

# Softwarehandbuch

## Schunk Motion Protokoll

### Motion Control Schunk V2.11

Superior Clamping and Gripping



## Impressum

### Urheberrecht:

Diese Anleitung ist urheberrechtlich geschützt. Urheber ist die SCHUNK GmbH & Co. KG. Alle Rechte vorbehalten.

### Technische Änderungen:

Änderungen im Sinne technischer Verbesserungen sind uns vorbehalten.

**Dokumentnummer:** 0389051

**Auflage:** 02.00 | 17.01.2022 | de

Sehr geehrte Kundin,

sehr geehrter Kunde,

vielen Dank, dass Sie unseren Produkten und unserem Familienunternehmen als führendem Technologieausrüster für Roboter und Produktionsmaschinen vertrauen.

Unser Team steht Ihnen bei Fragen rund um dieses Produkt und weiteren Lösungen jederzeit zur Verfügung. Fragen Sie uns und fordern Sie uns heraus. Wir lösen Ihre Aufgabe!

Mit freundlichen Grüßen

Ihr SCHUNK-Team

Customer Management

Tel. +49-7133-103-2503

Fax +49-7133-103-2189

cmg@de.schunk.com



**Betriebsanleitung bitte vollständig lesen und produktnah aufbewahren.**

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemein .....</b>	<b>7</b>
1.1	Darstellung der Warnhinweise .....	7
1.2	Elektrischer Anschluss .....	8
1.3	Anzeigenelemente .....	9
1.4	Werkseinstellungen .....	9
1.5	Booten .....	9
1.6	SMP Protokoll für CAN/Profibus/USB .....	10
1.6.1	Daten Format .....	10
1.6.2	Datenrahmen .....	12
1.6.3	Besonderheiten bei USB Kommunikation .....	13
1.6.4	Besonderheiten bei CAN .....	13
1.6.5	Besonderheiten bei Profibus .....	14
1.6.6	Fragmentierung .....	16
1.7	Benutzerverwaltung .....	18
1.7.1	Nutzer .....	18
1.7.2	Profi .....	18
1.7.3	Advanced .....	18
1.7.4	Root .....	18
1.7.5	Schunk .....	19
1.7.6	Disabled .....	19
1.8	Pseudoabsolutwertgeber .....	19
1.8.1	Voraussetzungen .....	19
1.8.2	Funktion .....	19
1.9	Stillstandskommütierung .....	21
1.9.1	Voraussetzungen .....	21
1.9.2	Funktion .....	21
<b>2</b>	<b>Einheitensystem .....</b>	<b>22</b>
<b>3</b>	<b>Kommandos .....</b>	<b>23</b>
3.1	Bewegung .....	24
3.1.1	CMD REFERENCE .....	24
3.1.2	MOVE POS .....	25
3.1.3	MOVE POS REL .....	27
3.1.4	MOVE VEL .....	29
3.1.5	SET TARGET VEL .....	30
3.1.6	SET TARGET ACC .....	30
3.1.7	SET TARGET JERK .....	31
3.1.8	SET TARGET CUR .....	32
3.1.9	SET TARGET POS .....	33
3.1.10	SET TARGET POS REL .....	33
3.1.11	CMD STOP .....	34
3.1.12	CMD FAST STOP .....	35
3.2	Spontanmeldungen .....	36
3.2.1	CMD INFO .....	36

3.2.2	CMD MOVE BLOCKED .....	37
3.2.3	CMD POS REACHED .....	38
3.2.4	CMD ERROR .....	38
3.2.5	GET STATE.....	39
3.2.6	CMD TOGGLE IMPULSE MESSAGE .....	42
3.3	Einstellungen .....	42
3.3.1	CALIB CURRENT .....	42
3.3.2	SET CONFIG EXT .....	43
3.3.3	GET CONFIG EXT .....	44
3.4	Sonstige .....	45
3.4.1	CMD REBOOT .....	45
3.4.2	CHANGE USER.....	45
3.4.3	CMD GET DIO (0xE9).....	45
3.5	Fragmentierung.....	47
3.5.1	FRAG ACK (0x87).....	47
3.5.2	FRAG START (0x84).....	47
3.5.3	FRAG MIDDLE (0x85) .....	47
3.5.4	FRAG END (0x86) .....	48
<b>4</b>	<b>Info- und Fehlermeldungen.....</b>	<b>49</b>
4.1	Fehler Behandlung.....	49
4.1.1	CMD ERROR .....	49
4.1.2	CMD WARNING .....	50
4.1.3	CMD INFO .....	50
4.1.4	CMD ACK .....	50
4.1.5	Detaillierte Fehlerinformationen.....	50
4.2	Info- und Fehlercodes.....	51
4.2.1	INFO BOOT.....	51
4.2.2	INFO NO RIGHTS .....	51
4.2.3	INFO UNKNOWN COMMAND.....	51
4.2.4	INFO FAILED.....	51
4.2.5	NOT REFERENCED.....	51
4.2.6	INFO SEARCH SINE VECTOR .....	51
4.2.7	INFO NO ERROR.....	52
4.2.8	INFO COMMUNICATION ERROR.....	52
4.2.9	INFO TIMEOUT.....	52
4.2.10	INFO WRONG DATA TYPE .....	52
4.2.11	INFO RESTART.....	52
4.2.12	INFO CHECKSUM.....	52
4.2.13	INFO VALUE LIMIT MAX.....	52
4.2.14	INFO VALUE LIMIT MIN.....	53
4.2.15	INFO MESSAGE LENGTH .....	53
4.2.16	INFO WRONG PARAMETER.....	53
4.2.17	INFO UNKNOWN PARAMETER.....	53
4.2.18	ERROR FILE NOT FOUND.....	53
4.2.19	ERROR FILE IS CORRUPT .....	53

4.2.20	ERROR FILE TYPE WRONG.....	53
4.2.21	ERROR FILE SYSTEM WRONG.....	53
4.2.22	ERROR FILE READ .....	53
4.2.23	ERROR FILE IS NOT CREATED .....	54
4.2.24	ERROR FILE WRITE .....	54
4.2.25	ERROR REBOOT.....	54
4.2.26	ERROR MOTOR PHASE .....	54
4.2.27	ERROR WRONG RAMP TYPE .....	54
4.2.28	ERROR WRONG DIRECTION .....	54
4.2.29	ERROR CONFIG MEMORY .....	54
4.2.30	ERROR SOFT LOW .....	54
4.2.31	ERROR SOFT HIGH .....	55
4.2.32	ERROR SERVICE.....	55
4.2.33	ERROR FAST STOP .....	55
4.2.34	ERROR TOW .....	55
4.2.35	ERROR VPC3.....	55
4.2.36	ERROR FRAGMENTATION .....	55
4.2.37	ERROR COMMUTATION.....	56
4.2.38	ERROR I2T .....	56
4.2.39	ERROR CURRENT.....	56
4.2.40	ERROR TOO FAST .....	56
4.2.41	ERROR POS SYSTEM.....	56
4.2.42	ERROR RESOLVER CHECK FAILED.....	56
4.2.43	ERROR MATH.....	56
4.2.44	ERROR CALIB CURRENT .....	57
4.2.45	ERROR INITIALIZE.....	57
4.2.46	ERROR INTERNAL.....	57
4.2.47	ERROR CONNECTION TEMP LOW .....	57
4.2.48	ERROR CONNECTION TEMP HIGH .....	57
4.2.49	ERROR MOTOR TEMP LOW .....	58
4.2.50	ERROR MOTOR TEMP HIGH.....	58
4.2.51	ERROR TEMP LOW .....	58
4.2.52	ERROR TEMP HIGH .....	58
4.2.53	ERROR LOGIC LOW .....	58
4.2.54	ERROR LOGIC HIGH.....	58
4.2.55	ERROR MOTOR VOLTAGE LOW .....	59
4.2.56	ERROR MOTOR VOLTAGE HIGH.....	59
4.2.57	ERROR CABLE BREAK .....	59
4.2.58	ERROR LIFE SIGN.....	60
4.2.59	ERROR CUSTOM DEFINED.....	60
4.2.60	ERROR OVERSHOOT.....	60
4.2.61	ERROR HARDWARE VERSION .....	60
4.2.62	ERROR SOFTWARE VERSION.....	60
<b>5</b>	<b>Konfigurationsparameter.....</b>	<b>61</b>
5.1	Allgemein.....	61
5.2	Parametrierung mit Hilfe eines USB-Sticks.....	62

5.2.1	Parameter lesen .....	62
5.2.2	Parameter schreiben .....	62
5.3	Parameter .....	62
5.3.1	Zyklisch .....	62
5.3.2	Permanent .....	66
5.3.3	Asynchron .....	103
<b>6</b>	<b>Bemerkungen .....</b>	<b>112</b>
6.1	Schnell-Test .....	112
6.2	Module .....	112
6.3	Protokoll .....	112
6.4	Profibus .....	112
<b>7</b>	<b>Anhang .....</b>	<b>113</b>
7.1	Auflistung der Enumerationen .....	113
7.2	Info- und Fehlercodes .....	116
7.3	Firmware Update .....	118

# 1 Allgemein

## 1.1 Darstellung der Warnhinweise

Zur Verdeutlichung von Gefahren werden in den Warnhinweisen folgende Signalworte und Symbole verwendet.



### **⚠ GEFAHR**

#### **Gefahren für Personen!**

Nichtbeachtung führt sicher zu irreversiblen Verletzungen bis hin zum Tod.



### **⚠ WARNUNG**

#### **Gefahren für Personen!**

Nichtbeachtung kann zu irreversiblen Verletzungen bis hin zum Tod führen.



### **⚠ VORSICHT**

#### **Gefahren für Personen!**

Nichtbeachtung kann zu leichten Verletzungen führen.

### **ACHTUNG**

#### **Sachschaden!**

Informationen zur Vermeidung von Sachschäden.

## 1.2 Elektrischer Anschluss

Das Modul verfügt über getrennte Eingangsklemmen für die Motorspannung und die Logikspannung (24V DC). Es wird empfohlen, beide Spannungen getrennt anzuschließen. Damit ist gewährleistet, dass bei einer Überlastung der Motorspannung die Logik weiter arbeitet, und der Zustand des Moduls immer bekannt ist. Bei Modulen mit einer Motorspannung > 24V DC ist eine getrennte Spannungsversorgung ohnehin unumgänglich, da die Logikspannung immer im Bereich zwischen 18V DC und 32V DC liegen muss.

### **ACHTUNG**

#### **Dauerhafte Beschädigung der Elektronik möglich!**

- Bei getrennter Versorgung ist darauf zu achten, dass ein Potentialausgleich zwischen den beiden Versorgungsspannungen vorgenommen wird (Massen zusammenführen)
- Es darf nur der Pluspol abgeschaltet werden, der Motor-GND muss immer verbunden sein.

---

### **HINWEIS**

Ist die Spannungsversorgung von Logik und Motor getrennt, kann von der Steuerung die Versorgungsspannung des Motors mittels Relais abgeschaltet werden, das Modul ist aber weiter am Bussystem aktiv.

---

Bei der Versorgung des Leistungsteils (Motor) ist ein Netzteil einzusetzen, welches in der Lage ist ausreichend Strom für das jeweilige Modul zu liefern. Bei der Verkabelung ist auf ausreichende Kabelquerschnitte zu achten.

Der Spannungsabfall am Kabel lässt sich durch folgende Formel berechnen:

$$\Delta U = (2 \times l \times I) / (\chi \times A) \text{ mit:}$$

I: Stromaufnahme der Last

l: Länge der Leitung

$\chi$ : elektrische Leitfähigkeit

Cu:  $\chi = 56 \text{ ((m) / (\Omega \times \text{mm}^2))}$

Al:  $\chi = 35 \text{ ((m) / (\Omega \times \text{mm}^2))}$

A: Leiterquerschnitt

### 1.3 Anzeigenelemente

Siehe Anleitung für das jeweilige Modul.

### 1.4 Werkseinstellungen

Siehe Anleitung für das jeweilige Modul.

### 1.5 Booten

Auf dem Modul sind Default-Werte für Bewegung als Standardwerte gesetzt. Dadurch kann das Modul direkt in Betrieb genommen werden, ohne das vorher Parameter gesetzt werden müssen. Folgende Default-Werte gelten nach dem Neustart:

- „Zielgeschwindigkeit“  
in [%] vom maximal Wert -> 10%, ▶ [5.3.2.2.7 \[ 73\]](#)
- „Zielbeschleunigung“  
in [%] vom maximal Wert -> 10%, ▶ [5.3.2.2.8 \[ 73\]](#)
- „Zielruck“  
in [%] vom maximal Wert -> 50%, ▶ [5.3.2.2.9 \[ 73\]](#)
- „Zielstrom“  
Nennstrom, ▶ [5.3.2.2.6 \[ 72\]](#)
- Spontanmeldungen aktiviert.
- Nutzer wird auf „Nutzer“ gestellt, ▶ [1.7 \[ 18\]](#)

## 1.6 SMP Protokoll für CAN/Profibus/USB

### 1.6.1 Daten Format

Daten werden von den Modulen im Intel Format (little Endian) versendet und beim Empfangen in eben diesem Format interpretiert.

#### Gleitkommazahlen

Die Norm IEEE 754 (Gleitkommazahlen) wurde in den frühen 1980er entwickelt, um unter anderem eine konsistente Gleitkommazahlenrepräsentation über verschiedene Rechnerarchitekturen zu erreichen. Dieser Standard wird eingehalten, wenn die Parameter als Gleitkommazahlen zum Modul gesendet bzw. von diesem zur Steuerung übermittelt werden. Eine Gleitkommazahl wird hierbei mit 32-Bit dargestellt.

Vorzeichen Bit	Exponent	Mantisse (normiert)
1 Bit (bit 32)	8 Bit (bit 23.. bit 30)	23 Bit (bit 1.. bit 22)
<i>s</i>	<i>e</i>	<i>f</i>

Da die Mantisse immer auf „1“ normiert ist, werden nur die Nachkommastellen gespeichert. Die führende „1“ wird nicht mit abgespeichert. Ein Float-Wert kann wie folgt berechnet werden.

$$(-1)^s \times 2^{e-127} \times (1.f)_{bin}$$

Beispiele:

	Vorzeichen	Exponent	Mantisse
	1 Bit	8 Bit	23 Bit
7/4	0	01111111	11000000000000000000000
-34.432175	1	10000100	00010011011101010001100
-959818	1	10010010	11010100101010010100000
+0	0	00000000	00000000000000000000000
-0	1	00000000	00000000000000000000000
$2^{-126} = 1.175 \times 10^{-38}$			
kleinste positive Zahl	0	00000001	00000000000000000000000
$(2-2^{-23}) 2^{127} = 3.403 \times 10^{38}$			
größte positive Zahl	0	11111110	11111111111111111111111
unendlich	0	11111111	11111111111111111111111
NaN	0	11111111	nicht alle „0“ oder „1“
$2^{-23} = 1.192 \times 10^{-7}$			
kleinste auflösende Zahl	0	01101000	00000000000000000000000
$2^{-128}$	0	00000000	01000000000000000000000

## Zweierkomplement

Das Zweierkomplement ist eine Möglichkeit um negative Zahlen im Binärsystem darzustellen. Bei dem Modul wird das Zweierkomplement für die Darstellung negativer Ganzzahlen benötigt.

Positive Zahlen werden im Zweierkomplement mit einer führenden 0 (Vorzeichenbit) versehen und ansonsten nicht verändert. Negative Zahlen werden mit einer führenden 1 als Vorzeichenbit versehen und wie folgt kodiert: Sämtliche Ziffern der entsprechenden positiven Zahl werden negiert. Zum Ergebnis wird 1 addiert. Beispielhafte Umwandlung der negativen Dezimalzahl -4(dez) ins Zweierkomplement:

1. Vorzeichen ignorieren und ins Binärsystem umrechnen:  $4_{\text{dez}} = 00000100_{\text{bin}} = 0x\ 04_{\text{hex}}$
2. Invertieren, da negativ:  $11111011_{\text{bin}} = 0x\ FB_{\text{hex}}$
3. Eins addieren, da negativ:  $11111011_{\text{bin}} + 00000001_{\text{bin}} = 11111100_{\text{bin}} = 0x\ FC_{\text{hex}} = -4_{\text{dez}}$

Etwas mathematischer:

Ist  $x$  eine negative Zahl, so errechnet sich  $x$  im Zweierkomplement ( $x_2$ ) mit  $n$  Stellen wie folgt:

$$x_2 = 2^n - |x|$$

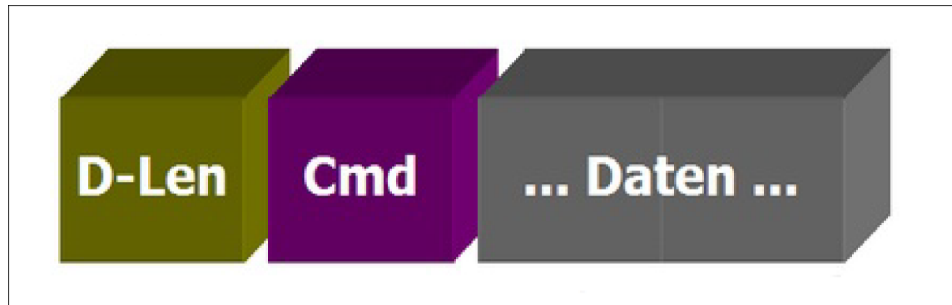
Dementsprechend gilt auch:

$$x_2 + |x| = 2^n$$

### 1.6.2 Datenrahmen

Der Datenrahmen des Motion-Protokolls umfasst immer folgende Elemente:

- D-Len (1 Byte)
- KOMMANDO CODE (1 Byte)



*Datenrahmen*

D-Len (Data Length) gibt die Anzahl der nachfolgenden Nutzdaten einschließlich des Kommando Bytes an. D-Len ist ein Byte breit, deshalb können mit einer Motion-Protokoll Nachricht maximal 254 Daten Bytes übertragen werden.

Im Anschluss an das D-Len Byte folgt immer der aus einem Byte bestehende Kommando-Code. Dem Kommando-Code folgen, falls notwendig, die jeweilig benötigten Parameter. Falls erforderlich, kann der Kommando-Code noch durch „Sub-Kommando-Codes“ erweitert werden.

Alle abgesandten Befehle werden sofort vom Modul mit einer Antwort (Acknowledge) bestätigt. Diese Antwort benutzt ebenfalls den oben beschriebenen Datenrahmen. (D-Len, Kommando-Code, evtl. Parameter). Wurde die Anfrage erfolgreich verarbeitet, besitzt D-Len immer einen Wert ungleich „0x02“. Ist die Anfrage fehlerhaft gewesen, besitzt D-Len genau den Wert „0x02“.

In den beiden folgenden Bytes ist die Ursache der *fehlerhaften Anfrage* beschrieben.

