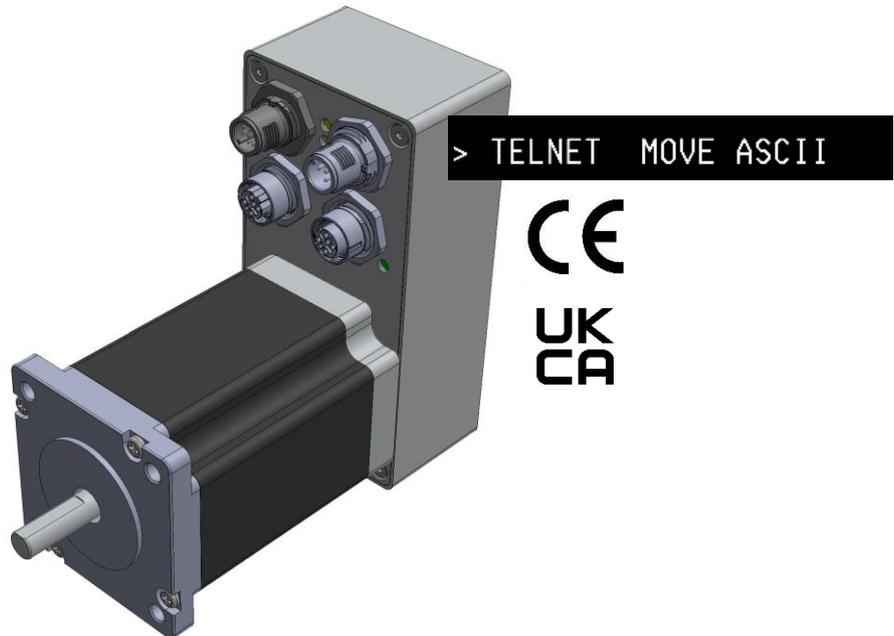


Spezifikation Colibri 4.0 Telnet



Diese Spezifikation beschreibt die Kommunikation mit dem Colibri 4.0 über eine Ethernet-Verbindung per ASCII-Befehle.

Version: 1.0

Datum: 17.03.2024

Inhalt

Spezifikation Colibri 4.0 Telnet.....	1
Verbindungsaufbau über Telnet	6
Als 1. Beispiel wird hier eine Bewegung erklärt:	6
Telnet-Lib in Python	7
Protokoll-Definition.....	8
Telegrammaufbau.....	8
Antworttelegramme	8
Fahrbefehle	9
MOTOR.....	9
Format	9
Beispiel: Endstufe einschalten	9
REF	9
Format	9
Beispiel: Referenziere auf den Positiven Anschlag	10
MOVE.....	10
Format	10
Beispiel: Fahre absolut auf die Zielposition 500 mit Frequenz 4000 Hz.....	11
FREQ.....	11
Format	11
Beispiel: Fahre endlos mit 4000 Hz, Beschleunigung und Verzögerung von 10000 Hz/s.....	11
PRINTMARK	12
Format	12
Beispiel: Druckmarkenfahrt Maxweg 1000 Schritte, mit 750 Hz, ACC/DEC 5000Hz/s	13
MARK	13
Format	13
Beispiel Markierfahrt Absolut mit 2 Absolut-Markierpositionen:	14
Beispiel Markierfahrt Relativ mit 2 Relativ-Markierpositionen:	15
Beispiel Markierfahrt Absolut mit zyklischen Markierpositionen:	15
TRACE	16
Format	16
Beispiel: Trace-Fahrt	17
Digitale Ein/Ausgänge bedienen.....	18
DIGOUT.....	18

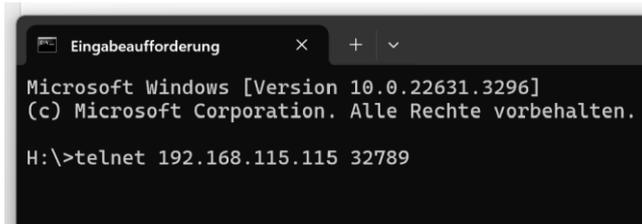
Format	18
Beispiel: Setze Ausgang 0 auf 1	18
DIGIN	18
Format	18
Beispiel: Lese Eingang 2.....	19
SIGIN	19
Format	19
Beispiel: Abfrage Eingangs-Signal.....	19
Werte und Parameter abfragen und verändern.....	20
PW	20
Format	20
Beispiel: Gib die Berechtigungsstufe mit dem Passwort 1234 frei	20
POS.....	20
Format	20
Beispiel: Aktuelle Position abfragen.....	20
STAT.....	21
Format	21
Beispiel: Abfragen des Status.....	21
LIST.....	21
Format	21
Beispiel: Gib alle Parameter für die aktive Berechtigungsstufe aus	22
PARAM	23
Format	23
Beispiel: Setze den Wert des Parameters mit dem Index 10 und Subindex 2 auf 123	23
PROFILE	23
Format	23
Beispiel: Ändere Dynamikprofil 2 auf Maximalgeschwindigkeit 6000 Hz und Beschleunigung sowie Verzögerung 15000 Hz/s	24
SCAN	24
Format	24
Beispiel: Melde Parameter mit dem Index 10 und dem Subindex 1 mit einem Intervall von 500 ms an	25
ENC	25
Format	26
Beispiel: Abfrage Encoder Rohwert.....	26

UMOT	26
Format	26
Beispiel: Auslesen Motorspannung	26
U24	27
Format	27
Beispiel: Auslesen Steuerspannung	27
TETA	27
Format	27
Beispiel: Auslesen Temperatur an der Endstufe	27
IDRIVE	28
Format	28
Beispiel: Auslesen des Fahrstroms	28
IHOLD	28
Format	28
Beispiel: Auslesen des Haltestroms	28
IREF	28
Format	29
Beispiel: Auslesen des Referenzstroms	29
SAVE	29
Format	29
Beispiel: Speichern im EEPROM	29
Systembefehle.....	30
RESTART	30
Format	30
Beispiel: Neustart.....	30
RESET	30
Format	30
Beispiel: Fehler löschen.....	30
REMOTE	31
Format	31
Beispiel: Gibt den Status des Remote-Modus aus	31
SELFTEST	31
Format	31
Beispiel: Starte Self-Test.....	31

