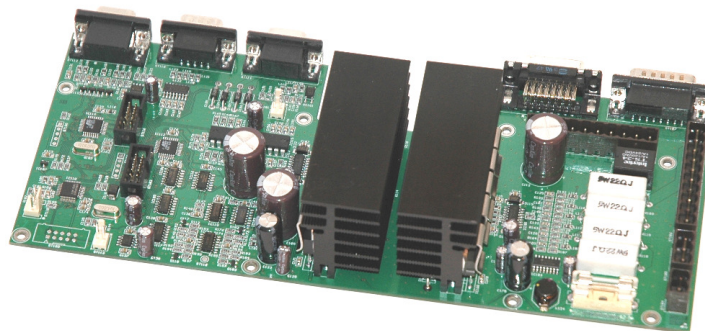


Motorsteuerungen CM-Serie

Benutzerhandbuch

Version 1.0



Calantec GmbH Automatisierungstechnik, Industrieelektronik

Aufgang B
Helmholtzstraße 2-9
10587 Berlin
<http://www.calantec.de>

Telefon: 030 453 01 519
Telefax: 030 453 01 531
email: info@calantec.de

Geschäftsführer: Hartmut Schäfer
Handelsregister: Amtsgericht Charlottenburg HRB 92065
USt-ID: DE232787423

Programmversion 1.03
18. Dezember 2006
© Hartmut Schäfer

Inhaltsverzeichnis

Kommunikation.....4
 Befehlssatz.....4
 A Beschleunigung.....4
 B Anhalten.....4
 C Konfiguration.....5
 D momentane Geschwindigkeit.....6
 E Echo.....7
 H Referenzfahrt.....7
 I Motorstrom – Bewegung.....7
 J Motorstrom – Standby.....7
 M Motor einschalten/ausschalten.....8
 N Notaus.....8
 P Absolut Positionieren.....8
 R Relativ Positionieren.....8
 T Sollposition abfragen.....8
 V Geschwindigkeit.....8
 X Status.....9
 Z Nullposition.....10

Kommunikation

Die Steuerung ist mit einem seriellen Interface ausgestattet, der genaue Typ ist im Datenblatt der Steuerung beschrieben. Die Baudrate beträgt in der Regel 19200 Baud, keine Parität, 8 Datenbits, 1 Stopbit. Es wird RTS/CTS-Handshake verwendet. Die Steuerung liefert das Zeichen „>“ als Prompt, wenn sie bereit ist, Befehlszeilen entgegenzunehmen. Eine Befehlszeile besteht aus bis zu 64 Zeichen, die mit einem Carriage-Return (#13) abgeschlossen werden. Wird in einer Zeile ein Syntax-Fehler gefunden, wird als nächstes Prompt „?“ verwendet. Wurde mit einer Befehlszeile eine Motorbewegung ausgelöst, wird als Prompt „-“ verwendet, solange die Bewegung nicht beendet ist. Somit kann durch Übertragen von Leerzeilen und Auswertung des Prompt das Bewegungsende abgewartet werden.

Alle eingestellten Parameter werden im EEPROM gespeichert und stehen auch nach dem Ausschalten zur Verfügung.

Der Befehlssatz der CL-Steuerungen kann erweitert und an kundenspezifische Anforderungen angepasst werden.

Ein Leerzeichen zwischen dem Befehlsbuchstaben und der ersten Zahl ist nicht nötig (bei den Beispielen wegen der besseren Lesbarkeit eingesetzt).

Werden alle Parameter angegeben, können diese auch durch Leerzeichen getrennt werden. Nur wenn ein Parameter ausgelassen wird, muß dies durch ein Komma angezeigt werden.

Befehle, die einen oder mehrere Parameter zurückgeben, schließen die Ausgabe mit einem Carriage-Return (#13) ab. Mehrere Parameter werden durch Kommata getrennt.

n1 Parameter für Motor 1
 n2 Parameter für Motor 2
 a Achsnummer
 [] optionaler Parameter

Befehlssatz

Der Befehlssatz und die Kommunikation können auf Kundenwunsch beliebig modifiziert und erweitert werden. Zusätzlich können ab Werk auch komplexe, kundenspezifische Programme (SPS-Ablauf, Regelalgorithmen etc.) hinterlegt werden.