Die Module haben darüber hinaus die Eigenschaft, ohne dass eine Anfrage gestellt worden ist, von sich aus Meldungen abzusetzen. Der Datenrahmen wird bei solchen „Spontanmeldungen“ ebenfalls eingehalten. Bei folgenden Ereignissen wird eine „Spontanmeldung“ ausgelöst:

- Es ist ein schwerwiegender Fehler aufgetreten.
- Eine Bewegung wurde korrekt beendet.
- Regelmäßige *Statusmeldung* falls aktiviert, ▶ [3.2 \[136\]](#).

### 1.6.3 Besonderheiten bei USB Kommunikation

Die USB Kommunikation ist nur für Parametrierung und Inbetriebnahme konzipiert. Für den Zugriff über USB wird empfohlen das Motion Tool Schunk (siehe Softwareanleitung "Motion Tool Schunk") zu verwenden.

### 1.6.4 Besonderheiten bei CAN

CAN als nachrichtenorientiertes Bussystem, benötigt neben dem Datenrahmen einen eindeutigen der Nachricht zugeordneten Identifier. Die Module unterstützen den Standard 11-Bit Identifier. Die unteren 8-Bit werden hierbei für die eindeutige Modul ID verwendet. Es können also maximal 255 Module adressiert werden.

Die freien 3-Bit sind wie folgt kodiert:

- 0x03 - Fehlernachricht eines Moduls
- 0x05 - Nachricht vom Master zu einem Modul
- 0x07 - Antwort vom Modul

Alle anderen Zustände werden nicht benutzt.

- Eine Nachricht zum Modul besitzt folgenden Identifier 0x5XX.  
(XX Moduladresse in hex Darstellung)
- Eine Nachricht vom Modul besitzt den Identifier 0x7XX.  
(XX Moduladresse in hex Darstellung)
- Im Fehlerfall werden die Nachrichten vom Modul an den Master mit dem Identifier 0x3XX versendet.  
(XX Moduladresse in hex Darstellung)

In einer CAN Nachricht können maximal 8 Byte verschickt werden. Es ist unter Umständen notwendig einen längeren Datenrahmen (D-Len > 7) in mehrere CAN Nachrichten zu verpacken. Dies geschieht mittels des Fragmentierungsprotokolls

▶ [1.6.6 \[16\]](#)

---

#### HINWEIS

**Eine Fragmentierung ist normalerweise nicht notwendig, da alle für den Betrieb der Module notwendigen Befehle in eine CAN Nachricht verpackt werden können.**

---

### 1.6.5 Besonderheiten bei Profibus

Beim Profibus DPV0 gibt es folgende Besonderheiten:

Ein *alternatives Profibusprotokoll* steht zur Auswahl

Das alternative Profibusprotokoll SDP kann direkt über die SPS bei der Konfigurierung des Moduls aktiviert werden.

Bei dem hier beschriebenen SMP-Protokoll ist die maximale Länge der auf einmal vom Master zum Modul zu übertragenden Daten ist auf 8 Byte begrenzt. Hiermit kann ein Modul komplett bedient werden. (Es werden maximal 7 Byte für eine Nachricht vom Master zum Modul benötigt)



Datenrahmen Profibus

Die maximale Länge der vom Modul zum Master gesendeten Daten (Antwort) ist auf 16 Byte begrenzt (GSD-Datei). Will man größere Datenmengen versenden / empfangen kann eine *Fragmentierung* ▶ 1.6.6 [□ 16] notwendig werden. In 16 Byte findet die längste im Normalbetrieb vorkommende Nachricht vom Modul zum Master Platz, es bleiben 2 Byte übrig. In diesen 2 Byte, welche immer am Ende der Profibusnachricht stehen (Byte 14, Byte 15) ist,

1. der aktuelle *Status* ▶ 3.2.5 [□ 39] des Moduls (Byte 14) **und**
2. ein so genannter Kommando Zähler (MsgCount) (Byte 15) enthalten.

Werden die Bytes 10-13 nicht benötigt, kann hier die aktuelle Position im jeweiligen *Einheitensystem* ▶ 2 [□ 22] ausgelesen werden.

#### HINWEIS

**Bytes 10-13 werden nur in der Antwort auf ein ▶ 3.2.5 [□ 39] verwendet, wenn alle Daten (Position, Geschwindigkeit und Strom) angefordert werden.**

**Bei fragmentierten Nachrichten werden die Bytes 10-15 für Daten verwendet.**

---

**HINWEIS**

Nur die oberen 8 Bit des Status-Wortes werden geschrieben. Der Fehlercode entfällt. Hierzu gibt es zum einen die erweiterte Diagnose unter Profibus, zum anderen wird im Fehlerfall der Fehlercode ▶ 4.1 [📄 49] in den Ausgangsdaten dargestellt. Wird eine Nachricht vom Master zum Modul versandt, so wird bei Profibus zusätzlich zur Antwort der MsgCount um 1 erhöht. Hiermit wird sichergestellt, dass trotz möglicher Spontanmeldungen jede Anfrage bestätigt wird.

---

**HINWEIS**

**Eine Spontanmeldung ▶ 3.2 [📄 36] erhöht den MsgCount nicht!**

---

Will man zum Beispiel auf eine Position fahren, an der sich das Modul aktuell befindet, meldet das Modul „Befehl verstanden“ und sofort im nächsten Profibus-Zyklus „Position erreicht“. Da eine mit dem Profibus verbundene Steuerung unter Umständen nicht in jedem Profibus-Zyklus die Daten abfragt, könnte die Acknowledge (Antwort) auf den Verfahrbefehl verloren gehen. Durch den MsgCount ist gewährleistet, dass eine Bestätigung auf die Anfrage eingegangen ist.

Über das *Status-Byte* ▶ 3.2.5 [📄 39] (Byte 14) erhält man immer die aktuelle Information über den Zustand des vorhandenen Moduls.

---

**HINWEIS**

**Das letzte Bit des MsgCount kann als Toggle-Bit ausgewertet werden. (Modul zum Master) Bei der Übertragung von Daten vom Master zum Modul kann das nicht benutzte Byte 8 als Toggle-Byte verwendet werden, bzw. Bit 63 als Toggle-Bit.**

---

Gruppen werden vollständig durch den in Profibus implementierten SYNC, FREEZE Mechanismus unterstützt.

Sollte eine konsistente Datenübertragung nicht möglich sein, so gibt es folgende Möglichkeiten das Modul zu betreiben:

- SYNC, UNSYNC Mechanismus einsetzen.
- D-Len auf „0“ setzen. Alle Daten auffüllen und sobald alle Daten vorhanden sind D-Len setzen.

### 1.6.6 Fragmentierung

#### HINWEIS

Eine Fragmentierung von Nachrichten ist für den Normalbetrieb nicht erforderlich!

Sollte eine Fragmentierung von Meldungen erforderlich sein, geschieht dies wie folgt:



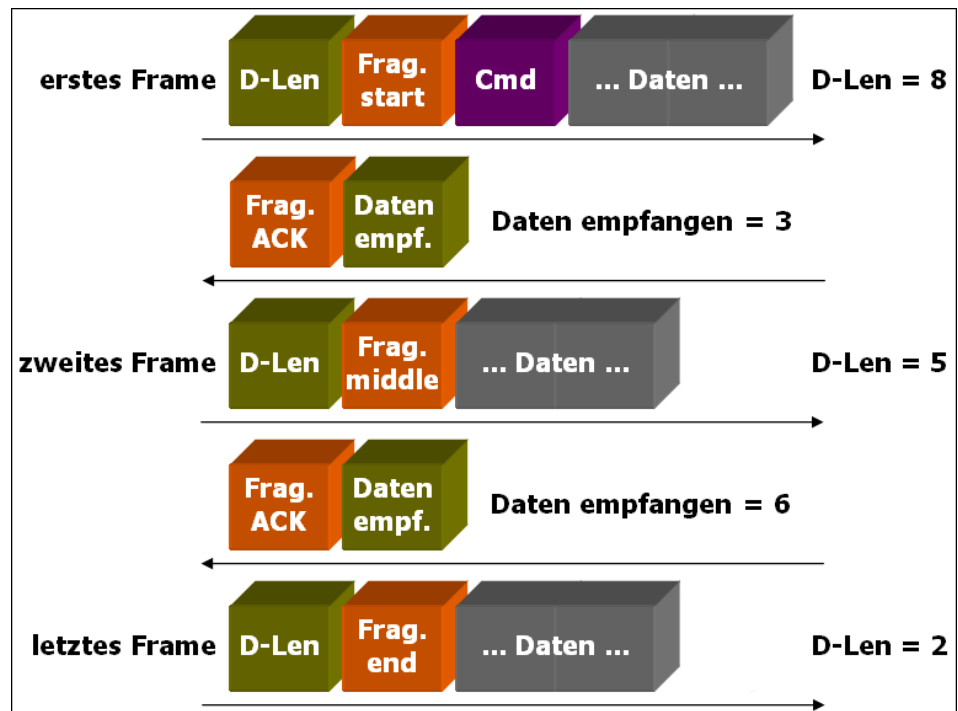
Fragmentierung

Zu Beginn jeder Nachricht wird die Länge der noch folgenden Nutzdaten gesendet. Anschließend wird eine Fragmentkennung gesendet. Diese Fragmentkennung wird **nicht** im Längenbyte (D-Len) erfasst.

- FragStart -> *erstes Fragment*
- FragMiddle -> *ein mittleres Fragment*
- FragEnd -> *letztes Fragment*

Diese einzelnen Fragmente können somit wieder zu einem kompletten *Datenrahmen* zusammengebaut werden, welcher anschließend interpretiert werden kann.

## Besonderheit Profibus



Fragmentierung Profibus

Da beim Profibus ständig ein „Token“ unterwegs ist, aus dem sich die jeweiligen Teilnehmer die für sie gültigen Daten herausfischen bzw. die für den Master notwendigen einschreiben, muss jedes erhaltene Fragment mit „FRAG ACK“ und dem D-Len Byte des erhaltenen Fragments bestätigt werden. Wird eine fragmentierte Nachricht an den Master gesendet, muss dieser jedes einzelne Fragment mit „FRAG ACK“ und dem D-Len Byte des erhaltenen Fragments bestätigen, damit das Modul das nächste Fragment verschicken kann. Sendet der Master eine fragmentierte Nachricht an das Modul, so muss er mit dem Senden des folgenden Fragmentes solange warten bis das Modul den Erhalt des Fragmentes bestätigt hat („FRAG ACK“ und dem D-Len Byte des erhaltenen Fragments). Das letzte Fragment muss nicht bestätigt werden.

## 1.7 Benutzerverwaltung

Das Modul ist mit einer Benutzerverwaltung ausgerüstet, um bestimmte Aktionen besonders zu schützen. Die Benutzerrechte können über „CHANGE USER“ ▶ 3.4.2 [☐ 45] bzw. mit dem Parameter "User" (0x7DDA) ▶ 5.3.3.33 [☐ 111] umgeschaltet werden.

### 1.7.1 Nutzer

Ist der Standard-Nutzer, welcher nach dem Einschalten des Moduls immer aktiviert wird. Dieser kann das Modul vollständig bedienen. Parametrieren ist nur für die wichtigsten *Parameter* ▶ 5 [☐ 61] gestattet.

### 1.7.2 Profi

Ist der Profi-Nutzer, welcher den vollen Funktionsumfang des „Nutzer“ hat und zusätzlich weitere Parameter schreiben kann. Bei falscher Parametrierung kann es zu unvorhergesehenem Verhalten des Moduls kommen. Das Modul kann jedoch nicht zerstört werden.

Das Standard-Kennwort für die Profi-Rechte lautet „Schunk“.

### 1.7.3 Advanced

Ist der Advanced-Nutzer, welcher den vollen Funktionsumfang des „Profi“ hat und zusätzlich weitere Parameter verstellen kann.

#### **ACHTUNG**

**Eine Fehlbedienung oder eine falsche Parametrierung kann zur Zerstörung der Elektronik oder des Motors führen.**

---

### 1.7.4 Root

Ist der Root-Nutzer, welcher vollen Zugriff auf das Modul hat. Es sind alle Parameter einstellbar und weitere Funktionen für Testzwecke zugänglich.

#### **ACHTUNG**

**Eine Fehlbedienung oder eine falsche Parametrierung kann zur Zerstörung der Elektronik oder des Motors führen.**

---

### 1.7.5 Schunk

Parameter mit Schunk-Nutzer können nur im Werk bzw. durch qualifiziertes Servicepersonal geändert werden.

### 1.7.6 Disabled

Parameter mit "Disabled-Nutzer" können nicht geändert werden.

## 1.8 Pseudoabsolutwertgeber

### 1.8.1 Voraussetzungen

Die Module unterstützen die Pseudoabsolutwertgeberfunktion, wenn folgende Voraussetzungen erfüllt sind:

- Bremse
- FRAM (*Hardware Version* ungerade)
- *Positionsmesssystem* ▶ 5.3.2.5.2 [□ 90] Resolver
- oder Positionsmesssystem Encoder mit Indexspur und
  - Motortyp DC ▶ 5.3.2.2.3 [□ 71]
  - oder Motortyp BLDC
  - oder Motortyp PMSM

### 1.8.2 Funktion

Beim Einfall der Bremse wird die aktuelle Position in einen nichtflüchtigen Speicher abgespeichert. Wird die Logikspannung abgeschaltet, so wird versucht mit der verbleibenden Restenergie die aktuelle Position abzuspeichern.

#### Resolver

Beim erneuten Einschalten des Moduls wird nun die zuvor gespeicherte Position mit einem Kontrollwert verglichen. Ist diese Kontrolle erfolgreich, wird die abgespeicherte Position mit der aktuellen Position des Resolvers verglichen. Sind auch diese Positionen gleich, so ist das Modul nicht neu zu referenzieren.

#### HINWEIS

**Wird der Resolver im unbestromten Zustand um genau eine Umdrehung gedreht, so wird beim Wiedereinschalten die angezeigte Position fehlerhaft sein.**

#### Encoder mit Indexspur

Beim erneuten Einschalten des Moduls wird die zuvor gespeicherte Position mit einem Kontrollwert verglichen. Ist diese Kontrolle erfolgreich, wird die abgespeicherte Position zur aktuellen Position. Gleichzeitig wird der Abstand zum nächsten Indexpuls berechnet.

Beim ersten folgenden Verfahrbefehl wird beim Erreichen des Indexpuls der errechnete Abstand mit dem gemessenen Abstand verglichen. Stimmen beide Werte überein, so gilt das Modul als

referenziert. Der Indexpuls muss nach dem Senden des ersten Verfahrbefehls darüber hinaus innerhalb einer gewissen Zeit erreicht werden.

Tritt während der Bewegung zum Indexpuls ein *Fehler* ▶ 4.1 [📄 49] auf, so wird die Referenzierung gelöscht.

---

#### **HINWEIS**

**Nach einer erfolgreichen Referenzfahrt muss der Indexpuls mindestens einmal überfahren werden, um die Funktion zu aktivieren.**

---

---

#### **HINWEIS**

**Wird der Encoder im unbestromten Zustand bewegt, so ist es möglich, dass das Modul nach dem Wiedereinschalten mit der falschen Position eine Bewegung zum nächsten Indexpuls ausführt (max. eine Motorumdrehung).**

---

---

#### **HINWEIS**

**Wird der Encoder im unbestromten Zustand um genau eine Umdrehung gedreht, so wird beim Wiedereinschalten die angezeigte Position fehlerhaft sein.**

---

## 1.9 Stillstandskommütierung

### 1.9.1 Voraussetzungen

- Motortyp DC ▶ 5.3.2.2.3 [171]
- oder Motortyp BLDC
- oder Motortyp PMSM und
  - *Positionsmesssystem* ▶ 5.3.2.5.2 [190] „Encoder mit Indexspur“ und vorhandenen Hallgebern
  - oder Positionsmesssystem Resolver

### ACHTUNG

**Bewegungsrichtung bei Blockkommütierung und Sinuskommütierung müssen übereinstimmen. Bei unterschiedlichen Drehrichtungen sind Phasen zu tauschen und die Kommutiertabelle ▶ 5.3.2.2.14 [175] anzupassen.**

### 1.9.2 Funktion

Sind alle Voraussetzungen gegeben, wird das Modul versuchen die Stillstandskommütierung durchzuführen. Bei Modulen mit Absolutwertmesssystemen kann direkt nach dem Einschalten die Sinus-Kommütierung aktiviert werden, da die Lage des „Sinus-Zeigers“ bekannt ist.

Bei Modulen mit Encodermesssystem ist die Lage des „Sinus-Zeigers“ erst nach Erreichen des Indexpulses bekannt. Daher wird das Modul mit Blockkommütierung zum ersten Indexpuls gefahren (bei der Fahrt mit Blockkommütierung kann die Kraft etwas geringer sein) und anschließend auf Sinus-Kommütierung umgestellt.

Die Lage des „Sinus-Zeigers“ zum Indexpuls wird über den Parameter *Positionierung Offset* Positionierung eingestellt bzw. „nachjustiert“. Wird dieser Wert zu „0“ gesetzt wird beim nächsten Bewegungsbefehl durch Bestromen der Motorphasen ein „Sinus-Zeigers“ gesucht, welcher *abgespeichert* Positionierung wird. Die *Referenzierung* ▶ 1.8 [119] wird hierbei gelöscht.

### HINWEIS

**Zur Zeigersuche sollte das Modul in alle Richtungen frei beweglich sein. Modul wird ruckartig bis zu zwei Motorumdrehungen bewegt. Eine Kommunikation mit dem Modul ist in dieser Zeit nicht möglich.**

## 2 Einheitensystem

Bei Linear- und Greifer-Produkten ist folgendes Einheitensystem festgelegt:

- Position [mm]
- Geschwindigkeit [mm/s]
- Beschleunigung [mm/s<sup>2</sup>]
- Ruck [mm/s<sup>3</sup>]
- Stromwerte [A]
- Zeiten [s]

Bei rotatorischen Einheiten gilt:

- Position [Grad]
- Geschwindigkeit [Grad/s]
- Beschleunigung [Grad/s<sup>2</sup>]
- Ruck [Grad/s<sup>3</sup>]
- Stromwerte [A]
- Zeiten [s]

---

## 3 Kommandos

---

### HINWEIS

**Dieses Kapitel gilt nicht für das Schunk Drive Protocol (SDP).**

---

Alle Beispiele führen nur den Datenrahmen auf. Die Besonderheiten der verschiedenen Bussysteme sind ab Kapitel ▶ 1.6.3 [13] beschrieben. Einige ausgesuchte Beispiele für die verschiedenen Bussysteme sind im *Anhang* ▶ 7 [113] aufgeführt.