Heartbeat	32
Format	32
Beispiel: Heartbeat aktivieren mit einer Intervallzeit von 500ms	32

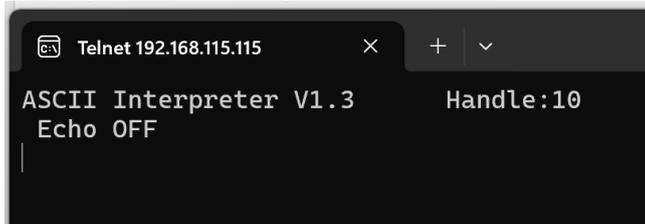
Verbindungsaufbau über Telnet

- Die Kommunikation erfolgt über eine Telnet TCP Verbindung über IP-Adressen, hierbei wird mit dem jeweiligen Motor eine separate Telnet-Verbindung aufgebaut (IP-Adresse und Port)
- Der Antrieb ist über den Port 32789 per Telnet erreichbar
- z.B. Aufbau über die Windows Kommandozeile:
Im Beispiel hat der Motor die IP-Adresse 192.168.115.115



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. Alle Rechte vorbehalten.
H:\>telnet 192.168.115.115 32789
```

Bei erfolgreichem Verbindungsaufbau meldet sich der Motor wie folgt:



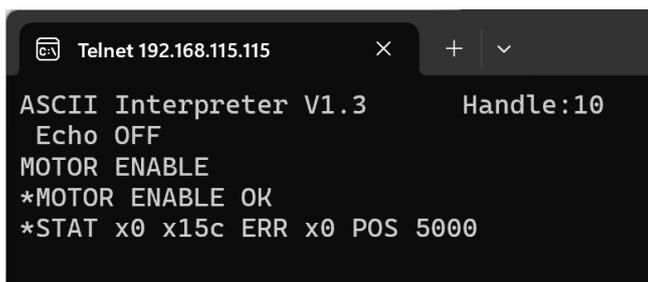
```
Telnet 192.168.115.115
ASCII Interpreter V1.3      Handle:10
Echo OFF
```

Jetzt kann mit dem Motor per ASCII-Befehle kommuniziert werden, Befehle siehe die nachfolgenden Kapitel.

Als 1. Beispiel wird hier eine Bewegung erklärt:

Zunächst muss der Motor enabled werden:

Befehl: MOTOR ENABLE



```
Telnet 192.168.115.115
ASCII Interpreter V1.3      Handle:10
Echo OFF
MOTOR ENABLE
*MOTOR ENABLE OK
*STAT x0 x15c ERR x0 POS 5000
```

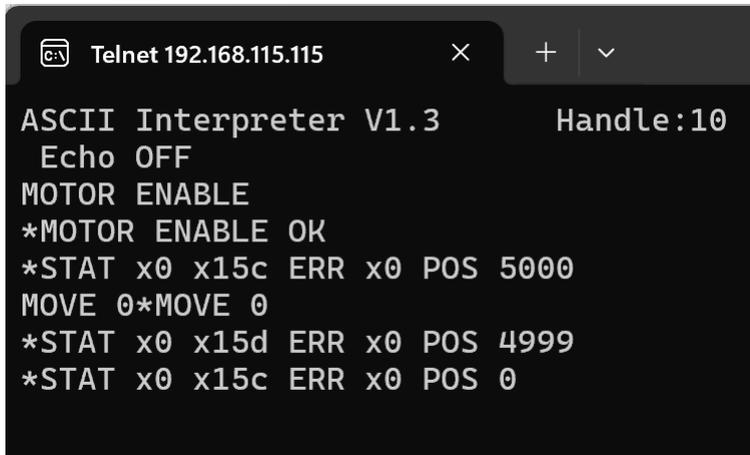
Der Motor antwortet mit

```
*MOTOR ENABLE OK
*STAT X0 X15c ERR x0 POS 5000
```

Das bedeutet der Befehl Enable war erfolgreich und der Motor sendet noch seinen aktuellen Status mit, momentane Position ist die 5000.

Fahrt auf Position 0

Befehl: MOVE 0



```
Telnet 192.168.115.115
ASCII Interpreter V1.3      Handle:10
Echo OFF
MOTOR ENABLE
*MOTOR ENABLE OK
*STAT x0 x15c ERR x0 POS 5000
MOVE 0*MOVE 0
*STAT x0 x15d ERR x0 POS 4999
*STAT x0 x15c ERR x0 POS 0
```

Der Motor antwortet mit

```
*MOVE 0
*STAT X0 X15d ERR x0 POS 4999
*STAT X0 X15c ERR x0 POS 0
```

Das bedeutet der Befehl Move wird ausgeführt

- beim Start der Bewegung wird der Status gesendet
- beim Erreichen der Zielposition wird erneut der Status gesendet

Telnet-Lib in Python

Hier ist die Python Beschreibung wie Telnet aufgebaut wird und wie hier eine Kommunikation stattfinden kann:

<https://docs.python.org/3/library/telnetlib.html>

Protokoll-Definition

- Befehle, die zum Antrieb gesendet werden, sind nicht Case-Sensitiv.
- Keine Unterscheidung zwischen Groß-/Kleinschreibung
- Antworten vom Antrieb sind immer Upper-Case
- Jedes Telegramm muss mit einem Zeilenumbruch ASCII 0x13 0x10 abgeschlossen werden.
- Innerhalb eines Telegramms dürfen keine Zeilenumbrüche vorkommen
- Der Antrieb sendet selbstständig keine Telegramme ohne vorherige Anfrage, Ausnahme sind hierbei die Befehle SCAN und REMOTE
- Alle Zahlenwerte können als Hex übergeben, hierzu muss der Parameter mit dem Präfix X beginnen, z.B. MOVE A X01F4 für MOVE A 500

Telegrammaufbau

Alle Blöcke eines Telegramms sind mit Leerzeichen ' ' ASCII 0x32 getrennt.

Beispiel:

```
COMMAND PARAM1 PARAM2
```

- Beinhaltet ein Telegramm ein Array, müssen dessen Elemente mit einem Semikolon ';' ASCII 0x59 getrennt werden.
- Vor dem ersten Element muss sich ebenfalls ein Semikolon ';' ASCII 0x59 befinden.
- Innerhalb eines Arrayelements werden Parameter wiederum mit einem Leerzeichen ' ' ASCII 0x32 getrennt.

Beispiel:

```
COMMAND PARAM1 ; PARAM2 PARAM3 ; PARAM4 PARAM5
```

Alle Befehle lassen sich somit zerlegen, indem sie zuerst per Semikolon ';' ASCII 0x59 und die daraus entstandenen Elemente nochmals per Leerzeichen ' ' ASCII 0x32 getrennt werden.

Antworttelegramme

Antworttelegramme (mit Ausnahme der Befehle HELP und ?) beinhalten immer den ausgeführten Befehl mit dem Prefix '*' ASCII 0x42, zudem können Zusatzinformationen enthalten sein.

Der Befehl zum Aktivieren der Endstufe wird an den Antrieb gesendet

```
MOTOR ENABLE
```

Als Antwort wird bei erfolgreicher Durchführung folgendes Telegramm gesendet

```
*MOTOR ENABLE
```

Fahrbefehle

Parameter in Klammern () sind optionale Parameter, die möglichen Kombinationen der Parameter sind im Kapitel des Entsprechenden Befehls zu finden.

MOTOR

Senden von Motorkommandos

Format

Befehl:

MOTOR <Option>

Antwort:

*MOTOR <Option>

<OPTION>

ENABLE

DISABLE

STOP

ESTOP

BEDEUTUNG

Einschalten der Endstufe

Ausschalten der Endstufe

Anhalten des Antriebs mit Rampen

Anhalten des Antriebs ohne Rampen

Beispiel: Endstufe einschalten

Befehl:

MOTOR ENABLE

Antwort:

*MOTOR ENABLE

REF

Ausführen der Referenzfahrt mit Angabe der Methode(optional).

Format

Befehl:

REF (<Methode>)

Antwort:

*REF (<Methode>)

Mögliche Kombinationen

REF

REF <Methode>

<METHODE>

0

1

2

3

ART DER REFERENZFAHRT

Keine Referenzfahrt, Position 0 setzen

auf Anschlag in positiver Richtung

auf Anschlag in negativer Richtung

auf Referenzschalter in positiver Richtung

4	auf Referenzschalter in negativer Richtung
5	auf Endschalter in positiver Richtung
6	auf Endschalter in negativer Richtung
7	Doppelblockreferenzfahrt

Angabe optional: Sollte dieser Wert nicht gesetzt sein, wird die Methode verwendet, die im zugehörigen Parameter eingestellt ist.

Beispiel: Referenziere auf den Positiven Anschlag

Befehl:

REF 1

Antwort:

*REF 1

MOVE

Ausführen eines Bewegungsbefehls

Format

Befehl:

MOVE <Option> (X)<Steps> ((X)<Frequency>) ((X)<Acceleration>) ((X)<Deceleration>)

Antwort:

*MOVE <Option> (X)<Steps> ((X)<Frequency>) ((X)<Acceleration>) ((X)<Deceleration>)

Mögliche Kombinationen

MOVE <Option> (X)<Steps>

MOVE <Option> (X)<Steps> (X)<Frequency>

MOVE <Option> (X)<Steps> (X)<Frequency> (X)<Acceleration>

MOVE <Option> (X)<Steps> (X)<Frequency> (X)<Acceleration> (X)<Deceleration>

<OPTION>

A	Absolute Bewegung
R	Relative Bewegung

<STEPS>

DWORD: -...0...+	Zielposition bzw. Distanz in Schritten (1 Umdrehung = 400 Schritte)
------------------	------------------------------------------------------------------------

<FREQUENCY>

DWORD: 0...20000 HZ	Zielgeschwindigkeit in Hz 400Hz = 1 Umdrehung/s
---------------------	----------------------------------------------------

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Maximalfrequenz, Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<ACCELERATION>

DWORD: 0...100000 HZ/S	Beschleunigung in Hz/s 400Hz/s = 1 U/s ²
------------------------	--------------------------------------------------------

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<DECELERATION>

DWORD: 0...100000 HZ/S

Verzögerung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, wird für die Verzögerung der für die Beschleunigung gültige Wert verwendet.

Beispiel: Fahre absolut auf die Zielposition 500 mit Frequenz 4000 Hz**Befehl:**

MOVE A 500 4000

Antwort:

*MOVE A 500 4000

FREQ

Ausführen einer Endlosbewegung mit variablen Dynamikwerten.

Format**Befehl:**

FREQ (X)<Frequency> ((X)<Acceleration>) ((X)<Deceleration>)

Antwort:

*FREQ (X)<Frequency> ((X)<Acceleration>) ((X)<Deceleration>)

Mögliche Kombinationen

FREQ (X)<Frequency>

FREQ (X)<Frequency> (X)<Acceleration>

FREQ (X)<Frequency> (X)<Acceleration> (X)<Deceleration>

<FREQUENCY>

DWORD: 0...20000 HZ

Zielgeschwindigkeit in Hz
400Hz = 1 Umdrehung/s

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Maximalfrequenz, Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<ACCELERATION>

DWORD: 0...100000 HZ/S