A Beschleunigung

Syntax: A [n1] [,n2]

Einstellung der Beschleunigung in 931,3 Mikroschritten / s². n=1..10000

Interne Skalierung: (n/65536)/(128 µs)²

Beispiel:

A 100 setzt Beschleunigung von Motor 1
 A ,100 setzt Beschleunigung von Motor 2

A 100,200 setzt Beschleunigung von Motor 1 und Motor 2

B Anhalten

Syntax: B [a]

Die Motorbewegung wird mit der eingestellten Verzögerung angehalten.

Beispiel:

B	bremsst beide Motoren
B 0	bremsst beide Motoren
B 1	bremsst Motor 1
B 2	bremsst Motor 2

C Konfiguration

Konfiguration des Kommunikationscontrollers und der Motorcontroller

Zur Unterscheidung, welcher Controller zu programmieren ist, gibt es zwei Syntax-Varianten:

Syntax 1: C n,m der Kommunikationscontroller wird konfiguriert

Syntax 2: C a,n,m der Motorcontroller a wird konfiguriert

Wird bei Syntax 2 als Controllernummer 0 angegeben, so werden alle Motorcontroller mit dem gleichen Wert konfiguriert.

Konfiguration des Kommunikationscontrollers

n=31 Einstellung des Joystick-Modus:

m=0: der Joystick ist immer aus

m=1: der Joystick ist immer eingeschaltet (dieser Modus sollte nicht verwendet werden)

m=3: der Joystick kann mit dem Schalter ein- und ausgeschaltet werden

n=100..115 Schreiben der Benutzer-Variablen im RAM. Diese Speicherzellen werden beim Einschalten der Steuerung auf 0 gesetzt. Es können 32-Bit-Werte im Bereich von m=-2147483648..+2147483647 verwendet werden.

Beispiel:

C 100 123 die Speicherzelle 100 wird auf den Wert 123 gesetzt

n=200..215 Schreiben der Benutzer-Variablen im EEPROM. Der Inhalt dieser Speicherzellen bleibt auch beim Ausschalten der Steuerung erhalten. Es können 32-Bit-Werte im Bereich von m=-2147483648..+2147483647 verwendet werden.

Beispiel:

C 200 01082004 die Speicherzelle 200 wird auf den Wert 01082004 gesetzt (z.B. Datum)

Konfiguration der Motorcontroller

n=0 Die aktuelle Joystickposition wird als Mittelposition gespeichert. Der

- Parameter m wird nicht ausgewertet, muß aber angegeben werden (z.B. 0).
- n=1 Die erste Joystickschwelle wird eingestellt ($m=0..255$)
- n=2 Die zweite Joystickschwelle wird eingestellt ($m=0..255$)
- n=3 Der erste Joystick-Geschwindigkeitsfaktor wird eingestellt ($m=0..255$)
- n=4 Der zweite Joystick-Geschwindigkeitsfaktor wird eingestellt ($m=0..255$)
- n=5 Die für die Joystickfahrt anzuwendende Beschleunigung wird eingestellt ($m=0..10000$)
Die Joystick-Geschwindigkeit berechnet sich folgendermaßen:
 $v = \text{Pos}(\text{AD-Schwelle1}) * \text{Faktor1} + \text{Pos}(\text{AD-Schwelle1-Schwelle2}) * \text{Faktor2}$
 { Def.: $\text{Pos}(x) = x$ für $x \geq 0$ und $\text{Pos}(x) = 0$ für $x < 0$ }
 $\text{AD} = \text{Abs}(\text{AD_Wandler_Wert} - \text{Korrekturfaktor}) / 4$ (Korrekturfaktor ca. 512)
- n=30 Die für den Motor zuständige Joystickachse wird ausgewählt
 $m=0$ X-Achse
 $m=1$ Y-Achse
 $m=2$ Drehknopf
- n=32 Die Joystickpolarität wird eingestellt
 $m=1$ normal
 $m=-1$ umgekehrte Drehrichtung
- n=34 Einstellung der Maske für die Limitschalter in positiver Richtung. Dabei sind die Bits 0..7 des Parameters m den Eingängen 1..8 zugordnet. Ist ein Bit gesetzt, wird der entsprechende Eingang überwacht, ist es 0, wird der Eingang maskiert.
Beim Überfahren des Limitschalters in positiver Richtung wird die Fahrt unterbrochen (siehe n=38), es kann dann nur in negativer Richtung weitergefahren werden.
- n=35 Einstellung der Polarität der Limitschalter in positiver Richtung. Werden Öffner verwendet, muß das zum Eingang korrespondierende Bit auf 1 gesetzt werden, bei Schließern auf 0. Hinweis: Die Referenzfahrt funktioniert nur mit Öffnern und aus Sicherheitsgründen (Kabelbruch, Stecker wird abgezogen) sollten auch nur Öffner verwendet werden.
- n=36 Einstellung der Maske für die Limitschalter in negativer Richtung. Siehe n=34.
- n=37 Einstellung der Polarität der Limitschalter in negativer Richtung. Siehe n=35.
- n=38 Ist $m=0$ (Werkseinstellung), wird die Fahrt beim Erreichen eines Limitschalters sofort angehalten (eventueller Nachteil: Schrittverlust), ist $m=1$ wird mit der eingestellten Verzögerung abgebremst (Nachteil: Bremsweg eventuell zu lang)
Hinweis: Während einer Referenzfahrt ist die Limitschalterüberwachung