---

### HINWEIS

**Bei allen aufgeführten Beispielen wird davon ausgegangen, dass ein Greifer angeschlossen ist welcher in [mm] verfahren wird.**

---

Bei allen Beispielen sind nur die *notwendigen Parameter* aufgeführt, die *optionalen Parameter* werden nicht aufgeführt. In den Beispielen steht „M“ für Master und „S“ für Slave (Modul).

## 3.1 Bewegung

### 3.1.1 CMD REFERENCE

KOMMANDO CODE: 0x92

BESCHREIBUNG: Es wird eine Referenzfahrt ausgeführt. Die Art der Referenzierung wird einmalig in den *Konfigurationsparameter* ▶ 5.3.2.4.1 [82] festgelegt.

PARAMETER (Master -> Slave): Keine

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich. Modul führt Kommando aus.

BEISPIEL:

	D-Len	Cmd	Param
M->S	0x01	0x92	
S->M	0x03	0x92	0x4F 0x4B

SONSTIGES: Spontanantwort möglich. Je nach *Referenzierart* ▶ 5.3.2.4.1 [82] kann es zu „*CMD MOVE BLOCKED*“ ▶ 3.2.2 [37] oder „*CMD POS REACHED*“ ▶ 3.2.3 [38] kommen. Dies hängt vom Flag „*MOVE ZERO AFTER REFERENCING*“ ▶ 5.3.2.4.4 [88] ab. Ein gesetztes Flag löst nach der Referenzierung eine Positionsfahrt aus => „*CMD POS REACHED*“ ▶ 3.2.3 [38]. Während der Referenzierung werden für die Bewegungen die *eingestellten Parameter* ▶ 5.3.2.4.5 [88] übernommen. Nach der Referenzierung werden die zuletzt eingestellten Werte (vor Referenzfahrt) wieder hergestellt. Bitte auch Besonderheiten des jeweiligen *Positionsmesssystems* ▶ 5.3.2.5.2 [90] beachten.

---

#### HINWEIS

**Eine einmal erfolgreich durchgeführte Referenzierung bleibt nach Abschalten der Betriebsspannungen unter bestimmten Voraussetzungen erhalten. Siehe hierzu „Pseudoabsolutwertgeber“ ▶ 1.8 [19]**

---

#### HINWEIS

**Vor einer Referenzfahrt sind bei einem Greifer alle Werkstücke zu entnehmen.**

---

### 3.1.2 MOVE POS

KOMMANDO CODE: 0xB0

BESCHREIBUNG: Bewegt das Modul an eine festgelegte Position. Die Position wird im konfigurierten *Einheitensystem* ▶ 2 [□ 22] vorgegeben. Für die Positionsfahrt wird das *konfigurierte Verfahrprofil* ▶ 5.3.2.5.11 [□ 93] zugrundegelegt.

PARAMETER (Master -> Slave)\*:

- *Position* ▶ 3.1.9 [□ 33] (optional), im konfigurierten Einheitensystem.
- *Geschwindigkeit* ▶ 3.1.5 [□ 30] (optional), welche für die Positionsfahrt verwendet wird. Bei Verfahrprofil „No Ramp“ ▶ 5.3.2.5.11 [□ 93] nicht relevant.
- *Beschleunigung* ▶ 3.1.6 [□ 30] (optional), welche für die Positionsfahrt verwendet wird. Bei Verfahrprofil „No Ramp“ ▶ 5.3.2.5.11 [□ 93] nicht relevant.
- *Strom* ▶ 3.1.8 [□ 32] (optional), welcher bei der Positionsfahrt nicht überschritten werden darf. Sollte die Reglerstruktur „CURRENT SPEED“ ▶ 5.3.2.3.15 [□ 80] aktiv sein, muss dieser Wert übermittelt werden (Ruck wird benötigt). Der Wert muss mindestens „0“ sein, ansonsten wird „INFO WRONG PARAMETER“ ▶ 4.2.16 [□ 53] erzeugt.
- *Ruck* (optional), welcher für die Positionsfahrt verwendet wird. Sollte das Verfahrprofil ungleich „Ruckbegrenzt“ ▶ 5.3.2.5.11 [□ 93] sein, kann dieser Wert nicht mit übermittelt werden („INFO WRONG PARAMETER“ ▶ 4 [□ 49]).

ANTWORT (Slave->Master): Falls möglich, wird die Zeit zurückgegeben die das Modul voraussichtlich für die Bewegung braucht. Wenn keine Berechnung der Zeit möglich sein sollte, wird die Anfrage beim Erfolg mit „OK“ (0x4F4B) bestätigt und das Modul führt die Bewegung aus.

#### HINWEIS

**Sollte das Verfahrprofil auf „Sprung“ konfiguriert sein, wird das ruckbegrenzte Verfahrprofil verwendet. Bei Verfahrprofil „Sprung“ würde der Antrieb zu heftige Bewegungen ausführen.**

BEISPIEL 1:

	D-Len	Cmd	Param	
M->S	0x05	0xB0	0x00 0x00 0x20 0x41	Fahre auf Position 10.0 [mm]
S->M	0x05	0xB0	0xCD 0xCC 0x04 0x41	Werde Position in 8.3 [s] voraussichtlich erreichen

BEISPIEL 2:

	D-Len	Cmd	Param	
M->S	0x01	0xB0		Fahre auf zuletzt eingestellte Position
S->M	0x05	0xB0	0xCD 0xCC 0x04 0x41	Werde Position in 8.3 [s] voraussichtlich erreichen

SONSTIGES: Spontanantwort wird beim Erreichen der Position „*CMD POS REACHED*“ ▶ 3.2.3 [☐ 38] oder bei Blockierung der Positionsfahrt „*MOVE BLOCKED*“ ▶ 3.2.2 [☐ 37] erfolgen.

Alle Parameter sind in der angegebenen Reihenfolge zu übermitteln. Soll nur der Strom vorgegeben werden, so muss zwingend die Position, Geschwindigkeit und Beschleunigung mit angegeben werden. Nachfolgende Parameter müssen nicht mit übermittelt werden. Alle Parameter bleiben bis zum Neustart oder einer Änderung dieser Parameter erhalten.

### 3.1.3 MOVE POS REL

KOMMANDO CODE: 0xB8

BESCHREIBUNG: Bewegt das Modul um eine festgelegte Strecke. Die Positionsänderung wird im konfigurierten *Einheitensystem* ▶ 2 [□ 22] vorgegeben. Für die Positionsfahrt wird das *konfigurierte Verfahrprofil* ▶ 5.3.2.5.11 [□ 93] zugrundegelegt.

PARAMETER (Master -> Slave)\*:

- *Positionsänderung* ▶ 3.1.10 [□ 33] (optional), im konfigurierten Einheitensystem
- *Geschwindigkeit* ▶ 3.1.5 [□ 30] (optional), welche für die Positionsfahrt verwendet wird. Bei Verfahrprofil „No Ramp“ ▶ 5.3.2.5.11 [□ 93] nicht relevant.
- *Beschleunigung* ▶ 3.1.6 [□ 30] (optional), welche für die Positionsfahrt verwendet wird. Bei Verfahrprofil „No Ramp“ ▶ 5.3.2.5.11 [□ 93] nicht relevant.
- *Strom* ▶ 3.1.8 [□ 32] (optional), welcher bei der Positionsfahrt nicht überschritten werden darf. Sollte die Reglerstruktur „CURRENT SPEED“ ▶ 5.3.2.3.15 [□ 80] aktiv sein, muss dieser Wert übermittelt werden (da Ruck benötigt wird). Der Wert muss mindestens „0“ sein, ansonsten wird „INFO WRONG PARAMETER“ ▶ 4.2.16 [□ 53] erzeugt.
- *Ruck* (optional), welcher für die Positionsfahrt verwendet wird. Sollte das Verfahrprofil ungleich „Ruckbegrenzt“ ▶ 5.3.2.5.11 [□ 93] sein kann dieser Wert nicht mit übermittelt werden („INFO WRONG PARAMETER“ ▶ 4 [□ 49]).

#### HINWEIS

**Sollte das Verfahrprofil auf „Sprung“ konfiguriert sein, wird das ruckbegrenzte Verfahrprofil verwendet. Bei Verfahrprofil „Sprung“ würde der Antrieb zu heftige Bewegungen ausführen.**

ANTWORT (Slave->Master): Falls möglich, wird die Zeit zurückgegeben die das Modul für die Bewegung voraussichtlich braucht. Wenn keine Berechnung der Zeit möglich sein sollte wird die Anfrage beim Erfolg mit „OK“ (0x4F4B) bestätigt und das Modul führt die Bewegung aus.

BEISPIEL 1:

	D-Len	Cmd	Param	
M->S	0x05	0xB8	0x00 0x00 0x20 0x41	Fahre um 10.0[mm] weiter
S->M	0x05	0xB8	0xCD 0xCC 0x04 0x41	Werde hierzu 8.3[sek.] voraussichtlich benötigen

BEISPIEL 2:

	D-Len	Cmd	Param	
M->S	0x01	0xB8		Fahre um die zuletzt eingestellte Distanz weiter
S->M	0x05	0xB8	0xCD 0xCC 0x04 0x41	Werde hierzu 8.3[sek.] voraussichtlich benötigen

**Sonstiges:** Spontanantwort wird beim Erreichen der Position "CMD POS REACHED" ▶ 3.2.3 [☐ 38] oder bei Blockierung der Positionsfahrt "MOVE BLOCKED" ▶ 3.2.2 [☐ 37] erfolgen. Alle Parameter sind in der angegebenen Reihenfolge zu übermitteln. Soll nur der Strom vorgegeben werden, so muss zwingend die Positionsänderung, Geschwindigkeit und Beschleunigung mit angegeben werden. Nachfolgende Parameter müssen nicht mit übermittelt werden. Alle Parameter bleiben bis zum Neustart oder einer Änderung dieser Parameter erhalten.

### 3.1.4 MOVE VEL

KOMMANDO CODE: 0xB5

BESCHREIBUNG: Eine Geschwindigkeitsfahrt wird ausgeführt.

PARAMETER (Master -> Slave):

- *Geschwindigkeit* ▶ 3.1.5 [□ 30] im konfigurierten *Einheitensystem* ▶ 2 [□ 22].
- *Strom* ▶ 3.1.8 [□ 32] (optional), welcher bei der Geschwindigkeitsfahrt nicht überschritten werden darf. Erzeugt bei Controller Struktur „CURRENT SPEED“ ▶ 5.3.2.3.15 [□ 80] „INFO WRONG PARAMETER“ ▶ 4 [□ 49].

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich.  
Modul führt Kommando aus.

BEISPIEL 1:

	D-Len	Cmd	Param	
M->S	0x05	0xB5	0x9A 0x99 0x31 0x41	Führe Geschwindigkeitsfahrt mit 11.1[(mm)/(s)] aus
S->M	0x03	0xB5	0x4F 0x4B	

BEISPIEL 2:

	D-Len	Cmd	Param	
M->S	0x05	0xB5		Führe Geschwindigkeitsfahrt mit der zuletzt eingestellten Geschwindigkeit aus.
S->M	0x03	0xB5	0x4F 0x4B	

SONSTIGES: Sonstiges: Spontanmeldung „CMD MOVE BLOCKED“ ▶ 3.2.2 [□ 37] kann erfolgen, wenn das Modul während der Fahrt blockiert wurde.

### 3.1.5 SET TARGET VEL

KOMMANDO CODE: 0xA0

BESCHREIBUNG: Der Parameter Geschwindigkeit wird hiermit gesetzt.

PARAMETER (Master -> Slave):

- *Geschwindigkeit* im vorgegeben *Einheitensystem*  
▶ 2 [□ 22].

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x05	0xA0	0x33 0x33 0x43 0x41	Setze Geschwindigkeit auf 12.2 [mm/s]
S->M	0x03	0xA0	0x4F 0x4B	

SONSTIGES: Dieser Wert bleibt nach einmaligem erfolgreichen Schreiben erhalten, bis das Modul neu gestartet, oder dieser Wert erneut geändert wird. Der Wert wird erst für den nächsten Bewegungsbefehl übernommen und ist nicht sofort aktiv.

### 3.1.6 SET TARGET ACC

KOMMANDO CODE: 0xA1

BESCHREIBUNG: Der Parameter Beschleunigung wird hiermit gesetzt.

PARAMETER (Master -> Slave):

- *Beschleunigung* im vorgegeben *Einheitensystem*  
▶ 2 [□ 22].

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x05	0xA1	0x00 0x00 0xF0 0xA2	Setze Beschleunigung auf 120 [mm/s <sup>2</sup> ]
S->M	0x03	0xA1	0x4F 0x4B	

SONSTIGES: Dieser Wert bleibt nach einmaligem erfolgreichen Schreiben erhalten, bis das Modul neu gestartet, oder dieser Wert erneut geändert wird. Der Wert wird erst für den nächsten Bewegungsbefehl übernommen und ist nicht sofort aktiv.

### 3.1.7 SET TARGET JERK

KOMMANDO CODE: 0xA2

BESCHREIBUNG: Der Parameter Ruck wird hiermit gesetzt.

PARAMETER (Master -> Slave): *Ruck* im vorgegeben *Einheitensystem* ▶ 2 [□ 22].

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich

BEISPIEL:

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x05	0xA2	0x00 0x00 0x7A 0x44	Setze Ruck auf 1000 [mm/ s <sup>3</sup> ]
S->M	0x03	0xA2	0x4F 0x4B	

SONSTIGES: Dieser Wert bleibt nach einmaligem erfolgreichen Schreiben erhalten, bis das Modul neu gestartet, oder dieser Wert erneut geändert wird. Der Wert wird erst für den nächsten Bewegungsbefehl übernommen und ist nicht sofort aktiv.

### 3.1.8 SET TARGET CUR

KOMMANDO CODE: 0xA3

BESCHREIBUNG: Der Parameter Strom wird hiermit gesetzt.

PARAMETER (Master -> Slave): *Strom* im vorgegeben  
*Einheitensystem* ▶ 2 [📄 22]

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x05	0xA3	0xCD 0xCC 0x2C 0x40	Setze Strom auf 2.7 [A]
S->M	0x03	0xA3	0x4F 0x4B	

SONSTIGES: Dieser Wert bleibt nach einmaligem erfolgreichem Setzen erhalten bis das Modul neu gestartet oder dieser Wert erneut geändert wird. Der Wert wird sofort übernommen und ist auch sofort aktiv.

#### HINWEIS

Hiermit können während der Bewegung die Stromsollvorgaben geändert werden. Es ist somit bei Greifern möglich einen bereits gegriffenen Gegenstand „nachzugreifen“ (Greifkraft erhöhen bzw. verringern).

#### HINWEIS

Wird bei Bewegungen nur dann berücksichtigt, wenn die Reglerstruktur dies zulässt. Reglerstruktur ▶ 5.3.2.3.15 [📄 80] „CURRENT SPEED“ lässt keine unterlagerte Stromregelung zu.

### 3.1.9 SET TARGET POS

KOMMANDO CODE: 0xA6

BESCHREIBUNG: Der Parameter „Position“ für den nächsten "MOVE POS" ▶ 3.1.2 [□ 25] Befehl wird hiermit gesetzt.

PARAMETER (Master -> Slave):

- *Position* im vorgegebenen *Einheitensystem* ▶ 2 [□ 22]

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x05	0xA6	0x00 0x00 0xc8 0x43	Setze Position auf 400,0 [°]
S->M	0x03	0xA6	0x4F 0x4B	„OK“

SONSTIGES: Dieser Wert bleibt nach einmaligem erfolgreichem Schreiben erhalten, bis das Modul neu gestartet, oder dieser Wert erneut geändert wird. Der Wert wird erst für den nächsten Bewegungsbefehl übernommen und ist nicht sofort aktiv.

### 3.1.10 SET TARGET POS REL

KOMMANDO CODE: 0xA7

BESCHREIBUNG: Der Parameter „Relativposition“ für das nächste "MOVE POS REL" ▶ 3.1.3 [□ 27] Kommando wird hiermit gesetzt.

PARAMETER (Master -> Slave):

- *relativ Position* im vorgegebenen *Einheitensystem* ▶ 2 [□ 22].

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x05	0xA7	0x00 0x00 0xc8 0x43	Setze Relativposition auf 400,0 [°]
S->M	0x03	0xA7	0x4F 0x4B	„OK“

SONSTIGES: Dieser Wert bleibt nach einmaligem erfolgreichem Schreiben erhalten, bis das Modul neu gestartet, oder dieser Wert erneut geändert wird. Der Wert wird erst für den nächsten Bewegungsbefehl übernommen und ist nicht sofort aktiv.

### 3.1.11 CMD STOP

KOMMANDO CODE: 0x91

BESCHREIBUNG: Das Modul wird abgebremst und in der aktuellen Position gehalten. Bei Modulen mit entsprechend *konfigurierter Haltebremse* fällt diese ein, ansonsten wird das Modul aktiv geregelt.

PARAMETER (Master -> Slave): Keine

ANTWORT (Slave->Master): „OK“ (0x4F4B) wenn erfolgreich

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x01	0x91		
S->M	0x03	0x91	0x4F 0x4B	

SONSTIGES: Kann der Befehl nicht ordnungsgemäß ausgeführt werden, so wird dieser mit einer *Fehlermeldung* ▶ 4 [49] beantwortet. z.B. Modul soll in aktiver Regelung bleiben, kann dies aber nicht, da ein schwerwiegender Fehler anliegt.

#### HINWEIS

Bei Modulen ohne Bremse bzw. entsprechend konfigurierter Haltebremse ▶ 5.3.2.7.3 [96] wird der maximal zulässige Strom für die Regelung auf den Nominalstrom ▶ 5.3.2.2.6 [72] begrenzt um eine Überhitzung des Motors zu vermeiden. Das Modul kann daher evtl. „durchsacken“.

### 3.1.12 CMD FAST STOP

KOMMANDO CODE: 0x90

BESCHREIBUNG: Ist eine Bremse vorhanden und entsprechend *konfiguriert* ▶ 5.3.2.7.3 [□ 96], fällt diese sofort ein, und die Motorphasen werden kurzgeschlossen. Motor wird „stromlos geschaltet“.

PARAMETER (Master -> Slave): Keine

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x01	0x90		
S->M	0x02	0x88	0xD9	Schnellstop ausgeführt

ANTWORT (Slave->Master): Fehlermeldung „*ERROR FAST STOP*“ ▶ 4 [□ 49] wird ausgelöst.

SONSTIGES: Kann nur durch „*CMD ACK*“ ▶ 4.1 [□ 49] wieder zurückgesetzt werden.



#### ⚠️ WARNUNG

##### Verletzungsgefahr!

Bei Modulen ohne Bremse kann das Modul „durchsacken“, da der Motor beim Schnellstop stromlos geschaltet wird!

#### HINWEIS

Diese Art des Anhaltens führt zu einem starken mechanischen Verschleiß der Bremse.