Beschleunigung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<DECELERATION>

DWORD: 0...100000 HZ/S

Verzögerung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, wird für die Verzögerung der für die Beschleunigung gültige Wert verwendet.

Beispiel: Fahre endlos mit 4000 Hz, Beschleunigung und Verzögerung von 10000 Hz/s**Befehl:**

FREQ 4000 10000

Antwort:

*FREQ 4000 10000

PRINTMARK

Mit diesem Befehl wird eine Druckmarkenfahrt gestartet.

Hierbei wird eine relative Positionierung gestartet, während dieser Positionierung wird auf das digitale Eingangssignal „Druckmarke“ (kann auf einen beliebigen Eingang gemappt werden) geachtet. Erkennt der Motor eine positive Flanke auf diesem Signal, so stoppt der Antrieb sofort mit der eingestellten Verzögerungsrampe, wird keine positive Flanke erkannt wird die Zielposition (Maxweg) angefahren und der Fehler Druckmarke gesetzt, Bit 6 (Druckmarkenfehler) im Fehlerwort der Statusausgabe.

Die Druckmarkenfahrt kann über die Parameter „Printmark settings“ eingestellt werden. Außerdem können einige Parameter direkt mit dem Kommando übergeben werden. Werden die Parameter nicht mit übergeben, so werden die Parametereinstellungen aus dem Antrieb verwendet. Weitere Details zur Druckmarkenfahrt siehe Handbuch.

Format

Befehl:

PRINTMARK (x)<MaxWeg> ((X)<Frequency>) ((X)<Acceleration>) ((X)<Deceleration>)

Antwort:

*PRINTMARK (x)<MaxWeg> ((X)<Frequency>) ((X)<Acceleration>) ((X)<Deceleration>)

Mögliche Kombinationen

PRINTMARK (X)<MaxWeg>

PRINTMARK (X)<MaxWeg> (X)<Frequency>

PRINTMARK (X)<MaxWeg> (X)<Frequency> (X)<Acceleration>

PRINTMARK (X)<MaxWeg> (X)<Frequency> (X)<Acceleration> (X)<Deceleration>

<MAXWEG>

DWORD: -...0...+

Distanz in Schritten
(1 Umdrehung = 400 Schritte)

<FREQUENCY>

DWORD: 0...20000 HZ

Zielgeschwindigkeit in Hz
400Hz = 1 Umdrehung/s

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Maximalfrequenz, Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<ACCELERATION>

DWORD: 0...100000 HZ/S

Beschleunigung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<DECELERATION>

DWORD: 0...100000 HZ/S

Verzögerung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, wird für die Verzögerung der für die Beschleunigung gültige Wert verwendet.

Beispiel: Druckmarkenfahrt Maxweg 1000 Schritte, mit 750 Hz, ACC/DEC 5000Hz/s

Befehl:
PRINTMARK 1000 750 5000

Antwort:
*PRINTMARK 1000 750 5000

MARK

Die Markierungsfahrt startet eine relative oder absolute Positionierung. Während dieser Positionierung kann das Signal "Markierer" schrittgenau für eine definierbare Länge gesetzt werden. Das Signal "Markierer" kann auf einen beliebigen Ausgang gemappt werden.

Es werden die Parameter direkt mit dem Kommando übergeben werden. Der Erste Parametersatz bis zum ersten Semikolon bezeichnet die Fahrparameter, nach dem ersten Semikolon kommen die Markierungspositionen, hier können bis zu 20 Positionen oder ein zyklischer definierbarer Abstand angegeben werden. Werden die Fahrparameter nicht mit übergeben, so werden die Parametereinstellungen aus dem Antrieb verwendet.

Format

Befehl:
MARK <Option> (X)<Zielposition> ((X)<Frequency>) ((X)<Acceleration>)
(X)<Deceleration>;(X)<AnzMarkPos> (X)<MarkPos[1]>
(X)<MarkLen[1]>...(X)<MarkPos[AnzMarkPos]> (X)<MarkLen[AnzMarkPos]>

Antwort:
MARK <Option> (X)<Zielposition> ((X)<Frequency>) ((X)<Acceleration>)
(X)<Deceleration>;(X)<AnzMarkPos> (X)<MarkPos[1]>
(X)<MarkLen[1]>...(X)<MarkPos[AnzMarkPos]> (X)<MarkLen[AnzMarkPos]>

Mögliche Kombinationen

MARK <Option> (X)<Zielposition>; (X)<AnzMarkPos> (X)<MarkPos[1]>
(X)<MarkLen[1]>...(X)<MarkPos[AnzMarkPos]> (X)<MarkLen[AnzMarkPos]>

MARK <Option> (X)<Zielposition> (X)<Frequency>; (X)<AnzMarkPos> (X)<MarkPos[1]>
(X)<MarkLen[1]>...(X)<MarkPos[AnzMarkPos]> (X)<MarkLen[AnzMarkPos]>

MARK <Option> (X)<Zielposition> (X)<Frequency> (X)<Acceleration>; (X)<AnzMarkPos>
(X)<MarkPos[1]> (X)<MarkLen[1]>...(X)<MarkPos[AnzMarkPos]> (X)<MarkLen[AnzMarkPos]>

MARK <Option> (X)<Zielposition> (X)<Frequency> (X)<Acceleration>
(X)<Deceleration>;(X)<AnzMarkPos> (X)<MarkPos[1]>
(X)<MarkLen[1]>...(X)<MarkPos[AnzMarkPos]> (X)<MarkLen[AnzMarkPos]>

<OPTION>

A	Absolute Bewegung auch die angegebene Markierpositionen werden Absolut, ausgewertet
R	Relative Bewegung

<ZIELPOSITION>

DWORD: -...0...+	Zielposition bzw. Distanz in Schritten (1 Umdrehung = 400 Schritte)