ausgeschaltet!

D momentane Geschwindigkeit

Syntax: D [a]

Die momentane Geschwindigkeit eines oder mehrerer Motoren wird ausgegeben.

Beispiel:

D gibt die Geschwindigkeit aller Motoren durch Kommata
getrennt aus
D 1 gibt die Geschwindigkeit von Motor 1 aus
D 2 gibt die Geschwindigkeit von Motor 2 aus

E Echo

Syntax: E n

Zu Testzwecken kann mit $n=1$ ein Echo der seriell übertragenen Zeichen eingeschaltet werden. Diese Einstellung wird im EEPROM gespeichert.

Beispiel:

E 0 Echo wird ausgeschaltet
E 1 Echo wird eingeschaltet

H Referenzfahrt

Syntax: H [n1] [,n2]

Das Achssystem führt eine Referenzfahrt durch. Dabei wird der mit $n1$ bzw $n2$ angegebene Eingang überwacht, das Vorzeichen der Zahl bestimmt die Fahrtrichtung. Motor 1 hat die Eingänge 1 bis 4 (3 und 4 nur am D-Sub Stecker), Motor 2 hat nur Eingang 1. Bei einer positiven Zahl wird der Endschalter in positiver Richtung gesucht, bei einer negativen Zahl in negativer Richtung. Der Referenzschalter muß ein Öffner sein.

Beispiel:

H 1, 1 beide Motoren suchen Referenzschalter 1 in positiver Richtung
H , -1 Motor 2 sucht den Referenzschalter 1 in negativer Richtung
H -2 Motor 1 sucht Referenzschalter 2 in negativer Richtung

I Motorstrom – Bewegung

Syntax: I [n1] [,n2]

Einstellung des Motor-Nennstroms in Prozent. $n=0$ ist Minimum, $n=100$ ist Maximum. Motor 1: maximal 4 A (Skalierung 100%=5 A), Motor 2: maximal 1.5 A (Skalierung 100%=1.5 A)

Beispiel:

I 80 der Nennstrom für Motor 1 wird eingestellt (4 A)
I , 70 der Nennstrom für Motor 2 wird eingestellt (1.0 A)
I 80, 70 der Nennstrom für Motor 1 und 2 wird eingestellt

J Motorstrom – Standby

Syntax: J [n1] [,n2]

Einstellung des Motor-Standbystroms in Prozent. n=0 ist Minimum, n=100 ist Maximum. Motor 1: maximal 4 A (Skalierung 100%=5 A), Motor 2: maximal 1.5 A (Skalierung 100%=1.7 A)

Beispiel:

J 50	der Standby-Strom für Motor 1 wird eingestellt (2.5 A)
J , 40	der Standby-Strom für Motor 2 wird eingestellt (0.7 A)
J 50, 40	der Standby-Strom für Motor 1 und 2 wird eingestellt

M Motor einschalten/ausschalten

Syntax: M [n1] [, n2]

Bei n=0 wird der Motor ausgeschaltet, bei n=1 wird der Motor eingeschaltet

Beispiel:

M 1	Motor 1 wird eingeschaltet
M 0, 1	Motor 1 wird ausgeschaltet, Motor 2 wird eingeschaltet
M , 0	Motor 2 wird ausgeschaltet

N Notaus

Syntax: N

Schaltet die Endstufenstromversorgung ab.