## 3.2 Spontanmeldungen

Bei bestimmten Ereignissen kann sich das Modul selbständig melden. Es wird eine sog. „Spontanmeldung“ ausgelöst. Diese Meldungen werden über den Standard *Datenrahmen* gesendet. (D-Len, CmdCode, Parameter). Es ist möglich Spontanmeldungen zu *deaktivieren* ▶ 3.2.6 [42].

### HINWEIS

**Bei Profibus wird der MsgCount bei solchen Meldungen nicht erhöht, da von der Steuerung keine Daten angefordert wurden.**

### 3.2.1 CMD INFO

KOMMANDO CODE: 0x8A

BEISPIEL:

	D-Len	Cmd	Param	
S->M	0x02	0x8A	0x10	„INFO TIMEOUT“ ▶ 4.2.9 [52]

BESCHREIBUNG: Das Modul sendet eine Informationsnachricht.

BEMERKUNG: Bei behobenen Fehlern und einem Neustart des Moduls werden ebenfalls „Info Nachrichten“ versendet.

(▶ 4.2.1 [51] ▶ 4.2.7 [52])

### 3.2.2 CMD MOVE BLOCKED

KOMMANDO CODE: 0x93

BESCHREIBUNG: Der aktuelle Bewegungsbefehl wurde unterbrochen. Der Motor war kurzzeitig blockiert, bzw. ist immer noch blockiert.

ANTWORT (Slave->Master): Aktuelle *Position* im eingestellten *Einheitensystem* ▶ 2 [□ 22]

BEISPIEL:

	D-Len	Cmd	Param	
S->M	0x05	0x93	0xA4 0x70 0x9D 0x3F	Stehe an Position 1.23 [mm]

SONSTIGES: Der Motor gilt als blockiert, wenn alle der folgende Voraussetzungen erfüllt sind:

- Der Motor dreht sich mit einer Geschwindigkeit unterhalb der *Bewegungsschwelle* ▶ 5.3.2.5.9 [□ 93]
- der Zielstrom erreicht ist (+/-15%)
- die in *waitMoveBlocked* ▶ 5.3.2.5.13 [□ 94] parametrisierte Zeit ist abgelaufen.

Siehe auch *Flag „Bewegung blockiert“*

#### HINWEIS

**Bei Erkennung der Blockade wird bei Sinuskommutierung der eingestellte Strom um Wurzel(2) verringert. (Formfaktor Effektivwert zu Spitzenwert bei Sinusförmigen Strömen.)**

#### HINWEIS

**Eine absolut sichere Erkennung, ob ein Gegenstand gegriffen wurde, ist hiermit nicht möglich.**

### 3.2.3 CMD POS REACHED

KOMMANDO CODE: 0x94

BESCHREIBUNG: Eine Positionsfahrt hat die Zielposition erreicht.

ANTWORT (Slave->Master): *Position* (aktuell) in eingestellten *Einheitensystem* ▶ 2 [□ 22]

BEISPIEL:

	D-Len	Cmd	Param	
S->M	0x05	0x94	0xCD 0xCC 0x2C 0x40	Habe Position 2.7 [mm] erreicht.

SONSTIGES: Das Modul steht. Bei vorhandener Bremse fällt diese je nach *Konfiguration* ▶ 5.3.2.7.3 [□ 96] ein.

### 3.2.4 CMD ERROR

KOMMANDO CODE: 0x88

BESCHREIBUNG:

Ein schwerwiegender Fehler ist aufgetreten, welcher einen Benutzereingriff notwendig macht. Diese Fehler müssen mit „*CMD ACK*“ ▶ 4.1.4 [□ 50] quittiert werden. Das Modul ist nicht betriebsbereit. Der Motor ist stromlos geschaltet und die Bremse eingefallen. Diese Fehlermeldungen werden regelmäßig alle 15 Sekunden vom Modul zur Steuerung gesendet, bis der Fehler quittiert ist.

- CAN: die ersten drei Bit des Identifiers ergeben sich zu „0x3“
- Profibus: eine erweiterte Diagnose wird generiert

ANTWORT (Slave->Master): *Fehlercode* ▶ 4 [□ 49]

BEISPIEL:

	D-Len	Cmd	Param	
S->M	0x02	0x88	0xDE	Fehler „ <i>ERROR CURRENT</i> “ ist aufgetreten.



#### ⚠️ WARNUNG

**Verletzungsgefahr! Bei Modulen ohne Bremse kann das Modul „durchsacken“, da der Motor bei einem schwerwiegenden Fehler stromlos geschaltet wird.**

### 3.2.5 GET STATE

KOMMANDO CODE: 0x95

BESCHREIBUNG: Liefert den Status des Moduls, sowie einige weitere Informationen, wenn gewünscht. Das Modul kann diesen Status selbständig in regelmäßigen Abständen versenden.

PARAMETER (Master -> Slave):

- Keine  
Modul liefert einmalig die Daten. Hiermit kann das vorher eingestellte zyklische Senden von Daten abgeschaltet werden.
- *Zeit* (4 Byte)  
Das Modul sendet selbständig im eingegebenen Zeitintervall (im jeweiligen *Einheitensystem* ) seinen Status.
- *Zeit* (4 Byte) *Daten* (1 Byte)  
Das Modul sendet selbständig im eingegebenen Zeitintervall (im jeweiligen *Einheitensystem* 18) seinen Status.  
Mit dem Code „Modus“ wird parametrierung, welche Daten zusätzlich zum Status mitgeliefert werden sollen:  
Bit 1 (0x01): Position  
Bit 2 (0x02): Geschwindigkeit  
Bit 3 (0x04): Strom

ANTWORT (Slave->Master): Optionale Daten (den Code „Modus“ beachten), im Anschluss daran der Status (2 Byte), welcher wie folgt aufgebaut ist:

Referenziert	Bit 1	0x01
Bewegung	Bit 2	0x02
Programmablauf	Bit 3	0x04
Warnung	Bit 4	0x08
Fehler	Bit 5	0x10
Bremse	Bit 6	0x20
Bewegung blockiert	Bit 7	0x40
Position erreicht	Bit 8	0x80

- Bit 1: Das Modul ist referenziert.
- Bit 2: Das Modul bewegt sich.
- Bit 3: Das Modul befindet sich im Programm Modus (Ein internes Ablaufprogramm ist aktiv).
- Bit 4: Es liegt eine *Warnung* vor.
- Bit 5: Es liegt ein *Fehler* vor.
- Bit 6: Die Bremse ist eingefallen.
- Bit 7: Eine *Bewegung wurde unterbrochen* .
- Bit 8: Die *Zielposition Position wurde erreicht* .
- Bit 9-16 enthalten im Fehlerfall zusätzlich den *Fehlercode* .

BEISPIEL 1:

	D-Len	Cmd	Param	
M->S	0x06	0x95	0x00 0x00 0x80 0x3F 0x07	Status-Information soll zyklisch jede Sekunde vom Modul geliefert werden. Außer Status-Bits werden auch Position, Geschwindigkeit und Strom mitversendet.
S->M	0x0F	0x95	0xD6 0xA3 0x70 0x41 0x56 0xC9 0x41 0x40 0x3C 0x41 0xEB 0x3E 0x03 0x00	zyklisch jede Sekunde Position: 0xD6..0x41, Geschwindigkeit: 0x56..0x40, Strom: 0x3C..0x3E; Modul ist in Bewegung und Referenziert (0x03); kein Fehler (0x00)

BEISPIEL 2:

	D-Len	Cmd	Param	
M->S	0x01	0x95		Status-Information einmalig abfragen. Dabei werden zuletzt abgefragte Parameter mitversendet.
S->M	0x0F	0x95	0x53 0x63 0xB7 0x41 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x61 0xD9	Position: 0x53..0x41, Geschwindigkeit: 0x00..0x00, Strom: 0x00..0x00; Modul referenziert, Bewegung beendet, Bremse betätigt (0x61); Schnellstop betätigt (0xD9)

## BEISPIEL 3:

	D-Len	Cmd	Param	
M->S	0x06	0x95	0x00 0x00 0x00 0x00 0x01	Status-Information einmalig abfragen. Außer Status-Bits wird auch Position mitversendet.
S->M	0x07	0x95	0x00 0x00 0x00 0x00 0x20 0x00	Position: 0x00..0x00; Modul nicht referenziert, Bremse betätigt (0x20); kein Fehler (0x00)

SONSTIGES: Will man Position, Geschwindigkeit und Strom unter CAN in einer Nachricht erhalten, so ist das *Fragmentierprotokoll* zu verwenden. Unter Profibus finden alle Informationen in einer Profibusnachricht platz. Ein einmal gesetzter Datencode „Daten“ wird beibehalten und braucht somit nicht jedes mal neu gesetzt werden. Wird das Modul eingeschaltet, ist der Datencode auf „0x07“ gesetzt, es werden also alle möglichen Status-Informationen komplett übermittelt.

**HINWEIS**

**Unter Profibus werden, wenn alle Parameter (Position, Geschwindigkeit, Strom) übermittelt wurden nur die unteren 8 Bit des Status angezeigt. Diese kommen nun auf Byte 14 zu liegen, wo profibusspezifisch 15 der Status immer aktuell geliefert wird. Auf Byte 15 folgt der MsgCount, welcher die oberen 8 Bit des Status Wortes überschreibt.**

**HINWEIS**

**Unter Profibus wird standardgemäß in Byte 10-13 die aktuelle Position übermittelt. Werden mit GET STATE alle Parameter angefordert, so wird die aktuelle Position in Bytes 10-13 mit dem aktuellen Stromwert überschrieben. Daher ist es empfehlenswert unter Profibus das „Modus“ Byte maximal auf „0x06“ zu setzen (Geschwindigkeit und Strom werden übermittelt).**

**HINWEIS**

**Unter Profibus sollte man mit dem „automatischen“ Abrufen vorsichtig sein. Hier kann es unter Umständen günstiger sein die Daten zu „pollen“. Insbesondere dann, wenn der FREEZE Mechanismus genutzt wird.**

### 3.2.6 CMD TOGGLE IMPULSE MESSAGE

KOMMANDO CODE: 0xE7

BESCHREIBUNG: Spontanmeldungen können hiermit aktiviert bzw. deaktiviert werden.

ANTWORT (Slave->Master): Wird der Befehl mit „ON“ (0x4F 0x4E) bestätigt, sind Spontanmeldungen aktiv. Wird der Befehl mit „OFF“ (0x4F 0x46 0x46) bestätigt, sind Spontanmeldungen abgeschaltet.

BEISPIEL:

	D-Len	Cmd	Param	
M->S	0x01	0xE7		
S->M	0x04	0xE7	0x4F 0x46 0x46	Spontanmeldung deaktiviert.

SONSTIGES: Spontanmeldungen sind beim Neustart des Moduls immer aktiviert.

## 3.3 Einstellungen

### 3.3.1 CALIB CURRENT

KOMMANDO CODE: 0x8F

BESCHREIBUNG: Die Stromsensoren werden abgeglichen. (Offset eingestellt).

ANTWORT (Slave->Master): Offset-Werte für die Phasen A,B und C als Int16.

SONSTIGES: Die ermittelten Werte werden automatisch ins EEPROM übernommen ▶ [5.3.2.2.17 \[ 76\]](#). Bei deaktiviertem ▶ [5.3.2.2.16 \[ 75\]](#) dritten Stromsensor wird der Offset für die Phase C immer zu "2048" gesetzt. Schlägt der Abgleich fehl, wird ein *ERROR CALIB CURENT* ▶ [4.2.44 \[ 57\]](#) ausgelöst

---

#### HINWEIS

Dieser Befehl benötigt *Advanced -Rechte* ▶ [5.3.3.33 \[ 111\]](#).

---

#### HINWEIS

Ein unkalibriertes Modul verhält sich nicht erwartungsgemäß, z. B. unsaubere Fahrt, Stromwerte nicht sinnvoll, Bewegungen nur in eine Richtung möglich.

---

### 3.3.2 SET CONFIG EXT

KOMMANDO CODE: 0x83

BESCHREIBUNG: Einzelne Konfigurationsparameter im Modul werden gesetzt und *dauerhaft gespeichert* ▶ 5.3 [□ 62].

- Pro Kommando kann immer nur ein einziger Konfigurationsparameter gezielt gesetzt werden.
- Die Konfigurationsparameter werden über einen 16-Bit breiten Konfigurationscode identifiziert.
- Die Bedeutung eines Konfigurationscodes hängt von dem angeschlossenen Gerät ab. Siehe hierzu entsprechende Ergänzungsanleitung.
- Der Wert eines zu setzenden Konfigurationsparameters kann geräteabhängig in verschiedenen Datenformaten übermittelt werden (siehe Definition vom „Datentyp“ unten).

PARAMETER (Master -> Slave):

- Konfigurationscode (2 Byte)
- Datentyp (1 Byte)
- Konfigurationsparameter (n Bytes)

Der „Konfigurationscode“ ▶ 5.3 [□ 62] identifiziert den zu setzenden Konfigurationsparameter. Die Definition des Datentyps, ▶ 7.1 [□ 113].

Die zurückgelieferten Werte von „Konfigurationscode“ ▶ 5.3 [□ 62] entspricht den angeforderten und erlauben so eine Zuordnung der Antwort zur Anfrage.

SONSTIGES: Zum komfortablen Setzen der Konfigurationsparameter kann das mitgelieferte Software Tool (siehe Softwareanleitung "Motion Tool Schunk") verwendet werden.

---

#### HINWEIS

**Einige Konfigurationsparameter können nur geändert werden, wenn das Modul gestoppt ist, und die Regelung deaktiviert ist. Vorher also ggf. Schnellstop ▶ 3.1.12 [□ 35] auslösen.**

---

#### HINWEIS

**Einige Konfigurationsparameter können nicht geändert werden. Wenn auf solche „Read-Only“ Daten mit „SET CONFIG EXT“ zugegriffen wird, dann wird mit Fehlercode“INFO NO RIGHTS“ 0x03 geantwortet.**

---

### 3.3.3 GET CONFIG EXT

KOMMANDO CODE: 0x82

BESCHREIBUNG: Bestimmte Konfigurationsparameter können aus dem Modul ausgelesen werden.

- Pro Kommando kann immer nur ein einzelner Konfigurationsparameter gezielt abgefragt werden.
- Die Konfigurationsparameter werden über einen 16-Bit breiten Konfigurationscode ▶ 5.3 [□ 62] identifiziert.
- Der Wert eines zurückgelieferten Konfigurationsparameters kann geräteabhängig in verschiedenen *Datenformaten* ▶ 3.3.2 [□ 43] zurückgegeben werden.

PARAMETER (Master -> Slave):

- Konfigurationscode (2 Byte)

Der „Konfigurationscode“ identifiziert den angefragten Konfigurationsparameter.

ANTWORT (Slave->Master):

- Konfigurationscode (2 Byte)
- Datentyp (1 Byte)
- Konfigurationsparameter (n Bytes)

Die zurückgelieferten Werte von „Konfigurationscode“ entsprechen den angeforderten und erlauben so eine Zuordnung der Antwort zur Anfrage. Der „Datentyp“ ist entsprechend *SET CONFIG EXT* ▶ 3.3.2 [□ 43] definiert, ▶ 7.1 [□ 113]. Die Anzahl n der Bytes des Konfigurationsparameters hängt vom „Datentyp“ ab. Beim Datentyp „DT\_STRING“ ist im ersten Byte des Konfigurationsparameters die Anzahl der folgenden ASCII Zeichen enthalten.

SONSTIGES: Zum komfortablen Lesen der Konfigurationsparameter kann das mitgelieferte *Software Tool* (siehe Softwareanleitung "Motion Tool Schunk") verwendet werden.

## 3.4 Sonstige

### 3.4.1 CMD REBOOT

KOMMANDO CODE: 0xE0

BESCHREIBUNG: Das Modul wird neu gestartet.

PARAMETER (Master -> Slave): Keine

ANTWORT (Slave->Master): Befehl wird mit „OK“ (0x4F4B) bestätigt. Danach startet das Modul neu und meldet sich nach erfolgreichem Reboot mit „INFO BOOT“ ▶ 4 [□ 49].

### 3.4.2 CHANGE USER

KOMMANDO CODE: 0xE3

BESCHREIBUNG: Der aktuelle Benutzer des Moduls wird gewechselt.

PARAMETER (Master -> Slave):

- kein Nutzer „Nutzer“ ▶ 5.3.3.33 [□ 111] wird eingestellt.
- *Kennwort* für den jeweiligen Nutzer. ANTWORT (Slave->Master): Der Befehl wird immer mit „OK (0x4F4B)“ bestätigt. Angehängt (1 Byte) wird der aktuelle Benutzer (0 = User, 1 = Diag, 2 = Profi, 3 = Advanced).

SONSTIGES: Bei der Eingabe eines falschen Kennwortes wird immer „Nutzer“ eingestellt. Bei Neustart eines Moduls ist immer „Nutzer“ aktiv.

### 3.4.3 CMD GET DIO (0xE9)

KOMMANDO CODE: 0xE9

BESCHREIBUNG: Digitale Ein-/ Ausgänge können gelesen werden.

PARAMETER (Master -> Slave):

Es sind 3 Typen von Anfragen möglich, welche sich in der Anzahl der übergebenen Parameter und der Art und Anzahl der angefragten digitalen Ein- Ausgänge unterscheiden. Abhängig vom Typ (1-3) der Anfrage werden auch unterschiedliche Antworten vom Modul zurückgeschickt.

**Typ 1:** Es wird der aktuelle Zustand aller digitalen Eingänge als Bit-Vektor gelesen. (Die Ausgänge können mit Typ 1 Anfragen nicht gelesen werden.):

PARAMETER (Master -> Slave):

- kein

ANTWORT (Slave->Master):

- *Bit Vektor* (4 Byte)

**Typ 2:** Ein bestimmter digitaler Eingang oder Ausgang wird gelesen. Welcher Eingang oder Ausgang gelesen werden soll wird durch den Parameter „Adresse“ bestimmt. Der „Zustand“ des adressierten digitalen Ein- oder Ausgangs wird in Bit 0 des übertragenen Datenbytes der Antwort übermittelt:

PARAMETER (Master -> Slave):

- Adresse (4 Byte)

ANTWORT (Slave->Master):

- Zustand (1 Byte)

**Typ 3:** Eine aufeinanderfolgende Reihe von digitalen Eingängen oder Ausgängen wird gelesen. Welche Eingänge oder Ausgänge gelesen werden sollen wird durch die Parameter „Adresse“ und „Länge“ bestimmt. Beginnend mit dem Ein- oder Ausgang an „Adresse“ werden so viele Ein-/ Ausgänge gelesen wie der Parameter „Länge“ angibt. In der Antwort wird der Zustand der adressierten digitalen Ein- oder Ausgänge als Bit-Vektor in den übertragenen n Datenbytes „Zustände“ übermittelt. Die Anzahl der übermittelten Datenbytes hängt von der angefragten „Länge“ ab und berechnet sich wie folgt: „(Länge + 7) / 8“, wobei „/“ die Integer Division (Division ohne Rest) bezeichnet. Für „Länge“ von 1-8 wird also mit einem Byte geantwortet, für „Länge“ von 9-16 wird mit zwei Bytes geantwortet, usw.:

PARAMETER (Master -> Slave):

- Adresse (4 Byte)
- Länge (1 Byte)

ANTWORT (Slave->Master):

- Zustände (n Bytes)

BEMERKUNG

- Anfragen vom Typ 1 können nur Eingänge lesen (nicht Ausgänge).
- Welche Adressen für Typ 2 bzw. 3 gültig sind hängt vom Modul ab. Siehe hierzu jeweilige Ergänzungsanleitung.