------------------	------------------------------------------------------------------------

<FREQUENCY>

DWORD: 0...20000 HZ	Zielgeschwindigkeit in Hz 400Hz = 1 Umdrehung/s
---------------------	----------------------------------------------------

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Maximalfrequenz, Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<ACCELERATION>

DWORD: 0...100000 HZ/S	Beschleunigung in Hz/s 400Hz/s = 1 U/s ²
------------------------	--------------------------------------------------------

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<DECELERATION>

DWORD: 0...100000 HZ/S	Verzögerung in Hz/s 400Hz/s = 1 U/s ²
------------------------	-----------------------------------------------------

Angabe optional: Sollte dieser Wert nicht gesetzt sein, wird für die Verzögerung der für die Beschleunigung gültige Wert verwendet.

<ANZMARKPOS>

BYTE: 0...20	Anzahl der Markierungen, wird hier 0 angegeben muss nur eine Folgeparametersatz angegeben werden, der dann zyklische abgearbeitet wird, d.h. alle x Schritte wird der Ausgang für die angegebene Länge aktiviert.
--------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<MARKPOS>

DATENTYP: INT32	Startpunkt Markierung
-----------------	-----------------------

<MARKLEN>

DATENTYP: UINT32	Länge der Markierung
------------------	----------------------

Beispiel Markierfahrt Absolut mit 2 Absolut-Markierpositionen:

Starten einer Markierfahrt auf die Zielposition Absolut 1000
 Frequenz 5000Hz; Beschleunigung 500Hz/s Verzögerung 300 Hz/s
 2 Markierungen ausgelöst an der Absolut-Position 50 für 10 Schritte und an der Absolut-Position 150 für 20 Schritte

Befehl:

MARK A 1000 5000 500 300 ; 2 50 10 150 20

Antwort:

*MARK A 1000 5000 500 300 ; 2 50 100 150 200

Beispiel Markierfahrt Relativ mit 2 Relativ-Markierpositionen:

Starten einer Markierfahrt mit einer Distanz von 1000 Schritten,
Frequenz von 5000Hz Beschleunigung 500Hz/s Verzögerung von 300 Hz/s
2 Markierungen ausgelöst nach 50 Schritten für 10 Schritte und nach 150 Schritten für 20 Schritte

Befehl:

MARK R 1000 5000 500 300 ; 2 50 10 150 20

Antwort:

*MARK R 1000 5000 500 300 ; 2 50 100 150 200

Beispiel Markierfahrt Absolut mit zyklischen Markierpositionen:

Starten einer Markierfahrt auf die Zielposition Absolut 1000
Frequenz 5000Hz; Beschleunigung 500Hz/s Verzögerung 300 Hz/s
Hierbei werden zyklisch alle 100 Schritte Markierungen ausgelöst, die eine Länge von 20 Schritten haben

Befehl:

MARK A 1000 5000 500 300 ; 0 100 20

Antwort:

*MARK A 1000 5000 500 300 ; 0 100 20

TRACE

Startet eine Messfahrt und speichert die Position schrittgenau bei jeder erkannten steigenden Flanke des Signals "TRACE" während des gesamten Weges. Die Anzahl der speicherbaren Positionen ist auf 20 begrenzt. Wenn nach dem Erreichen der Zielposition keine Traceposition gefunden wurde, so wird Bit 6 (Druckmarkenfehler) im Fehlerwort der Statusausgabe gesetzt.

Format

Befehl:

TRACE <Option> (X)<Zielposition> (X)<MinSigLen> ((X)<Frequency>) ((X)<Acceleration>)
 ((X)<Deceleration>)

Antwort:

*TRACE <Option> (X)<Zielposition> (X)<MinSigLen> ((X)<Frequency>) ((X)<Acceleration>)
 ((X)<Deceleration>)

Mögliche Kombinationen:

TRACE <Option> (X)<Zielposition> (X)<MinSigLen>
 TRACE <Option> (X)<Zielposition> (X)<MinSigLen> ((X)<Frequency>)
 TRACE <Option> (X)<Zielposition> (X)<MinSigLen> ((X)<Frequency>) ((X)<Acceleration>)
 TRACE <Option> (X)<Zielposition> (X)<MinSigLen> ((X)<Frequency>) ((X)<Acceleration>)
 ((X)<Deceleration>)

<OPTION>

A	Absolute Bewegung (speichert die Positionen als Absolut-Position ab)
R	Relative Bewegung (speichert die Positionen relativ zur Startposition ab)

<ZIELPOSITION>

DWORD: -...0...+	Zielposition bzw. Distanz in Schritten (1 Umdrehung = 400 Schritte)
------------------	------------------------------------------------------------------------

<MINSIGLEN>

DWORD: -...0...+	Mindestlänge Signal in Schritten (0 = nur positive Flanke)
------------------	---------------------------------------------------------------

<FREQUENCY>

DWORD: 0...20000 HZ	Zielgeschwindigkeit in Hz 400Hz = 1 Umdrehung/s
---------------------	----------------------------------------------------

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Maximalfrequenz, Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<ACCELERATION>