P Absolut Positionieren

Syntax: P [n1] [, n2]

Fahrt zur Position n1 bzw. n2. n=-2147483648..+2147483647

P 1000	Motor 1 fährt zur Position 1000
P , -50	Motor 2 fährt zur Position -50
P -50, 300	Motor 1 fährt zur Position -50, Motor 2 zur Position 300

R Relativ Positionieren

Syntax: R [n1] [, n2]

Relativfahrt um n1 bzw. n2 Schritte ab aktueller Position.

n=-2147483648..+2147483647

R 1000	Motor 1 verfährt um +1000 Schritte
R , -50	Motor 2 verfährt um -50 Schritte
R -50, 300	Motor 1 verfährt um -50 Schritte, Motor 2 um +300

T Sollposition abfragen

Syntax: T [a]

Die aktuelle Sollposition einer oder mehrerer Motoren wird ausgegeben.

Beispiel:

T	gibt die Position aller Motoren durch Kommata getrennt aus
T 1	gibt die Position von Motor 1 aus
T 2	gibt die Position von Motor 2 aus

V Geschwindigkeit

Syntax: V [n1] [,n2]

Einstellung der Geschwindigkeit in 30,52 Mikroschritten / s. n=0..65535

Interne Skalierung: (n/256)/128 μ s

Beispiel:

V 500 die Geschwindigkeit für Motor 1 wird eingestellt
 V , 700 die Geschwindigkeit für Motor 2 wird eingestellt
 V 500, 700 die Geschwindigkeit für Motor 1 und 2 wird eingestellt

X Status

Status des Kommunikationscontrollers und der Motorcontroller

Zur Unterscheidung, welcher Controller abgefragt wird, gibt es zwei Syntax-Varianten:

Syntax 1: X n der Kommunikationscontroller wird abgefragt

Syntax 2: X a, n der Motorcontroller a wird abgefragt

Wird bei Syntax 2 als Controllernummer 0 angegeben, so werden alle Motorcontroller abgefragt und die Werte durch Kommata getrennt ausgegeben.

Status des Kommunikationscontrollers

- n=0 Es wird der Wert 1 zurückgegeben, wenn die Endstufenstromversorgung eingeschaltet ist, 0 wenn nicht.
- n=1 Der Zustand der Eingänge des Kommunikationscontrollers wird ausgegeben. Dabei ist Bit 0 auf 0, wenn der Joystick angeschlossen und eingeschaltet ist. Bit 1 entspricht Eingang 3 von Motorcontroller 1 und Bit 2 entspricht Eingang 4 von Motorcontroller 1. Bit 3 entspricht Eingang 5 (Pin 6 am Motor/Referenzschalterstecker)
- n=2 Die Spannung der 24 V-Versorgung wird gemessen. Ausgabe in 100 mV, Genauigkeit ca. 5%.
- n=3 Die Spannung der 42 V-Versorgung wird gemessen. Ausgabe in 100 mV, Genauigkeit ca. 5%.
- n=4 Es wird die Controllerbezeichnung und die Software-Version ausgegeben.
- n=5 Die Seriennummer des Controllers wird ausgegeben.
- n=6 Der letzte Fehler wird im Klartext ausgegeben.
- n=7 Der aktuell eingestellte Joystick-Modus (siehe C, n=31) wird ausgegeben.
- n=8 Der aktuelle Zustand des Joysticks wird angegeben. Der Wert ist 1, wenn der Joystick eingeschaltet ist, und 0, wenn der Joystick ausgeschaltet oder nicht angesteckt ist.
- n=100..115 Auslesen der Benutzer-Variablen im RAM. Siehe (siehe C, n=100..115)
- n=200..215 Auslesen der Benutzer-Variablen im EEPROM. Siehe (siehe C,

n=200..215)

Status der Motorcontroller

- n=0 Der Zustand der Eingänge wird ausgegeben (Eingang 1 entspricht Bit 0, Eingang 2 entspricht Bit 1 etc.)
- n=1 Rückgabe 1, wenn Motor eingeschaltet, sonst 0
- n=2 Rückgabe 1, wenn der betreffende Motor sich bewegt, sonst 0
- n=3 Der aktuelle Wert des AD-Wandlers (Joystick) wird ausgegeben, Bereich 0..1023
- n=4 Die Programmversion und Revision des Motorcontrollers wird ausgegeben

Z Nullposition

Syntax: Z [n1] [,n2]

Die aktuelle Position wird als Position n übernommen.

n=-2147483648..+2147483647

Beispiel:

- Z 0 Die Position von Motor 1 wird auf 0 gesetzt
- Z ,1000 Die Position von Motor 2 wird auf 1000 gesetzt
- Z -1000,0 Die Position von Motor 1 wird auf -1000 gesetzt, die von Motor 2 auf 0