**Beispiel:**

Typ 1	D-Len	Cmd	Param	
M->S	0x01	0xE9		Typ1: Alle Eingänge lesen
S->M	0x05	0xE9	0x01 0x02 0x04 0x08	Typ 1: Eingang 1, 10, 19 und 28 sind gesetzt, alle anderen Eingänge sind gelöscht

## 3.5 Fragmentierung

### 3.5.1 FRAG ACK (0x87)

KOMMANDO CODE: 0x87

BESCHREIBUNG: Bestätigung für ein ordnungsgemäß verarbeitetes Fragment.

PARAMETER (Master -> Slave): *D-Len Code des erhaltenen Fragments* (UInt16), wenn die Steuerung dem Modul das Fragment bestätigt.

ANTWORT (Slave->Master): *D-Len Code des erhaltenen Fragments* (UInt16), wenn das Modul der Steuerung das Fragment bestätigt.

BEISPIEL: Siehe *Ausgesuchte Beispiele* 130.

---

#### HINWEIS

**Dieser Befehl wird nur beim Profibus 18) benötigt, wenn eine Fragmentierung von Nachrichten notwendig wird.**

---

### 3.5.2 FRAG START (0x84)

KOMMANDO CODE 0x84

BESCHREIBUNG: Zeigt bei einer fragmentierten Nachricht, dass es sich um das erste Fragment handelt.

PARAMETER (Master -> Slave): Keine

ANTWORT (Slave->Master): Keine

BEISPIEL: Siehe *Ausgesuchte Beispiele* 130.

SONSTIGES: Wird direkt hinter dem D-Len Byte geschrieben. Zählt allerdings nicht zu D-Len, da es nur als Marker dient.

### 3.5.3 FRAG MIDDLE (0x85)

KOMMANDO CODE: 0x85

BESCHREIBUNG: Zeigt bei einer fragmentierten Nachricht, dass es sich um ein mittleres Fragment handelt.

PARAMETER (Master -> Slave): Keine

ANTWORT (Slave->Master): Keine

BEISPIEL: Siehe *Ausgesuchte Beispiele* 130.

SONSTIGES: Wird direkt hinter dem D-Len Byte geschrieben. Zählt allerdings nicht zu D-Len, da es nur als Marker dient.

### 3.5.4 FRAG END (0x86)

KOMMANDO CODE 0x86

BESCHREIBUNG: Zeigt bei einer fragmentierten Nachricht, dass es sich um das letzte Fragment handelt.

PARAMETER (Master -> Slave): Keine

ANTWORT (Slave->Master): Keine

BEISPIEL: Siehe *Ausgesuchte Beispiele*. 130.

Sonstiges: Wird direkt hinter dem D-Len Byte geschrieben. Zählt allerdings nicht zu D-Len, da es nur als Marker dient.

## 4 Info- und Fehlermeldungen



### Fehlermeldung

Im Fehlerfall hat D-Len im Datenrahmen vom Modul zur Steuerung immer genau den Wert "2".

Im Kommando-Byte steht entweder das Kommando welches fehlgeschlagen ist oder ein Fehler-Kommando, ▶ 4.1 [49].

Das Parameter-Byte enthält die Information über die Fehlerursache.

### 4.1 Fehler Behandlung

#### 4.1.1 CMD ERROR

KOMMANDO CODE: 0x88

BESCHREIBUNG:

Ein schwerwiegender Fehler ist aufgetreten, welcher einen Benutzereingriff notwendig macht. Diese Fehler müssen mit „*CMD ACK*“ ▶ 4.1.4 [50] quittiert werden. Das Modul ist nicht betriebsbereit. Der Motor ist stromlos geschaltet und die Bremse eingefallen. Diese Fehlermeldungen werden regelmäßig alle 15 Sekunden vom Modul zur Steuerung gesendet, bis der Fehler quittiert ist.

Alle *Fehlercodes* ▶ 4.2 [51] als Übersicht.

- CAN: die ersten drei Bit des Identifiers ergeben sich zu „0x3“
- Profibus: eine erweiterte Diagnose wird generiert



### ⚠️ WARNUNG

#### Verletzungsgefahr!

Bei Modulen ohne Bremse kann das Modul „durchsacken“, da der Motor bei einem schwerwiegenden Fehler stromlos geschaltet wird!

#### 4.1.2 CMD WARNING

KOMMANDO CODE: 0x89

BESCHREIBUNG:

- Softwareendanschläge wurden verletzt. Hier wird zunächst ein Schnellstop ausgelöst und muss quittiert werden. Anschließend ist das Modul bedingt betriebsbereit. (Fahrten aus dem Softwareendanschlag heraus sind gestattet). Ist das Modul aus dem Softwareendanschlag herausgefahren, wird die Warnung gelöscht.
- *Temperaturobergrenze* ▶ 5.3.2.1.13 [□ 69] wurde überschritten. Das Modul wird sich in 1[*min*] abschalten, wenn die Temperatur nicht gesenkt wird.

Diese Fehlermeldungen werden regelmäßig vom Modul zur Steuerung gesendet (alle 30 Sec.). Alle *Fehlercodes*

▶ 4.2 [□ 51] als Übersicht.

- CAN: die ersten drei Bit des Identifiers ergeben sich zu „0x3“
- Profibus: eine erweiterte Diagnose wird generiert

#### 4.1.3 CMD INFO

KOMMANDO CODE: 0x8A

BEISPIEL:

	D-Len	Cmd	Param	
S->M	0x02	0x8A	0x10	„INFO TIMEOUT“ ▶ 4.2.9 [□ 52]

BESCHREIBUNG: Das Modul sendet eine Informationsnachricht.

BEMERKUNG: Bei behobenen Fehlern und einem Neustart des Moduls werden ebenfalls „Info Nachrichten“ versendet.

(▶ 4.2.1 [□ 51] ▶ 4.2.7 [□ 52])

#### 4.1.4 CMD ACK

KOMMANDO CODE: 0x8B

BESCHREIBUNG: Quittierung einer anstehenden Fehlermeldung.

BEISPIELE:

OK

OK + no error

OK + no error + <error>

#### 4.1.5 Detaillierte Fehlerinformationen

Über die Parameter

*errorDetail* ▶ 5.3.3.5 [□ 104], *errorLine* ▶ 5.3.3.4 [□ 104] und

*errorFile* ▶ 5.3.3.6 [□ 104] können für den Service wichtige

Informationen zu dem zuletzt anliegenden Fehler abgefragt werden.

## 4.2 Info- und Fehlercodes

### 4.2.1 INFO BOOT

Das Modul hat erfolgreich gebootet.

Code: **0x0001**

Wird nach dem Einschalten und erfolgreicher Initialisierung des Moduls ausgelöst. Tritt diese Meldung während des Betriebes gehäuft auf, muss die Logikspannung überprüft werden. Außerdem besteht die Möglichkeit eines defekten Leistungstreibers.

### 4.2.2 INFO NO RIGHTS

Es fehlen die passenden Rechte um das Kommando auszuführen.

Code: **0x03**

### 4.2.3 INFO UNKNOWN COMMAND

Das gesendete Kommando ist unbekannt.

Code: **0x04**

### 4.2.4 INFO FAILED

Das Kommando ist fehlgeschlagen.

Code: **0x05**

Alle Parameter sind richtig gesetzt, aber aus anderen Gründen ist die Ausführung des Kommandos derzeit nicht möglich. z. B. befindet sich das Modul im Schnellstop-Zustand.

### 4.2.5 NOT REFERENCED

Modul ist nicht referenziert und kann daher das Kommando nicht ausführen.

Code: **0x06**

Um eine Positionsfahrt auszuführen ist eine Referenzierung notwendig.

### 4.2.6 INFO SEARCH SINE VECTOR

Code:: 0x0007

Der Raumzeiger für die Sinus-Kommutierung wird gesucht. Die Phasen werden mit 60% des Maximalstromes bestromt.

Code: **0x0007**

Wenn die Stillstandkommutierung nicht aktiv ist und die Kommutierart auf PMSM steht, wird einmalig nach dem Einschalten und vor der Durchführung eines Bewegungsbefehles der Raumzeiger gesucht, ▶ [1.9 \[21\]](#) und ▶ [5.3.2.2.3 \[71\]](#).

#### **4.2.7 INFO NO ERROR**

Es liegen keine weiteren Fehlermeldungen an.

Code: **0x0008**

Wenn kein Fehler und keine Warnung vorhanden ist oder wenn das Modul aus den Softwareendanschlägen herausgefahren ist, wird diese Infomeldung angezeigt.

#### **4.2.8 INFO COMMUNICATION ERROR**

Ein Fehler in der Kommunikation ist aufgetreten.

Code: **0x09**

Der Anschluss des Kommunikationskabels und externen Einflüsse auf das Kommunikationskabel müssen geprüft werden.

#### **4.2.9 INFO TIMEOUT**

Es ist in der Kommunikation eine Zeitüberschreitung aufgetreten.

Code: **0x10**

Daten konnten nicht verschickt werden oder es wurden noch weiter Daten erwartet die nicht rechtzeitig eingetroffen sind.

#### **4.2.10 INFO WRONG DATA TYPE**

Der Datentyp passt nicht zum Parameter.

Code: **0x12**

#### **4.2.11 INFO RESTART**

Das Modul oder die Steuerung wurde neu gestartet.

Das Modul ist nicht betriebsbereit. Die Meldung ist quittierpflichtig, zum Quittieren müssen die Bits "Halt" und "Schnell Halt" auf "1" gesetzt sein, MCSL\_Beispielhafte\_Fahrbeefhle.

Code: **0x13**

#### **4.2.12 INFO CHECKSUM**

Die Checksumme ist fehlerhaft, die Daten sind ungültig.

Code: **0x19**

#### **4.2.13 INFO VALUE LIMIT MAX**

Der vorgegebene Wert überschreitet den maximal erlaubten Vorgabewert.

Code: **0x1B**

**4.2.14 INFO VALUE LIMIT MIN**

Der vorgegebene Wert unterschreitet den minimal erlaubten Vorgabewert.

Code: **0x1C**

**4.2.15 INFO MESSAGE LENGTH**

Die Kommando-Länge passt nicht zu den erhaltenen Daten.

Code: **0x1D**

**4.2.16 INFO WRONG PARAMETER**

Einer der angegebenen Parameterwerte liegt außerhalb des zulässigen Bereiches.

Code: **0x1E**

Wird ein Parameterwert als nicht zulässig erkannt, werden alle alten Parameterwerte beibehalten, auch wenn die übrigen Parameterwerte im gültigen Bereich liegen sollten.

**4.2.17 INFO UNKNOWN PARAMETER**

Der angeforderte Parameter ist unbekannt.

Code: **0x23**

**4.2.18 ERROR FILE NOT FOUND**

Die zu verarbeitende Datei befindet sich nicht auf dem USB-Stick oder der USB-Stick ist defekt.

Code: **0x60**

**4.2.19 ERROR FILE IS CORRUPT**

Die zu verarbeitende Datei auf dem USB-Stick ist defekt.

Code: **0x61**

**4.2.20 ERROR FILE TYPE WRONG**

Der Datei-Typ ist nicht korrekt.

Code: **0x62**

- Datei neu erstellen.

**4.2.21 ERROR FILE SYSTEM WRONG**

Das Datei-System vom USB-Stick ist fehlerhaft.

Code: **0x64**

- Prüfen, ob der USB-Stick mit FAT16 oder FAT32 formatiert ist.

**4.2.22 ERROR FILE READ**

Beim Lesen der Datei ist ein Lesefehler aufgetreten.

Code: **0x65**

#### **4.2.23 ERROR FILE IS NOT CREATED**

Es konnte keine Datei erzeugt werden.

Code: **0x66**

- Prüfen, ob USB-Stick defekt oder Schreibschutz aktiviert ist.

#### **4.2.24 ERROR FILE WRITE**

Es ist beim Schreiben der Datei ein Schreibfehler aufgetreten.

Code: **0x67**

#### **4.2.25 ERROR REBOOT**

Ein Parameter wurde geschrieben, der ein Reboot benötigt.

Code: **0x7C**

- Modul aus- und einschalten.

#### **4.2.26 ERROR MOTOR PHASE**

Eine Motor-Phase ist nicht ordnungsgemäß angeschlossen.

Code: **0x7D**

#### **4.2.27 ERROR WRONG RAMP TYPE**

Für die Positionsfahrt ist kein gültiges Verfahrprofil ausgewählt.

Code: **0xC8**

#### **4.2.28 ERROR WRONG DIRECTION**

Das Modul bewegt sich bei der Prüfung des Pseudoabsolutwertgebers in die falsche Richtung

Code: **0xD1**

- Sinus-Zeiger prüfen.

#### **4.2.29 ERROR CONFIG MEMORY**

Der Konfigurationsbereich ist fehlerhaft. Das Schreiben des EEPROMs ist fehlgeschlagen oder das EEPROM ist defekt.

Code: **0xD2**

#### **4.2.30 ERROR SOFT LOW**

Das Modul hat den unteren Softwareendanschlag überfahren.

Code: **0xD5**

- Gegebenenfalls den Fehler quittieren und das Modul mit einem Verfahrbefehl aus dem Softwareendanschlag herausbewegen.

#### 4.2.31 ERROR SOFT HIGH

Das Modul hat den oberen Softwareendanschlag überfahren.

Code: **0xD6**

- Gegebenenfalls den Fehler quittieren und das Modul mit einem Verfahrbefehl aus dem Softwareendanschlag herausbewegen.

#### 4.2.32 ERROR SERVICE

Ein Fehler ist aufgetreten, der ausschließlich durch den SCHUNK behoben werden kann.

Code: **0xD8**

Mit der detaillierten Fehlerinformation kann der Fehler durch SCHUNK genau lokalisiert werden, ► [4.1.5 \[□ 50\]](#).

Mit dem Service von SCHUNK in Verbindung setzen und folgende Daten bereithalten:

- Modultyp
- Seriennummer des Moduls
- Beschreibung wie es zu dem Fehler gekommen ist

#### 4.2.33 ERROR FAST STOP

Es wurde ein Schnellstop ausgelöst, MCSL\_Eingehende Prozessdaten.

Das Modul ist nicht betriebsbereit. Es wird kein Fehlerzustand ausgelöst. Die Meldung ist quittierpflichtig.

Code: **0xD9**

#### 4.2.34 ERROR TOW

Ein Schleppfehler ist aufgetreten.

Code: **0xDA**

- Belastung reduzieren.
- Parameter "Schleppfehler" prüfen.

#### 4.2.35 ERROR VPC3

Der Regler arbeitet fehlerhaft oder ist defekt.

Code: **0xDB**

#### 4.2.36 ERROR FRAGMENTATION

Im Fragmentierungsprotokoll ist ein Fehler aufgetreten. Es sind Datenpakete verloren gegangen.

Code: **0xDC**

#### 4.2.37 ERROR COMMUTATION

Motor kann nicht kommutieren.

Code: **0xDD**

Tritt dieser Fehler gehäuft auf ist die Kommutierart falsch gewählt. Bei Block-Kommutierung sind die Hallgeber defekt oder nicht angeschlossen. Bei Sinus-Kommutierung liegt ein Fehler im Positionsmesssystem vor.

#### 4.2.38 ERROR I2T

Ein I<sup>2</sup>T Fehler ist aufgetreten.

Code: **0xDF**

- Belastung des Motors reduzieren.

#### 4.2.39 ERROR CURRENT

Der maximale Strom wurde überschritten.

Code: **0xDE**

- Belastung des Motors reduzieren.

#### 4.2.40 ERROR TOO FAST

Die maximale Geschwindigkeit wurde überschritten.

Code: **0xE4**

#### 4.2.41 ERROR POS SYSTEM

Das Positionsmesssystem arbeitet nicht richtig.

Code: **0xE5**

- Konfiguration des Moduls prüfen.

#### 4.2.42 ERROR RESOLVER CHECK FAILED

Ein Parameter der Resolvereinstellung ist fehlerhaft.

Code: **0xEB**

#### 4.2.43 ERROR MATH

Es ist ein mathematischer Fehler aufgetreten, z. B. Division durch Null.

Code: **0xEC**

Meist ist ein Konfigurationsparameter falsch und es kommt zu einer Überschreitung des Wertebereichs. In den meisten Fällen ist ein Reglerparameter falsch eingestellt.

Mit der detaillierten Fehlerinformation kann der Fehler durch SCHUNK genau lokalisiert werden, ► [4.1.5 \[ 50\]](#).

#### 4.2.44 ERROR CALIB CURRENT

Die gemessenen Werte der Stromsensoren liegen außerhalb der Toleranzgrenzen.

Code: **0xEE**

- Modul kalibrieren.
  - Tritt der Fehler wiederholt auf ist die Strommessung defekt.

#### 4.2.45 ERROR INITIALIZE

Das Modul konnte nicht richtig initialisiert werden.

Code: **0xE0**

- Konfigurationsparameter prüfen.

Mit der detaillierten Fehlerinformation kann der Fehler durch SCHUNK genau lokalisiert werden, ► [4.1.5 \[□ 50\]](#).

#### 4.2.46 ERROR INTERNAL

Ein interner Fehler ist aufgetreten.

Code: **0xE1**

Die Firmware befindet sich in einem undefinierten Zustand.

Mit dem Service von SCHUNK in Verbindung setzen und folgende Daten bereithalten:

- Modultyp
- Seriennummer des Moduls
- Beschreibung wie es zu dem Fehler gekommen ist

#### 4.2.47 ERROR CONNECTION TEMP LOW

Die minimal zulässige Temperatur der Kommunikationsplatine wurde unterschritten.

Code: **0x6A**

- Modul aufwärmen.

#### 4.2.48 ERROR CONNECTION TEMP HIGH

Die maximal zulässige Temperatur der Kommunikationsplatine wurde überschritten.

Code: **0x6B**

- Modul abkühlen lassen.
- Belastung reduzieren.

#### 4.2.49 ERROR MOTOR TEMP LOW

Die minimal zulässige Temperatur des Motors wurde unterschritten.

Code: **0x6C**

- Modul aufwärmen.

#### 4.2.50 ERROR MOTOR TEMP HIGH

Code:: 0x6D

Die maximal zulässige Temperatur des Motors wurde überschritten.