DWORD: 0...100000 HZ/S

Beschleunigung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, werden Beschleunigung und Verzögerung aus dem aktiven Dynamikprofil verwendet.

<DECELERATION>

DWORD: 0...100000 HZ/S

Verzögerung in Hz/s
400Hz/s = 1 U/s²

Angabe optional: Sollte dieser Wert nicht gesetzt sein, wird für die Verzögerung der für die Beschleunigung gültige Wert verwendet.

Beispiel: Trace-Fahrt

Trace-Fahrt Absolut mit Zielposition 1000, Signal-Mindestlänge von 5 Schritten, Geschwindigkeit 1000Hz, Beschleunigung und Verzögerung von 5000Hz/s

Befehl:

TRACE A 1000 5 1000 5000

Antwort:

*TRACE A 1000 5 1000 5000

Sobald das Signal Trace für min 5 Schritte high ist, wird die schrittgenaue Absolut-Position der steigenden Flanke abgespeichert. Es werden max. 20 steigende Flanken-Positionen abgespeichert, diese werden nach dem FIFO-Prinzip in einem reservierten Speicher abgelegt. Diese Werte können über den LIST Befehl ausgelesen werden.

Der Speicher wird bei jedem neuen Trace Befehl zuerst gelöscht, dann neu beschrieben.

Digitale Ein/Ausgänge bedienen

DIGOUT

Schreiben und lesen der Digitalen Ausgänge

Format

Befehl

Abfragen des gesamten Ausgangsbyte : DIGOUT (X)?

Antwort:

Rückgabe des gesamten Ausgangsbyte : *DIGOUT (X)<Wert>

Befehl

Abfragen eines bestimmten Bits(Ausgang): DIGOUT <Bit>?

Antwort:

Rückgabe des bestimmten Bits : *DIGOUT <Bit> <Wert>

Befehl

Setzen des gesamten Ausgangsbytes : DIGOUT (X)<Wert>

Antwort:

Rückgabe des gesamten Ausgangsbyte : *DIGOUT (X)<Wert>

Befehl

Setzen eines bestimmten Bits : DIGOUT (X)<Bit> (X)<Wert>

Antwort:

Rückgabe des bestimmten Bits : *DIGOUT <Bit> <Wert>

Beispiel: Setze Ausgang 0 auf 1

Befehl:

DIGOUT 0 1

Antwort:

*DIGOUT 0 1

DIGIN

Lesen der Digitalen Eingänge

Format

Befehl

Abfragen des gesamten Eingangsbyte : DIGIN (X)

Antwort:

Rückgabe des gesamten Eingangsbyte : *DIGIN (X)<Wert>

Befehl

Abfragen eines bestimmten Bits(Eingang): DIGIN <Bit>

Antwort:

Rückgabe des bestimmten Bits(Eingang) : *DIGIN <Bit> Wert>

Beispiel: Lese Eingang 2

Befehl:
DIGIN 2

Antwort:
*DIGIN 2 1

SIGIN

Lesen der Eingangssignale nach dem Mapping

Format

Befehl:
SIGIN <Signal>

Antwort:
*SIGIN <Signal><Wert>

<SIGNAL>

START	Start-Signal
PRINTMARK	Druckmarke
REFSW	Referenzschalter
LS+	Endschalter positiv
LS-	Endschalter negativ
ENABLE	Endstufe Enable
BIN0	Binär 0
BIN1	Binär 1
BIN2	Binär 2
BIN3	Binär 3
BIN4	Binär 4
RE0	Remote 0
RE1	Remote 1
RE2	Remote 2
RE3	Remote 3
RE4	Remote 4

<WERT>

0	False
1	True

Beispiel: Abfrage Eingang-Signal

Befehl:
SIGIN ENABLE

Antwort:
*SIGIN ENABLE 1

Werte und Parameter abfragen und verändern

PW

Setzen der Berechtigungsstufe

Format

Befehl:
PW (X)<Password>

Antwort:
*PW OK

<PASSWORD>

4 ZIFFERN

Passwort für die Berechtigungsstufe oder 0 zum abmelden

Beispiel: Gib die Berechtigungsstufe mit dem Passwort 1234 frei

Befehl:
PW 1234

Antwort:
*PW OK

POS

Abfragen oder Setzen der aktuellen Position in Schritten.

Format

Befehl:
POS (X)<Steps>

Antwort:
*POS (X)<Steps>

<STEPS>

DWORD: -...0...+

?

Position in Schritten
(1 Umdrehung = 400 Schritte)
Um die aktuelle Position abzufragen

Beispiel: Aktuelle Position abfragen

Befehl:
POS ?

Antwort:
*POS 500

STAT

Abfrage des Antriebsstatus.

Format

Befehl:

STAT

Antwort:

*STAT X<Status> X<Systemstatus> X<Error>

<STATUS>

WORD IN HEX

Statuswort des Antriebs
Bedeutung siehe Handbuch

<SYSTEMSTATUS>

WORD IN HEX

Systemstatuswort des Antriebs
Bedeutung siehe Handbuch

<ERROR>

WORD IN HEX

Error-Wort des Antriebs
Bedeutung siehe Handbuch

Beispiel: Abfragen des Status

Befehl:

STAT

Antwort:

*STAT X08 X00 X00

LIST

Ausgeben von Auflistungen

Format

Befehl:

LIST <Option> (X)

Antwort:

*LIST PARAMS (X)

<List of all Parameters formatted like *PARAM (X)<Index> (X)<Subindex> (X)<Value>>

*LIST DIAG <DiagCount> <LogCount>

<List of all diagnostical- (*DIAG) and logging- entries (*LOG)>

*LIST TRACES (X)

<List of all Trace positions formatted like *TRACES <TRACECount> (X)<Value>

<OPTION>

PARAMS (X)

Ausgabe aller für die Berechtigungsstufe
freigegebenen Parameter

DIAG	Ausgabe der Diagnosedaten
TRACES (X)	Ausgabe der Trace-Positionen, max. 20 abgespeicherte Positionen der letzten Trace Fahrt, die Positionen werden bei Start jeder Trace-Fahrt gelöscht. Sind keine Werte vorhanden, ist der TraceCount = 0

