor-Wicklungs-Temperatur

Befehl	Funktion	Beschreibung	Beispiel
GWTEM	Get Winding Temperatures	Abfrage der Wicklungs- Tem- peratur aller drei Motoren (Bereich 0 bis 1000 °C)	„GWTEM“ ->„WTEM 250 25 25“

Die drei Rückgabe- Werte entsprechen den Wicklungs-Temperaturen in °C der Motoren 1 bis 3.

6-16 Konfiguration der I/O Port Pins

Die Pins des I/O Steckers sind wie folgt konfigurierbar:

Nr. (Amp.20)	Bezeichnung
1	Port1.6, only Output (LSB)
3	Port1.7, only Output

Pin 1 und 3 sind reine Output-pins.

Mit dem folgendem Kommando können ihre Zustände geändert werden.

z.B.: Pin Nr.1 und Nr.3 auf „Low“ setzen „SP1L 0\n“

Befehl	Funktion	Beschreibung	Beispiel
SP1L Wert	Set Port 1L Wert	Setzen der Output-Pins des Ports 1L auf Wert = 0 bis 3 ($2^2 - 1$)	„SP1L 0\n“

Pin Nr.1 auf „High“ und Nr.3 auf „Low“ setzen „SP1L 1\n“

Pin Nr.1 auf „High“ und Nr.3 auf „High“ setzen „SP1L 3\n“

Die folgenden Pins können einzeln als In- oder Output Pins genutzt werden:

Mit dem folgendem Kommando kann ihre Konfiguration geändert werden.

Nr.(Amp.20)	Bezeichnung
5	Port1.8 (LSB)
7	Port1.9
9	Port1.10
11	Port1.11
13	Port1.12
15	Port1.13
17	Port1.14
19	Port1.15 (MSB)

Befehl	Funktion	Beschreibung	Beispiel
SPORT1H DIR Wert	Set Port 1H Direction Wert	Setzen der Pin-Richtung des Ports 1H auf In- oder Output Wert = 0 bis 255 ($2^8 - 1$) 0 Bit = Output 1 Bit = Input	„SPORT1H DIR 0\n“

z.B.: Nr.5 bis 19 als „Output-Pin“ setzen „SPORT1H DIR 255\n“ und z.B. mit „SP1H 15\n“ beschreiben.

Nr.5 bis 11 als „Input-Pin“ und 13 bis 19 als „Output-Pin“ setzen „SPORT1H DIR 15\n“

Nr.5 bis 19 auf „Input-Pin“ setzen „SPORT1H DIR 0\n“, und z.B. mit „GP1H \n“ auslesen.

Hinweis: Im Einschaltzustand sind die Pins Nr.5 bis Nr.19 als Inputs konfiguriert (default).

Befehl	Funktion	Beschreibung	Beispiel
SP1H Wert	Set Port 1H Wert	Setzen der als „Output“ definierten Pins des Ports 1H auf „High“ oder „Low“ Wert = 0 bis 255 ($2^8 - 1$) 0 Bit = „Low“ 1 Bit = „High“	„SP1H 240“
GP1H	Get Port 1H	Abfragen der als Input definierten Pins des Ports 1H auf den extern angelegten Pegel	„GP1H “ ->“P1H 15“

Hinweis:

Es sollte auf sauber Signaltrennung zwischen den Pins und der angeschlossenen Elektronik geachtet werden. Hier bieten sich Optokoppler an.

Bei der Nutzung der I/O Kommandos ist besondere Vorsicht geboten, denn durch falsches konfigurieren kann leicht ein Pin- Treiber zerstört werden.

z.B: Ein Input- Pin mit externen Low- Treibender Elektronik wird zum Output- Pin High- Treibend (die schwächere Elektronik wird zerstört)!

7 Grundeinstellung

Allgemein:

Timeout "SMZNM 20\n"

200ms

Antwort auf „SV“, „SEND 2\n“

aktuelle Geschwindigkeit, Motorströme und Odometrie

Controller Mode „SMODE 2\n“

Geschwindigkeits- und Stromregelung

Für Motor $x = 1$ bis 3

Encoder Motor ... "SENCO Mx 1000 6000 22\n"

1000 [2*Ticks/rot] Encoder

6000 [U/min] Max Motor Speed

22[μ m/Tick] linear Distance pro $\frac{1}{2}$ EncoderTick

Regler „SVREG Mx 5 80 0\n“

Regelverstärkung = 10

Integralanteil = 80

Differentialanteil = 0

Strombegrenzung „SCLIM Mx 200 60\n“

Max. Strom = 200 (3[A])

Nennstrom = 60 (1,1[A])

9 Regler-Struktur Diagramm

Folgendes Diagramm veranschaulicht den Datenfluss für eine Motor-Regelschleife.

Im „Mode 0“ entspricht
der Wert des „SV“
Kommandos der
„Output Line“.

Im „Mode 1“ entspricht
er der „Velo Out Line“,

und im „Mode 2“ ist
der Wert des „SV“
Kommandos der
„Des.Velocity Line“
zuzuordnen.