Code: **0x6D**

- Modul abkühlen lassen.
- Belastung reduzieren.

#### 4.2.51 ERROR TEMP LOW

Der minimal zulässige Temperaturbereich der Hauptplatine wurde unterschritten.

Code: **0x70**

- Modul aufwärmen.

#### 4.2.52 ERROR TEMP HIGH

Der maximal zulässige Temperaturbereich der Hauptplatine wurde überschritten.

Code: **0x71**

- Modul abkühlen lassen.
- Belastung reduzieren.

#### 4.2.53 ERROR LOGIC LOW

Die Logikspannung ist unterhalb des Grenzwertes.

Code: **0x72**

- Logikspannung prüfen.

#### 4.2.54 ERROR LOGIC HIGH

Die Logikspannung ist oberhalb des Grenzwertes.

Code: **0x73**

- Logikspannung prüfen.

**4.2.55 ERROR MOTOR VOLTAGE LOW**

Die Motorspannung ist unterhalb des Grenzwertes.

Code: **0x74**

- Motorspannung prüfen.
    - Gegebenenfalls ist das Netzteil für die Motorspannung unterdimensioniert oder die Kabel der Spannungsversorgung zum Modul sind falsch dimensioniert.
- 

**HINWEIS**

MotorVoltageLow ist ein schwerwiegender Fehler, wenn das Modul bewegt wird.

---

**4.2.56 ERROR MOTOR VOLTAGE HIGH**

Die Motorspannung ist oberhalb des Grenzwertes.

Code: **0x75**

**HINWEIS**

Tritt dieser Fehler wiederholt auf, wird das Modul gesperrt und kann nur durch SCHUNK wieder in Betrieb genommen werden.

---

- Motorspannung prüfen.
    - Gegebenenfalls kann ein externer Bremshopper eingesetzt werden.
- 

**4.2.57 ERROR CABLE BREAK**

Die Kommunikation zur Steuerung war unterbrochen.

Code: **0x76**

**HINWEIS**

Der Fehler wird erst angezeigt, wenn die Kommunikation wieder hergestellt ist.

---

- Kommunikationskabel prüfen.
  - Das Kommunikationskabel ist defekt.

#### 4.2.58 ERROR LIFE SIGN

Timeout der internen Modulkommunikation aufgrund eines internen Fehlers.

Code: **0x7A**

- Modul muss neu gestartet werden.

Möglich Ursachen:

- Externe Störungen auf der Logikspannung vorhanden, hierzu Vorgaben zu den Basisdaten in produktspezifischer Montage- und Betriebsanleitung beachten.  
Zusätzliche mögliche Maßnahme: Getrennte Zuführung von Logik- und Leistungsspannung über separate Netzteile.
- Störung auf der Busleitung, hierzu die Vorgaben zur Busphysik der PROFIBUS Nutzerorganisation e.V. (PNO) beachten.  
Mögliche Maßnahme: Senkung der Baudrate.

#### 4.2.59 ERROR CUSTOM DEFINED

In einer kundenspezifischen Funktion ist ein Fehler aufgetreten.

Code: **0x7B**

#### 4.2.60 ERROR OVERSHOOT

Das Modul ist über die Zielposition hinausgefahren.

Code: **0x82**

- Stromvorgabe erhöhen.
  - Der zu Abbremsen benötigte Strom ist zu gering.
- Reglerparameter prüfen.

#### 4.2.61 ERROR HARDWARE VERSION

Die Hardware der verschiedenen Komponenten passt nicht zusammen. Eine auf dem USB-Stick gespeicherte Datei kann nicht mit der vorliegenden Hardware verarbeitet werden.

Code: **0x83**

#### 4.2.62 ERROR SOFTWARE VERSION

Die Software der verschiedenen Komponenten passt nicht zusammen. Eine auf dem USB-Stick gespeicherte Datei kann nicht mit der vorliegenden Software-Version verarbeitet werden.

Code: **0x84**

## 5 Konfigurationsparameter

### 5.1 Allgemein

Der Zugriff auf die jeweiligen Parameter ist in der Firmware der Module geregelt. Es gibt hierbei verschiedene Nutzer, welche jeweils über Passwörter geschützt sind. Eine Änderung des Nutzers ist über den Befehl "CHANGE USER" ▶ 3.4.2 [□ 45] möglich bzw. über ändern des Parameters "user" (0x5610)

▶ 5.3.3.33 [□ 111]. Folgende Benutzer sind dem Modul bekannt:

1. Nutzer:  
Standard Nutzer. Ist beim Einschalten des Moduls aktiv. Kann das Modul sehr eingeschränkt parametrieren, aber vollständig bedienen.
2. Profi:  
Kann alle wichtigen Parameter ändern. Bei einer falschen Parametrierung kann es zu unvorhergesehenem Verhalten kommen. Das Modul kann aber nicht zerstört werden.
3. Advanced:  
Kann alle wichtigen Parameter ändern. Eine falsche Parametrierung kann zur Zerstörung des Moduls führen!
4. Root:  
Zugriff auf alle Parameter. Eine falsche Parametrierung kann zur Zerstörung des Moduls führen!
5. Schunk:  
Parameter kann nur im Werk geändert werden.
6. Disabled:  
Parameter kann nicht geändert werden.

Folgende Wertebereiche werden genutzt:

- MAX\_BOOL -> 1
- MAX\_INT8 -> 27
- MAX\_INT16 -> 32767
- MAX\_INT32 -> 2147483647
- MAX\_UINT8 -> 255
- MAX\_UINT16 -> 65535
- MAX\_UINT32 -> 4294967295
- MAX\_CHAR -> 0xFF
- MAX\_ENUM -> 0xFFFF
- MAX\_FLOAT -> 3.40282347E+38
- MIN\_BOOL -> 0
- MIN\_INT8 -> (-MAX INT8 - 1)
- MIN\_INT16 -> (-MAX INT16 - 1)
- MIN\_INT32 -> (-MAX INT32 - 1)

- MIN\_UINT8 -> 0
- MIN\_UINT16 -> 0
- MIN\_UINT32 -> 0
- MIN\_CHAR -> 0
- MIN\_ENUM -> 0
- MIN\_FLOAT -> 1.17549435E-38

### 5.2 Parametrierung mit Hilfe eines USB-Sticks

#### 5.2.1 Parameter lesen

- USB Stick ECM/V6R einstecken (siehe Hardwareanleitung)
- USB Stick aktivieren (siehe Hardwareanleitung)
- Idxx\_???.sav Datei wird erzeugt (xx = ID Nummer Modul hexadeziemal, ??? fortlaufende Nummer)
- Idxx\_???.sav Datei am PC öffnen

#### 5.2.2 Parameter schreiben

- Mit MTS Tool IDxx Set.par Datei erzeugen (xx = ID Nummer Modul bzw. " xx" wenn Parameter Datei auf Module mit unterschiedlicher ID geschrieben werden soll)
- Datei auf USB Stick speichern (gegebenfalls umbenennen)
- USB Stick in ECM/V6R einstecken (siehe Hardwareanleitung)
- USB Stick aktivieren (siehe Hardwareanleitung)
- Parameter schreiben aktivieren (siehe Hardwareanleitung)
- Idxx\_???.log Datei wird erzeugt (xx = ID Nummer Modul, ??? fortlaufende Nummer)
- Idxx\_???.sav Datei am PC öffnen (es wird Protokoll angezeigt welche Fehler während des Parametrierens aufgetreten sind)

### 5.3 Parameter

#### 5.3.1 Zyklisch

In diesem Abschnitt sind alle Daten aufgeführt, welche zyklisch mit dem Modul ausgetauscht werden. Ein Zugriff auf diese Daten ist prinzipiell azyklisch möglich, wird jedoch nicht empfohlen.

##### 5.3.1.1 ControlWord

PARAMETER CODE: 0x7D01

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ:(UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.1.2 DigitalOut**

PARAMETER CODE: 0x7D02

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**5.3.1.3 DesiredPosition**

PARAMETER CODE: 0x7D03

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**5.3.1.4 DesiredVel**

PARAMETER CODE: 0x7D04

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.1.5 **DesiredCur**

PARAMETER CODE: 0x7D05

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.1.6 **DesiredAcc**

PARAMETER CODE: 0x7D06

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.1.7 **DesiredJerk**

PARAMETER CODE: 0x7D07

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.1.8 **DesiredPrePosition**

PARAMETER CODE: 0x7D08

Zugriffsrechte lesen - schreiben: *USER* - *USER*

BESCHREIBUNG: nur mit Schunk Drive Protocol (SDP) für Profibus/PROFINET erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.1.9 **StatusWord**

PARAMETER CODE: 0x7D09

Zugriffsrechte lesen - schreiben: *USER* - *Disabled*

BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.1.10 ActiveCopy**

PARAMETER CODE: 0x7D0A

Zugriffsrechte lesen - schreiben: *USER - Disabled*BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**5.3.1.11 DigitalIn**

PARAMETER CODE: 0x7D0B

Zugriffsrechte lesen - schreiben: *USER - Disabled*BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**5.3.1.12 ActPosition**

PARAMETER CODE: 0x7D0C

Zugriffsrechte lesen - schreiben: *USER - Disabled*BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**5.3.1.13 ActVel**

PARAMETER CODE: 0x7D0D

Zugriffsrechte lesen - schreiben: *USER - Disabled*BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**5.3.1.14 ActCur**

PARAMETER CODE: 0x7D0E

Zugriffsrechte lesen - schreiben: *USER - Disabled*BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.1.15 **ErrorCode**

PARAMETER CODE: 0x7D0F

Zugriffsrechte lesen - schreiben: *USER - Disabled*

BESCHREIBUNG: nur mit *Schunk Drive Protocol (SDP)* erreichbar.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

#### 5.3.2 **Permanent**

Hier sind alle Parameter aufgeführt, welche dauerhaft im internen EEPROM des Moduls gespeichert werden.

Zugriff ist möglich über:

- MTS
- USB-Stick ▶ [5.2 \[ 62\]](#)
- ProfiBus DPV1 (Asynchroner Datenaustausch)

##### 5.3.2.1 **Gerät**

###### **Seriennummer**

PARAMETER CODE: 0x7D73

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Seriennummer des Moduls

Daten Typ: (UINT32) 0 - MAX\_UINT32

###### **Ist Greifer**

PARAMETER CODE: 0x7D74

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Zeigt an, ob das Modul ein Greifer ist.

Daten Typ: (BOOL) MIN-BOOL - MAX\_BOOL

**Motordrehrichtung invert.**

PARAMETER CODE: 0x7D76

Zugriffsrechte lesen - schreiben:

*USER - PROFI*BESCHREIBUNG: Drehrichtung des Motors wird festgelegt.

---

**HINWEIS****Falsche Konfiguration kann zu unerwarteten Effekten führen.  
(Modul dreht unerwartet sehr schnell!)**

---

Daten Typ: (BOOL) false - true

**Positionsmessung invert.**

PARAMETER CODE: 0x7D77

Zugriffsrechte lesen - schreiben:

*USER - PROFI*BESCHREIBUNG: Messrichtung des Positionsmesssystems wird festgelegt. Vertauschte A und B Spur des Encoders können per Software getauscht werden.

---

**HINWEIS****Falsche Konfiguration kann zu erwarteten Effekten führen.  
(Modul dreht unerwartet sehr schnell!)**

---

Daten Typ: (BOOL) false - true

**HINWEIS****Werden gleichzeitig Drehrichtung des Motors und des Positionsmesssystems umgekehrt, kann aus einem "linksdrehenden" Modul ein "rechtsdrehendes", bzw. aus einem "positiv" öffnenden Greifer ein "positiv" schließender Greifer konfiguriert werden.**

---

**Endlos**

PARAMETER CODE: 0x7D78

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Die Achse kann endlos drehen.

Daten Typ: (BOOL) false - true

---

**HINWEIS****Für Greifer ist diese Option nicht empfehlenswert!**

---

### **Digitale Eingänge**

PARAMETER CODE: 0x7D79

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Verwendung der digitalen Eingänge.

Daten Typ: DIGITAL\_IN\_NORMAL -  
MAX\_DIGITAL\_IN\_TYPE, ▶ 7.1 [□ 113]

### **Digitale Ausgänge**

PARAMETER CODE: 0x7D7A

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Verwendung der digitalen Ausgänge.

Daten Typ: DIGITAL\_OUT\_NORMAL -  
MAX\_DIGITAL\_OUT\_TYPE, ▶ 7.1 [□ 113]

### **Min. Position**

PARAMETER CODE: 0x7D7B

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Minimal erlaubte Position (Softwareendanschlag).  
Wird bei gesetztem "Endlos" ▶ 5.3.2.1.5 [□ 67] ignoriert. Wird  
beim Referenzieren ▶ 5.3.2.4.1 [□ 82] mit "Hubkontrolle"  
verwendet.

Daten Typ: (FLOAT) -MAX\_FLOAT - maxPosition

### **Max. Position**

PARAMETER CODE: 0x7D7C

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Maximal erlaubte Position  
(Softwareendanschlag). Wird bei gesetztem  
"Endlos" ▶ 5.3.2.1.5 [□ 67] ignoriert. Wird beim  
Referenzieren ▶ 5.3.2.4.1 [□ 82] mit "Hubkontrolle" verwendet.

Daten Typ: (FLOAT) minPosition - MAX\_FLOAT

**Min. Temperatur Motor**

PARAMETER CODE: 0x7D7F

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Minimal zulässige Arbeitstemperatur des Motors (falls Motortemperatursensor angeschlossen). Wird die Arbeitstemperatur unterschritten kommt es zu einer Fehlermeldung

▶ [4.2.49 \[□ 58\]](#).

Daten Typ: (FLOAT) -MAX\_FLOAT - maxTempMotor

**Max. Temperatur Motor**

PARAMETER CODE: 0x7D80

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximal zulässige Arbeitstemperatur des Motors (falls Motortemperatursensor angeschlossen). Wird die Arbeitstemperatur überschritten kommt es zu einer Fehlermeldung

▶ [4.2.50 \[□ 58\]](#).

Daten Typ: (FLOAT) -MAX\_FLOAT - maxTempBoard2

**Min. Temperatur Hauptplatine**

PARAMETER CODE: 0x7D7D

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Minimal zulässige Arbeitstemperatur. Wird die Arbeitstemperatur unterschritten kommt es zu einer Fehlermeldung ▶ [4.2.51 \[□ 58\]](#).

Daten Typ: (FLOAT) -MAX\_FLOAT - maxTempBoard1

**Max. Temperatur Hauptplatine**

PARAMETER CODE: 0x7D7E

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximal zulässige Arbeitstemperatur. Wird die Arbeitstemperatur überschritten kommt es zu einer Warnung. Sollte die Temperatur nicht innerhalb 1[min] sinken, kommt es zu einer Fehlermeldung ▶ [4.2.52 \[□ 58\]](#).

Daten Typ: (FLOAT) minTemoBoard1 - MAX\_FLOAT

**Min. Temperatur Opt. Komm.**

PARAMETER CODE: 0x7D81

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Minimal zulässige Arbeitstemperatur des zweiten Temperatursensors. Wird die Arbeitstemperatur unterschritten kommt es zu einer Fehlermeldung ▶ 4.2.47 [□ 57].

Daten Typ: (FLOAT) -MAX\_FLOAT - maxTempBoard2

**Max. Temperatur Opt. Komm.**

PARAMETER CODE: 0x7D82

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximal zulässige Arbeitstemperatur des zweiten Temperatursensors. Wird die Arbeitstemperatur überschritten kommt es zu einer Fehlermeldung ▶ 4.2.48 [□ 57].

Daten Typ: (FLOAT) minTempBoard2 - MAX\_FLOAT

**5.3.2.2 Motor**

**Seriennummer**

PARAMETER CODE: 0x7D1E

Zugriffsrechte lesen - schreiben:

*USER - SCHUNK*

BESCHREIBUNG: Seriennummer des eingebauten Motors.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**Spannung**

PARAMETER CODE: 0x7D1F

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Nennspannung des Motors. Hiervon wird die Ansteuerung der Bremse abgeleitet (Tastverhältnis), sowie die maximal zulässige PWM für Testzwecke berechnet. Für automatische Reglerkonfiguration wird dieser Wert ebenfalls benötigt.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT8) 24 oder 48

**ACHTUNG**

**Eine falsche Eingabe kann zur Zerstörung der Elektronik führen!**

---

**Typ**

PARAMETER CODE: 0x7D20

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Der Motortyp kann ausgewählt werden.

- DC (0x00): Bürstenbehafteter Gleichstrommotor
- BLDC (0x01): Elektronisch kommutierter bürstenloser Gleichstrommotor mit Block-Kommutierung.
- PMSM (0x02): Elektronisch kommutierter bürstenloser Gleichstrommotor mit Sinus-Kommutierung.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (ENUM) ▶ 7.1 [🔗 113] MOTOR\_DC - MAX\_MOTORTYPE -1

**ACHTUNG****Eine falsche Eingabe kann zur Zerstörung der Elektronik führen!****I<sup>2</sup>T**

PARAMETER CODE: 0x7D21

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Die I<sup>2</sup>T Überwachung kann aktiviert werden. Bei zu hoher Belastung wird ein I<sup>2</sup>T Fehler ▶ 4.2.38 [🔗 56] ausgelöst. Bei der I<sup>2</sup>T Überwachung wird davon ausgegangen, dass der maximal Strom für 3 Sekunden (entspricht 100%) anliegen darf. Wird ein Wert < 100% eingetragen verkürzt sich die Zeit dementsprechend (Die I<sup>2</sup>T Überwachung löst früher aus).

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT8) 10 - 100

### **Max. Strom**

PARAMETER CODE: 0x7D22

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximaler zulässiger Strom, welcher kurzzeitig durch die Phasen des Motors fließen darf. Wird dieser Strom längere Zeit (mehrere ms) überschritten wird ein Schnellstop ausgelöst mit Fehlermeldung ERROR CURRENT

▶ 4.2.39 [□ 56].

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - maxMeasureCurrent

### **ACHTUNG**

**Eine falsche Eingabe kann zur Zerstörung der Elektronik führen!**

---

### **Nom. Strom**

PARAMETER CODE: 0x7D23

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Strom der dauerhaft durch die Phasen des Motors fließen darf. Wird dieser längere Zeit überschritten, wird ein I<sup>2</sup>T Fehler ▶ 4.2.38 [□ 56] ausgelöst.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - maxCur

### **ACHTUNG**

**Eine falsche Eingabe kann zur Zerstörung der Elektronik führen!**

---

**Max. Geschwindigkeit**

PARAMETER CODE: 0x7D24

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximal zulässige Geschwindigkeit des Systems (antriebseitig).

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Max. Beschleunigung**

PARAMETER CODE: 0x7D25

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximale zulässige Beschleunigung des Systems (abtriebseitig).