Beispiel: Gib alle Parameter für die aktive Berechtigungsstufe aus

Befehl:

LIST PARAMS X

Antwort:

*LIST PARAMS X 4
*PARAM x1 x1 x8
*PARAM x1 x2 x9
*PARAM x1 x3 xA
*PARAM x1 x4 xB

Beispiel: Gib alle abgespeicherten Positionen der Trace Funktion zurück

Befehl:

LIST TRACES

Antwort:

*LIST TRACES 3
*TRACES 0 -100
*TRACES 1 0
*TRACES 2 500

PARAM

Abfragen und ändern von Parametern

Format

Befehl:

PARAM (X)<Index> (X)<Subindex> (X)<Value>

Antwort:

*PARAM (X)<Index> (X)<Subindex> (X)<Value>

<INDEX>

DATENTYP: DWORD

Index des Parameters

<SUBINDEX>

DATENTYP: DWORD

Subindex des Parameters

<VALUE>

DATENTYP: DWORD

Wert auf den der Parameter gesetzt werden soll
oder ? zum abfragen des Werts

Beispiel: Setze den Wert des Parameters mit dem Index 10 und Subindex 2 auf 123

Befehl:

PARAM XA 2 X7B

Antwort:

*PARAM XA 2 X7B

PROFILE

Abfragen, setzen und bearbeiten von Dynamikprofilen.

Format

Befehl:

PROFILE (X)<Index> ((X)<Frequency> (X)<Acceleration> ((X)<Deceleration>))

Antwort:

*PROFILE (X)<Index> ((X)<Frequency> (X)<Acceleration> ((X)<Deceleration>))

Mögliche Kombinationen

PROFILE ?

PROFILE (X)<Index>

PROFILE (X)<Index> ?

PROFILE (X)<Index> (X)<Frequency> (X)<Acceleration>

PROFILE (X)<Index> (X)<Frequency> (X)<Acceleration> (X)<Deceleration>

<INDEX>

DATENTYP: BYTE ZULÄSSIGE WERTE: 1..3	Index des Dynamikprofils für das aktivieren oder ändern oder ? für das Abfragen des aktiven Profils
-------------------------------------------------------	-----------------------------------------------------------------------------------------------------

<FREQUENCY> DWORD: 0...20000 HZ	Zielgeschwindigkeit in Hz 400Hz = 1 Umdrehung/s
-------------------------------------------------	----------------------------------------------------

Angabe optional: Optional, wird nur benötigt, wenn das Dynamikprofil geändert werden soll.

Wenn ? geschickt wird, werden die Parameter des angegebenen Profils zurückgegeben.

<ACCELERATION> DWORD: 0...100000 HZ/S	Beschleunigung in Hz/s 400Hz/s = 1 U/s ²
-------------------------------------------------------	--------------------------------------------------------

Angabe optional: Optional, nur benötigt wenn das Dynamikprofil geändert werden soll

<DECELERATION> DWORD: 0...100000 HZ/S	Verzögerung in Hz/s 400Hz/s = 1 U/s ²
-------------------------------------------------------	-----------------------------------------------------

Angabe optional: Optional, nur benötigt, wenn das Dynamikprofil geändert wird, sollte dieser Wert nicht gesetzt sein, wird der gleiche Wert von <Acceleration> verwendet

Beispiel: Ändere Dynamikprofil 2 auf Maximalgeschwindigkeit 6000 Hz und Beschleunigung sowie Verzögerung 15000 Hz/s

Befehl: PROFILE 2 6000 10000
Antwort: *PROFILE 2 6000 10000

SCAN

Anmelden von Parametern für die Event-gesteuerte Ausgabe ihrer Werte.

Maximal können 10 Parameter gescannt werden.

Format

Befehl: SCAN <Option> (X)<Index> (X)<Subindex> (X)<Interval> (X)
Antwort: *SCAN <Option> (X)<Index> (X)<Subindex> (X)<Interval> (X)
Mögliche Kombinationen: SCAN START (X)<Index> (X)<Subindex> (X)<Interval> X SCAN START (X)<Index> (X)<Subindex> (X)<Interval> SCAN STOP (X)<Index> (X)<Subindex>

<OPTION>	
START	Starten der Eventgesteuerten Ausgabe für den angegebenen Parameter
STOP	Stoppen der Eventgesteuerten Ausgabe für den angegebenen Parameter
<SCAN-INDEX>	
1-10	SCAN-Index des Parameters
<PARAMETER-INDEX>	
	Index des Parameters
<PARAMETER-SUBINDEX>	
	SUB-Index des Parameters
<INTERVAL >	
WERT: 10...10000MS	Unterdrückungszeit, bis Parameter bei Änderung erneut gesendet wird Wichtig bei Parametern die sich sehr schnell und sehr oft ändern können. Zum Beispiel die Position oder der Status. Wenn der Wert zu klein ist kann der Ausgangspuffer überlaufen oder andere Ausgaben blockiert werden.
X	
	Wenn X angegeben ist, wird der Parameterwert als Hex ausgegeben

Beispiel: Melde Parameter mit dem Index 10 und dem Subindex 1 mit einem Intervall von 500 ms an

Befehl:
 SCAN START X10 1 500 X

Antwort:
 *SCAN START X10 1 500 X

Antwort bei Parameteränderung:
 *PARAM X10 X1 X01F4

ENC

Ausgabe des Encoder-Rohwertes

Format

Befehl: ENC	
Antwort: *ENC POS:<Pos> EW:<EW> EP:<EP> EO:<EO>	
<POS>	
DATENTYP: DWORD	Aktuelle Position ohne Abweichungsschwelle
<EW>	
DATENTYP: DWORD	Position innerhalb einer Umdrehung (Winkel) entspricht Schritte / Umdrehung * 10
<EP>	
DATENTYP: DWORD	Position aus dem Encoder (nur Singleturn und Multiturn)
<EO>	
DATENTYP: DWORD	Offset zur Korrektur des Absolutwertgebers (nur Singleturn und Multiturn)

Beispiel: Abfrage Encoder Rohwert

Befehl: ENC	
Antwort: *ENC POS:100 EW:1000 EP:25 EO:0	

UMOT

Lesen der Motorspannung

Format

Befehl: UMOT (X)	
Antwort: *UMOT (X)<Value>	
<VALUE>	
DATENTYP: BYTE	Wert des Motorspannung in [V]

Beispiel: Auslesen Motorspannung

Befehl: UMOT	
------------------------	--

Antwort:
*UMOT 26

U24

Lesen der Steuerspannung

Format

Befehl:
U24 (X)

Antwort:
*U24 (X)<Value>

<VALUE>

DATENTYP: BYTE

Wert des Steuerspannung in [V]

Beispiel: Auslesen Steuerspannung

Befehl:
U24

Antwort:
*U24 26

TETA

Lesen der Temperatur an der Endstufe

Format

Befehl:
TETA (X)

Antwort:
*TETA (X)<Value>

<VALUE>

DATENTYP: BYTE

Wert des Steuerspannung in [°C]

Beispiel: Auslesen Temperatur an der Endstufe

Befehl:
TETA

Antwort:
*TETA 26

IDRIVE

Lesen und Schreiben des Fahrstroms

Format

Befehl:

IDRIVE (X)<Value>

Antwort:

*IDRIVE (X)<Value>

<VALUE>

DATENTYP: BYTE

Zulässige Werte: 0..100
Wert in % zum Schreiben
oder ? für das Lesen des Fahrstroms

Beispiel: Auslesen des Fahrstroms

Befehl:

IDRIVE ?

Antwort:

*IDRIVE 90

IHOLD

Lesen und Schreiben des Haltestroms

Format

Befehl:

IHOLD (X)<Value>

Antwort:

*IHOLD (X)<Value>

<VALUE>

DATENTYP: BYTE

Zulässige Werte: 0..100
Wert in % zum Schreiben
oder ? für das Lesen des Haltestroms

Beispiel: Auslesen des Haltestroms

Befehl:

IHOLD ?

Antwort:

*IHOLD 20

IREF

Lesen und Schreiben des Referenzstroms

Format

Befehl:

IREF (X)<Value>

Antwort:

*IREF (X)<Value>

<VALUE>

DATENTYP: BYTE

Zulässige Werte: 0..100
Wert in % zum Schreiben
oder ? für das Lesen des Referenzstrom

Beispiel: Auslesen des Referenzstroms

Befehl:

IREF ?

Antwort:

*IREF 20

SAVE

Speichern der Parameter im EEPROM

Format

Befehl:

SAVE <Option>

Antwort:

*SAVE <Option>

<OPTION>

ANGABE IN WELCHEM BEREICH DIE
PARAMETER GESICHERT WERDEN SOLLEN

KEINE ANGABE

Speichern aller Parameter im EEprom

FACTORY

Nur möglich mit der Berechtigungsstufe
"Hersteller"
Speichern der aktuellen Parameter als
Herstellerparameter im Flash

DEFAULT

Nur möglich mit der Berechtigungsstufe "Erweitert"
Speichern der aktuellen Parameter als
Standardparameter im EEPROM

Beispiel: Speichern im EEPROM

Befehl:

SAVE

Antwort:

*SAVE

Systembefehle

RESTART

Neustart eines Bereichs

Format

Befehl:

RESTART <Option>

Antwort:

*RESTART <Option>

<OPTION>

SOFTWARE
HARDWARE

Software-Restart
Hardware- Restart

Beispiel: Neustart

Befehl:

RESTART SOFTWARE

Antwort:

*RESTART SOFTWARE

RESET

Zurücksetzen eines Bereichs

Format

Befehl:

RESET <Option>

Antwort:

*RESET <Option>

<OPTION>

ERRORS
DIAG

Löschen aller anstehender Fehler
Zurücksetzen des Diagnosespeichers

Beispiel: Fehler löschen

Befehl:

RESET ERRORS

Antwort:
*RESET ERRORS

REMOTE

Aktivieren, Deaktivieren und Abfragen des Kommando-Interpreters.
Ist dieser Modus aktiv, werden Befehle Automatisch im Kommando-Interpreter abgearbeitet.

Format

Befehl:
REMOTE <Option>

Antwort:
*REMOTE <Option>

<OPTION>

?	Abfragen des aktuellen Status
ON	Einschalten
OFF	Ausschalten

Beispiel: Gibt den Status des Remote-Modus aus

Befehl:
REMOTE ?

Antwort:
*REMOTE ON

SELFTEST

Testen der internen Systemkomponenten

Format

Befehl:
SELFTEST

Antwort:
*SELFTEST <ComponentCount>
<List of all tests with format *SELFTEST <Component> <Result>

Beispiel: Starte Self-Test

Befehl:
SELFTEST

Antwort:
*SELFTEST 4
*SELFTEST IIC OK
*SELFTEST SPI OK

```
*SELFTEST ENC OK  
*SELFTEST EEPROM_1 OK
```

Heartbeat

Startet die Heartbeat-Funktion, bzw. beendet sie.

Format

Befehl:
HEARTBEAT <Value>

Antwort:
*HEARTBEAT

<VALUE>

100-65000

Intervall in ms, innerhalb dieser Zeit muss von der übergeordneten Steuerung irgendeine eine Anfrage oder ein Befehl gesendet werden. Egal welche.

0

Schaltet Heartbeat-Funktion aus.

Die Heartbeat Funktion ist nur bei sich drehendem Motor relevant, wenn bei sich drehendem Motor eine Anfrage in der Überwachungszeit ausbleibt, so wird der Motor mit der normalen STOP-Rampe angehalten. Außerdem wird im Fehlerwort das Bit 9 (HEARTBEAT Fehler) gesetzt.

Beispiel: Heartbeat aktivieren mit einer Intervallzeit von 500ms

Befehl:
HEARTBEAT 500

Antwort:
*HEARTBEAT 500