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Max. Ruck**

PARAMETER CODE: 0x7D26

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*BESCHREIBUNG: Maximaler zulässiger Ruck des Systems (abtriebsseitig). Der Ruck ist die zeitliche Änderung der Beschleunigung. Dieser Parameter wird nur ausgewertet, wenn *eine Positionsfahrt mit Ruckbegrenzung* ▶ 5.3.2.5.11 [□ 93] ausgeführt wird.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Polpaare**

PARAMETER CODE: 0x7D27

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Elektrische Polpaare des Motors. Wird nur bei bürstenlosen DC Motoren benötigt. Hat Auswirkungen auf Berechnung von Geschwindigkeiten, Positionen und Kommutierungsmuster.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

### **Anschlußwiderstand**

PARAMETER CODE: 0x7D28

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Anschlusswiderstand wird für Testfunktionen zur Begrenzung von maximalen Strömen benötigt, sowie für die automatische Reglerkonfiguration.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

### **ACHTUNG**

**Eine falsche Eingabe kann zur Zerstörung der Elektronik führen!**

---

### **Induktivität**

PARAMETER CODE: 0x7D29

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Anschlussinduktivität wird für automatische Reglerkonfiguration benötigt.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

### **ACHTUNG**

**Eine falsche Eingabe kann zur Zerstörung der Elektronik führen!**

---

### **Motorkonstante**

PARAMETER CODE: 0x7D2A

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Die Motorkonstante wird für Systemeinstellungen benötigt.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Kommutiertabelle**

PARAMETER CODE: 0x7D2B

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Für die Block-Kommutierung mit Hilfe von Hallgeber wird hier die für die Einheit gültige Hallgeber-Tabelle eingestellt. Bei fehlerhaftem Eintrag bewegt sich der Motor gar nicht oder erzeugt sehr wenig Moment.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT8) 0 - 5

**Strommessbereich**

PARAMETER CODE: 0x7D2C

Zugriffsrechte lesen - schreiben:

*USER - SCHUNK*

BESCHREIBUNG: Maximaler Strommessbereich des intern eingesetzten Stromsensors.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Max. Messdifferenz**

PARAMETER CODE: 0x7D2D

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: 3. Stromsensor notwendig. Maximal erlaubte Abweichung nach  $A + B + C = 0$ . Wenn Abweichung größer als eingestellter Wert wird ein *ERROR MOTOR PHASE*

▶ [4.2.26 \[ 54\]](#) erzeugt.

Nur schreibbar wenn Motor stromlos.

---

**HINWEIS**

**Der Wert "-1" deaktiviert den dritten Stromsensor (nicht von allen Hardwarevarianten unterstützt). => Ein Motor-Phasen-Bruch bzw. Motor-Phasen-Kurzschluss wird nicht erkannt.**

---

**HINWEIS**

**Wird der dritte Stromsensor aktiviert ist eine Kalibrierung ▶ [3.3.1 \[ 42\]](#) der Stromsensoren notwendig.**

---

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

### **Offset Phase A**

PARAMETER CODE: 0x7D2E

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Über *Kalibrierassistent* ▶ 3.3.1 [42] mit Advanced Rechten schreibbar. Nullpunktableich des ersten Stromsensors. Dieser Wert sollte sich in einem Bereich von 1700 - 2200 bewegen. Ansonsten besteht der Verdacht auf eine defekte Hardware.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT16) 1500 - 2600

---

### **HINWEIS**

**Ein falscher Wert kann zu unvorhersehbaren Verhalten des Antriebs führen (Fährt nur in eine Richtung, ruckelt stark, geht durch).**

---

### **Offset Phase B**

PARAMETER CODE: 0x7D2F

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Über *Kalibrierassistent* mit Advanced Rechten schreibbar. Nullpunktabgleich des zweiten Stromsensors. Dieser Wert sollte sich in einem Bereich von 1700 - 2200 bewegen. Ansonsten besteht der Verdacht auf eine defekte Hardware.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT16) 1500 - 2600

---

### **HINWEIS**

**Ein falscher Wert kann zu unvorhersehbaren Verhalten des Antriebs führen (Fährt nur in eine Richtung, ruckelt stark, geht durch).**

---

**Offset Phase C**

PARAMETER CODE: 0x7D30

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Über *Kalibrierassistent* ▶ 3.3.1 [42] mit Advanced Rechten schreibbar. Nullpunktgleich des dritten Stromsensors. Dieser Wert sollte sich in einem Bereich von 1700 - 2200 bewegen. Ansonsten besteht der Verdacht auf eine defekte Hardware.

Nur schreibbar wenn Motor stromlos.

Sollte die Hardware keinen dritten Stromsensor unterstützen wird dieser Wert immer zu 2048 geschrieben.

Daten Typ: (UINT16) 1500 - 2600

**HINWEIS**

**Ein falscher Wert kann zu unvorhersehbaren Verhalten des Antriebs führen (Fährt nur in eine Richtung, ruckelt stark, geht durch).**

**5.3.2.3 Regler****KR Strom**

PARAMETER CODE: 0x7D4B

Zugriffsrechte lesen - schreiben:

USER - PROFI

BESCHREIBUNG: Proportionalanteil des Stromreglers. Bei *Strombegrenzungsregelung* ▶ 5.3.2.3.15 [80] ist dies der Proportionalanteil für die Strombegrenzungsregelung

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**TN Strom**

PARAMETER CODE: 0x7D4C

Zugriffsrechte lesen - schreiben:

USER - PROFI

BESCHREIBUNG: Integralanteil des Stromreglers. Wird bei *Strombegrenzungsregelung* ▶ 5.3.2.3.15 [80] nicht benötigt.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**TD Strom**

PARAMETER CODE: 0x7D4D

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Differentialanteil des Stromreglers.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**KC Strom**

PARAMETER CODE: 0x7D4E

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Korrekturfaktor des Stromreglers für Integralanteil (Anti-windup).

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**KR Geschwindigkeit**

PARAMETER CODE: 0x7D4F

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Proportionalanteil des Geschwindigkeitsreglers

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**TN Geschwindigkeit**

PARAMETER CODE: 0x7D50

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Integralanteil des Geschwindigkeitsreglers

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**TD Geschwindigkeit**

PARAMETER CODE: 0x7D51

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Differentialanteil des Geschwindigkeitsreglers.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**KC Geschwindigkeit**

PARAMETER CODE: 0x7D52

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Korrekturfaktor des Stromreglers für Integralanteil (Anti-windup).

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**KR Position**

PARAMETER CODE: 0x7D53

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Proportionalanteil des Positionsreglers.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**TN Position**

PARAMETER CODE: 0x7D54

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Integralteil des Positionsreglers.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**TD Position**

PARAMETER CODE: 0x7D55

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Differentialanteil des Positionsreglers.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**KC Position**

PARAMETER CODE: 0x7D56

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Korrekturfaktor des Stromreglers für Integralanteil (Anti-windup).

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Positionsabweichung**

PARAMETER CODE: 0x7D59

Zugriffsrechte lesen - schreiben:

USER - PROFI

BESCHREIBUNG: Positionsfenster, in dem die Positionsregelung abgebrochen wird (je nach *Bremsenkonfiguration*

► 5.3.2.7.3 [96], wird weitergeregelt oder die Bremse fällt ein bzw. es wird Position erreicht gemeldet.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Max. Überschwinger**

PARAMETER CODE: 0x7D5A

Zugriffsrechte lesen - schreiben:

USER - ADVANCED

BESCHREIBUNG: Fährt das Modul bei einer Positionsfahrt über dieses Positionsfenster, so wird die Fehlermeldung *ERROR*

*OVERSHOOT* ► 4.2.60 [60] erzeugt. Dieser Wert muss größer der maximal erlaubten *Positionsabweichung*

► 5.3.2.3.13 [80] eingestellt werden.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

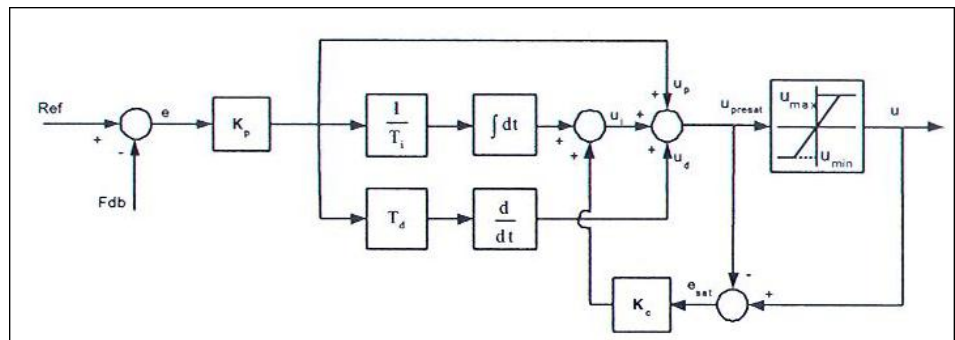
**Struktur**

PARAMETER CODE: 0x7D5B

Zugriffsrechte lesen - schreiben:

PROFI - PROFI

BESCHREIBUNG:



Regler Struktur

Alle Regler sind als PID-Regler mit Anti-windup Funktionalität ausgeführt. Der komplette Parametersatz ist nur über Root Rechte im SCHUNK Modus zu erreichen.

Folgende Parameter sind einstellbar:

- KR: Proportionalanteil des jeweiligen Reglers
- TN: Integralanteil des jeweiligen Reglers
- TD: Differentialanteil des jeweiligen Reglers

- KC: Korrekturfaktor für Integralanteil
- Stromgeschwindigkeit (0x00)  
Stromregelung und Geschwindigkeitsregelung funktionieren unabhängig voneinander.
- Kaskade (0x01)  
Positions-, Geschwindigkeit- und Stromregler sind in Kaskade geschaltet => Es sind stromgeregelte (eingestellter Strom wird nicht überschritten) Positions-, bzw. Geschwindigkeitsfahrten möglich (z.B. kein Vorpositionieren bei einem Greifvorgang notwendig). In diesem Modus wird der vorgegebene Strom ("SET TARGET CURENT" ▶ 3.1.8 [□ 32]) bei allen Verfahrarten nicht überschritten.
- Geschwindigkeit mit Strombegrenzung (0x02)  
Stromregelung ist nicht aktiv. Bei Geschwindigkeits- bzw. Positionsfahrten wird der vorgegeben Strom ("SET TARGET CURRENT" ▶ 3.1.8 [□ 32]) begrenzt. Hierbei wird im Gegensatz zur Kaskade der Strom nicht geregelt, sondern begrenzt (Strombegrenzungsregelung).
- Geschwindigkeit mit PWM Begrenzung (0x02)  
Stromregelung ist nicht aktiv. Bei Geschwindigkeit- bzw. Positionsfahrten wird das Tastverhältnis der PWM begrenzt. Das Verhältnis von Strom zu Tastverhältnis wird über den *Anschlusswiderstand* ▶ 5.3.2.2.11 [□ 74] des Motors berechnet.

---

### HINWEIS

**Da das Tastverhältnis der PWM direkt begrenzt wird (Spannungsbegrenzung), ist es möglich, dass der Antrieb nicht mehr seine volle Geschwindigkeit erreicht, daher können Positionsfahrten eventuell erheblich länger dauern als vorausberechnet.**

---

### HINWEIS

**Bei einer Änderung der Regelstruktur müssen evtl. die Regelparameter angepasst werden!**

---

Daten Typ: (ENUM) ▶ 7.1 [□ 113] CTRL\_CURRENT\_SPEED - MAX\_CTRL\_MODE-1

### 5.3.2.4 Referenzierung

#### Typ

PARAMETER CODE: 0x7D41

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: eReferenceType

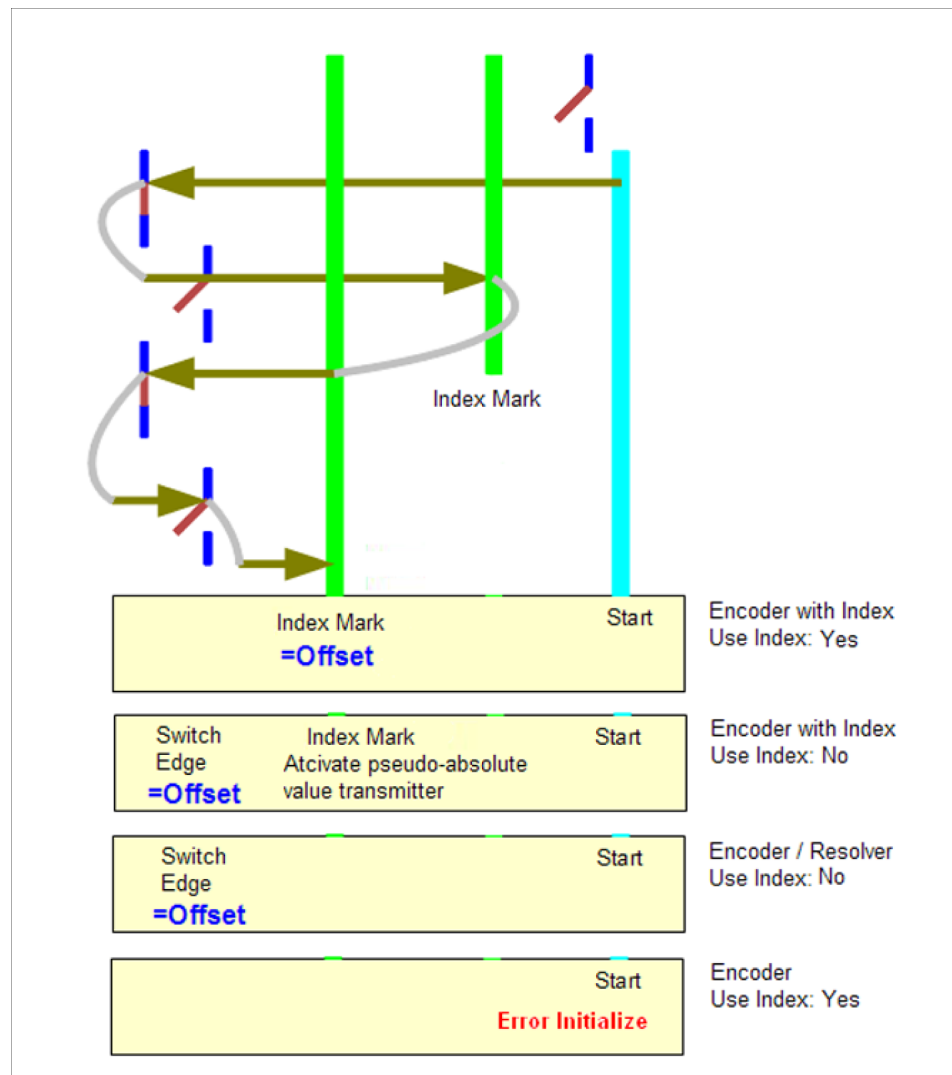
Die Art der Referenzierung kann festgelegt werden. Bitte Abschnitt Positionierung Type ▶ 5.3.2.5.2 [ 90] bei Einsatz von Encoder mit Indexspur beachten.

Interner Schalter links (0x00) / rechts (0x01)

Der Interne Referenzschalter wird zur Referenzierung herangezogen. Die Bewegungsrichtung bei aktivem Referenzschalter wird über Richtung "links" bzw. "rechts" bestimmt.

Externer Schalter IN1 links (0x02) / rechts (0x03)

Ein externer Referenzschalter (IN1) wird zur Referenzierung herangezogen. Die Bewegungsrichtung bei aktivem Referenzschalter wird über die Richtung "links" bzw. "rechts" bestimmt.



Referenzierung mit Schalter

**HINWEIS**

Es ist sicherzustellen, dass die Schaltflanken des Näherungsschalter für mind. 200ms anliegt.

(Referenziergeschwindigkeit anpassen, Schaltnocken anpassen)

**HINWEIS**

Es wird empfohlen nach Einbau in der Applikation eine "Grundreferenzierung" durchzuführen. Es kann ebenfalls eine "Grundreferenzierung" notwendig werden, wenn die Lage oder Belastung bei der Referenzierung geändert wird. Es kann sonst gehäuft zu einem **ERROR REFERENCED** ▶ 4.2.5 [ 51] kommen

Geschwindigkeit links (0x04 / rechts (0x05)

Zur Referenzierung wird eine Geschwindigkeitsfahrt ausgeführt. Fährt das Modul auf einen festen Anschlag, wird dieser als Referenzpunkt erkannt. Die Drehrichtung wird über "links" bzw. "rechts" festgelegt.

---

### HINWEIS

**Diese Art von Referenzierung ist nur empfehlenswert, wenn ein fester Endanschlag vorhanden ist!**

---

Geschwindigkeit mit Hubüberwachung links (0x06) / rechts (0x07)  
Zusätzlich zu oben genannter Vorgehensweise wird nach Anfahren des ersten festen Anschlages zum gegenüberliegenden festen Anschlag gefahren. Der zurückgelegte Weg muss größer als die Differenz der *Softwareendanschläge* ▶ 5.3.2.1.8 [□ 68] sein => Referenzierung erfolgreich

---

### HINWEIS

**Diese Art von Referenzierung ist nur empfehlenswert, wenn ein fester Endanschlag vorhanden ist!**

---

Strom links (0x08) / rechts (0x09)  
Es wird eine Stromfahrt ausgeführt. Der Strom wird solange erhöht bis sich das Modul bewegt. Überschreitet der Strom den *max. Referenzstrom* ▶ 5.3.2.4.5 [□ 88], wird davon ausgegangen, dass ein fester Anschlag erreicht worden ist, der als Referenzpunkt erkannt wird.

---

### HINWEIS

**Diese Art von Referenzierung ist nur empfehlenswert, wenn ein fester Endanschlag vorhanden ist!**

---

### HINWEIS

**Eine Verklemmung, Schwergängigkeit in der Mechanik oder ein "vergessenes" Werkstück kann ebenfalls zu einer Überschreitung des Nennstroms führen. Dies wird dann ebenfalls als fester Endanschlag interpretiert, obwohl keiner vorhanden ist.**

---

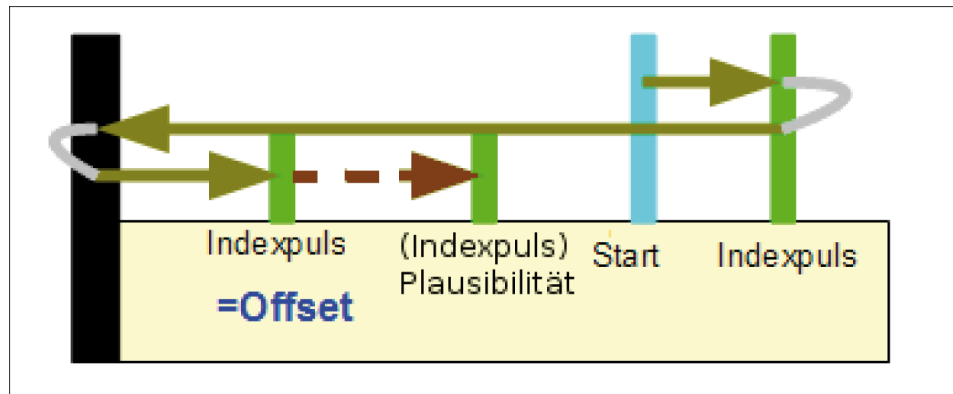
Strom mit Hubüberwachung links (0x0A) / rechts (0x0B)  
Zusätzlich zu oben genannter Vorgehensweise wird nach Anfahren des ersten festen Endanschlages zum gegenüberliegenden festen Endanschlag gefahren. Der zurückgelegte Weg muss größer als die Differenz der *Softwareanschläge* ▶ 5.3.2.1.8 [□ 68] sein => Referenzierung erfolgreich.

---

### HINWEIS

**Diese Art von Referenzierung ist nur empfehlenswert, wenn ein fester Endanschlag vorhanden ist!**

---



Referenzierung auf Anschlag mit aktivierter Indexspur

Keine (0x0C)

Nach Absenden des Befehls *CMD REFERENCE* wird die aktuelle Position als Referenzposition angesehen.

Daten Typ: (ENUM) ▶ 7.1 [113] REF\_SWITCH\_INTERN\_LEFT - MAX\_REFTYPE-1

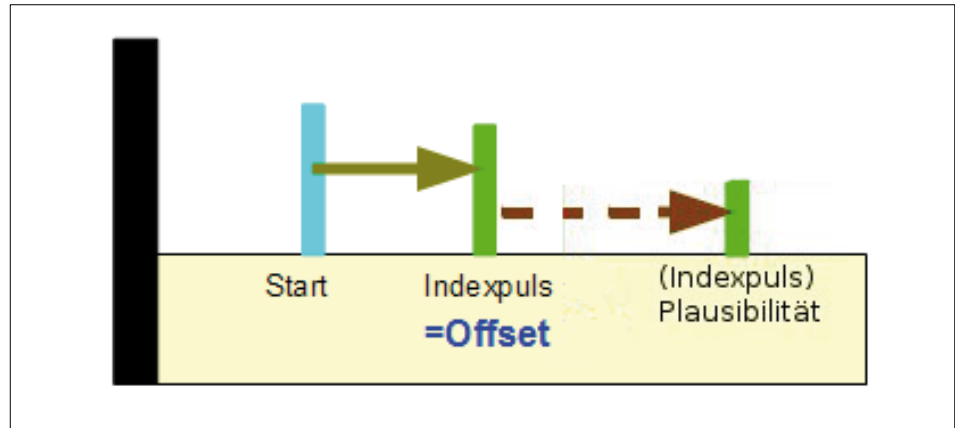
**Nutzung Index**

PARAMETER CODE: 0x7D42

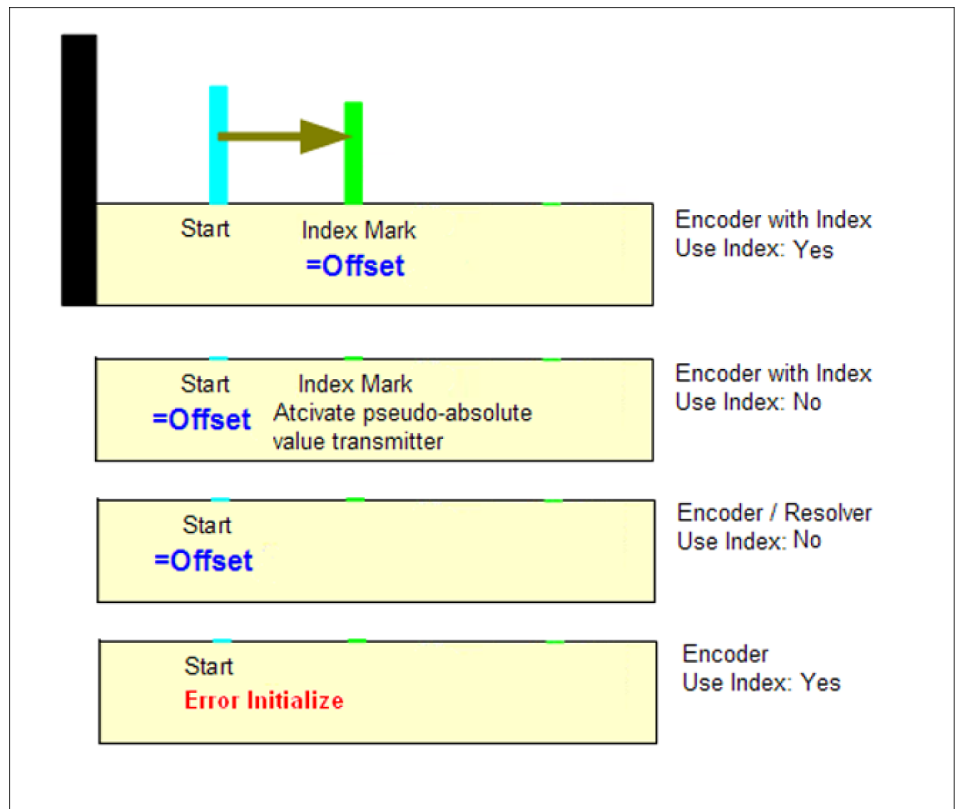
Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Gibt an ob die Indexspur eines Encoders bei der Referenzierung ausgewertet werden soll.



Referenzierung "Keine" mit aktivierter Indexspur



Referenzierung "Keine"

**HINWEIS**

Soll mit Indexpuls referenziert werden. Liegt der Indexpuls an einer ungünstigen Stelle, kann es vorkommen, dass bei mehrmaliger Referenzierung die Positionen jeweils um eine Motordrehung abweichen. Abhilfe: Referenzmarke leicht verschieben. Dies gilt für alle Referenzmarken außer "Interner Schalter" und "Externer Schalter".

---

Daten Typ: (BOOL) false - true

**Abstand zum Index**

PARAMETER CODE: 0x7D43

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Ist der Parameter "Abstand zum Index" zu "0" gesetzt wird während der Referenzfahrt der Abstand vom Referenzereignis (Schalterflanke dedektiert) zum Indexpuls gemessen und abgespeichert. Bei nachfolgenden Referenzfahrten wird dieser Abstand erneut ermittelt und mit dem gespeicherten Wert verglichen. Liegen die Werte innerhalb eines festgelegten *Toleranzfensters* ▶ 5.3.2.4.10 [📄 89] kann die Referenzierung erfolgreich abgeschlossen werden. Des weiteren kann ein "ungünstig" liegender Indexpuls (Indexpuls kur vor bzw. hinter Schalterflanke) "korrigiert" werden.

---

**HINWEIS**

Wurde der Abstand zwischen Indexpuls und Referenzereignis über die zulässige Toleranz ▶ 5.3.2.4.10 [📄 89] hinaus geändert, wird die Referenzierung mit **ERROR REFERENCED** ▶ 4.2.5 [📄 51] abgebrochen. Eine neue Vermessung des Abstandes vom Referenzereignis zum Indexpuls ist notwendig. Hierzu muss der Parameter **distanceFromIndex** ▶ 5.3.2.4.3 [📄 87] zu "0" gesetzt werden.

---

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

### **Nach Refer. 0 anfahren**

PARAMETER CODE: 0x7D44

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Nach erfolgreicher Referenzierung wird die Position "0.0" angefahren.

Daten Typ: (BOOL) false - true

### **Max. Referenzstrom**

PARAMETER CODE: 0x7D45

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Stromvorgabe in [%] vom *Nennstrom des Motors* ▶ 5.3.2.4.5 [□ 88]. Der Referenzstrom überschreitet den vorgegebenen Wert nicht.

Daten Typ: (UINT8) 0 - 200

### **Geschwindigkeit**

PARAMETER CODE: 0x7D46

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Geschwindigkeitsvorgabe für Referenzfahrten mit internem bzw. externem Referenzschalter und Geschwindigkeitsreferenzfahrten.

Daten Typ: (FLOAT) 0 - maxVel

### **Beschleunigung**

PARAMETER CODE: 0x7D47

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Beschleunigungsvorgabe für Referenzfahrten mit internem bzw. externem Referenzschalter und Geschwindigkeitsreferenzfahrten.

Daten Typ: (FLOAT) 0 - maxAcc

**Offset**

PARAMETER CODE: 0x7D48

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Positionsoffset nach erfolgreicher Referenzierung (Nullpunktverschiebung).

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**Timeout**

PARAMETER CODE: 0x7D49

Zugriffsrechte lesen - schreiben:

*USER - PROFI*BESCHREIBUNG: Zeit, welche die Referenzfahrt maximal dauern darf. Bei Überschreitung der Zeit wird der Antrieb stromlos geschaltet und es erfolgt eine *Fehlermeldung*

▶ 4.2.5 [□ 51]

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**max. Distanz Schalter**

PARAMETER CODE: 0x7D4A

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximal erlaubter Abstand vom Referenzereignis (Schalterflanke dedektiert) zum Indexpuls. Wird ein Encoderticks angegeben. Siehe auch ▶ 5.3.2.4.3 [□ 87].

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.2.5 Positionierung****Seriennummer**

PARAMETER CODE: 0x7D5F

Zugriffsrechte lesen - schreiben:

*USER - SCHUNK*

BESCHREIBUNG: Seriennummer des Positionsmesssystems.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

### Typ

PARAMETER CODE: 0x7D60

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Der Typ des Messsystems wird festgelegt.

Nur schreibbar wenn Motor stromlos.

Ein Neustart des Moduls ist erforderlich.

- Encoder  
Encoder Messsystem ohne Indexspur.
- Encoder Index  
Encoder mit Indexspur. Bei der *Referenzfahrten*  
▶ 5.3.2.4.1 [ 82] wird je nach *Konfiguration*  
▶ 5.3.2.4.2 [ 86] die Indexspur ausgewertet. Um sicherzugehen, dass eine Indexspur am richtigen Ort gefunden werden kann, kann es bei einigen Referenziersarten vorkommen, dass sich der Antrieb einige Male mit kurzen Bewegungen hin und her bewegt, bzw. kleine Bewegungen in die "falsche" Richtung macht. Die Module bewegen sich auch bei Referenzierart "Keine", da nach dem nächsten Indexpuls gesucht wird. Siehe auch *Pseudoabsolutwertgeber* ▶ 1.8 [ 19].
- Resolver  
Resolversystem bei dem der Erregerstrom einstellbar ist.  
*Amplitude* ▶ 5.3.2.5.5 [ 91] und *Frequenz*  
▶ 5.3.2.5.6 [ 92] beachten.

---

### HINWEIS

Bei einem Motor mit Resolver kann es beim "*Positioning ramp type*" ▶ 5.3.2.5.11 [ 93] "Ruckbegrenzt" zum "Rattern" kommen. In diesem Fall muss ein anderer "*Positioning ramp type*" ▶ 5.3.2.5.11 [ 93] gewählt werden.

---

### HINWEIS

Ein Neustart des Moduls ist erforderlich

---

- Encoder differenziell  
Differentieller Encoder ohne Indexspur

- Encoder Index differenziell  
differentieller Encoder mit Indexspur. Bei Referenzfahrten wird je nach *Konfiguration* ▶ 5.3.2.4.2 [□ 86] die Indexspur ausgewertet. Um sicherzugehen, dass eine Indexspur am richtigen Ort gefunden werden kann, kann es bei einigen Referenzierarten vorkommen, dass sich der Antrieb einige Male mit kurzen Bewegungen hin und her bewegt bzw. kleine Bewegungen in die "falsche" Richtung macht. Die Module bewegen sich auch bei Referenzierart "Keine", da nach dem nächsten Indexpuls gesucht wird. Siehe auch ▶ 1.8 [□ 19]

Daten Typ: (ENUM) ▶ 7.1 [□ 113] POS\_SYSTEM\_ENCODER - MAX\_POS\_SYSTEM\_TYPE-1

### Einbaulage

PARAMETER CODE: 0x7D61

Zugriffsrechte lesen - schreiben:

USER - ADVANCED

BESCHREIBUNG: Die Einbaulage des Positionsmesssystems.

- Antriebseitig: Das Positionsmesssystem ist direkt auf dem Antrieb montiert.
- Abtriebseitig: Das Positionsmesssystem ist direkt auf dem Abtrieb montiert.
- Zwischen Getriebeübersetzungen: Das Positionsmesssystem ist mitten im Getriebe montiert.

Daten Typ: (ENUM) ▶ 7.1 [□ 113] MOUNT\_INPUT\_SIDE - MAX\_POS\_MOUNT-1

### Ticks pro Umdrehung

PARAMETER CODE: 0x7D62

Zugriffsrechte lesen - schreiben:

USER - ADVANCED

BESCHREIBUNG: Ticks pro Umdrehung (4fach Auswertung).

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT16) 512 - MAX\_UINT16

### Erreger Amplitude

PARAMETER CODE: 0x7D63

Zugriffsrechte lesen - schreiben:

USER - ADVANCED

BESCHREIBUNG: Amplitude der Eingangsspannung an der Erregerspule [%]. Muss messtechnisch ermittelt werden. Die Ausgangsspannung an den Empfängerspulen darf nicht in die Sättigung gehen.

Daten Typ: (UINT8) 0 - 100

### Erreger Frequenz

PARAMETER CODE: 0x7D64

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Frequenz der Spannung an der Erregerspule in [kHz]. Erlaubte Werte: 8 [kHz], 4 [kHz], 2 [kHz], 1 [kHz].

Nur schreibbar wenn Motor stromlos.

Daten Typ: (ENUM) ▶ 7.1 [□ 113] RESOLVER\_FRQ\_1kHz - MAX\_RESOLVER\_FRQ-1

### ADC Offset

PARAMETER CODE: 0x7D65

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: ADC Offset um das Eingangssignal zu "zentrieren". Wird nur für Cos/Sin Messsysteme benötigt und Resolver.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

### Offset

PARAMETER CODE: 0x7D68

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Verdrehung des Positionsmesssystem gegenüber den Motorphasen. Wird im voreingestellten *Einheitensystem* ▶ 2 [□ 22] konfiguriert. Dieser Wert wird unter Umständen automatisch ermittelt. Siehe hierzu Stillstandkommutierung ▶ 1.9 [□ 21]. Wird dieser Wert zu "0" gesetzt wird der "*Sinus-Zeiger*" neu gesucht. Die *Referenzierung* wird hierbei gelöscht.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) - MAX\_FLOAT - MAX\_FLOAT

---

### HINWEIS

**Zur Zeigersuche sollte das Modul in alle Richtungen frei beweglich sein. Modul wird ruckartig bis zu zwei Motorumdrehungen bewegt. Eine Kommunikation mit dem Modul ist in dieser Zeit nicht möglich. Dieser Vorgang kann bis zu 30sek. benötigen.**

---

**Bewegungsschwelle**

PARAMETER CODE: 0x7D69

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Der Wert in [%] von der *Höchstgeschwindigkeit* ▶ 5.3.2.2.7 [□ 73]. Bei Unterschreitung wird das Modul als stehend behandelt. Siehe *Motion Tool Schunk Statusanzeige* (siehe Softwareanleitung "Motion Tool Schunk").

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 1 - 100

**Wartezeit Position erreicht**

PARAMETER CODE: 0x7D6A

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Nur gültig für Schunk Drive Protocol (SDP) für Profibus/PROFINET. Hiermit kann das Rücksetzen des Flags "*Position erreicht*" für eine definierte Zeit in [sec.] verzögert werden. Es wird empfohlen diesen Wert etwas größer als den SPS Zyklustakt einzustellen.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**Positionsrampe**

PARAMETER CODE: 0x7D6B

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

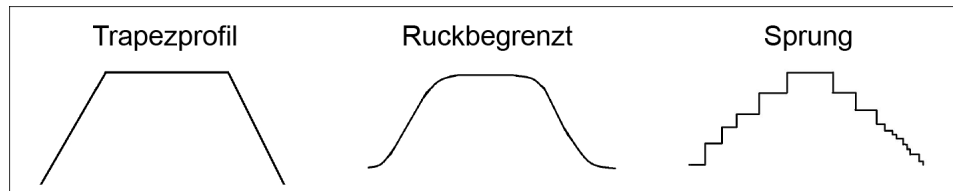
BESCHREIBUNG: Rampentyp für Positionsfahrt wird eingetragen.

Nur schreibbar wenn Motor stromlos.

- Trapez  
Für die Berechnung des Verfahrprofils wird ein Trapez zu Grunde gelegt.
- Ruckbegrenzt (0x01)  
Für Positionsfahrt "*MOVE POS*" ▶ 3.1.2 [□ 25] und "*MOVE POS REL*" ▶ 3.1.3 [□ 27] wird eine Bahn mit Ruckbegrenzung berechnet. Bei diesem Rampentyp wird der Parameter "Ruckbegrenzt" ("*SET TARGET JERK*" ▶ 3.1.7 [□ 31]) verwendet.
- Trapez SRU  
Für die Berechnung des Verfahrprofils wird ein Trapez zu Grunde gelegt. Eine Berechnung der Verfahrzeit erfolgt nicht. (Umschaltpunkte werden nach Positionen gesteuert)

- Sprung  
Hier wird kein Bahnprofil berechnet sondern direkt der Positionssprung vorgegeben. Die interne Bahnplanung ist abgeschaltet. Je nach Interpolationsakt des externen Interpolators kann es notwendig sein die *Regelparameter* ▶ 5.3.2.3.4 [113] anzupassen.

Daten Typ: (ENUM) ▶ 7.1 [113] RAMP\_TRAPEZ - MAX\_RAMP\_MODE-1



### Schleppfehler

PARAMETER CODE: 0x7D6C

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Schleppfehler. Dieser Wert darf während einer Positionsfahrt nicht überschritten werden. Bei Überschreitung kommt es zu einer Fehlermeldung ("ERROR TOW" ▶ 4.2.34 [55]).

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

### Wartezeit Bewegung blockiert

PARAMETER CODE: 0x7D6D

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Zeit in [sec], welche verstreichen muss um das Flag bzw. Kommando *CMD MOVE BLOCKED* ▶ 3.2.2 [37] auszulösen. Bei konfigurierter *Bremse* ▶ 5.3.2.7.2 [96] und "*normaler*" ▶ 5.3.2.7.3 [96] Bremsennutzung, darf dieser Wert nicht kleiner als der halbe *BremsenTimeout* ▶ 5.3.2.7.4 [97] sein. (Die "Blockierererkennung" greift ansonsten direkt nach dem Start eines Bewegungsbefehls.)

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

### 5.3.2.6 Getriebe

#### **Seriennummer**

PARAMETER CODE: 0x7D37

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Seriennummer des Getriebes.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

#### **Übersetzung 1**

PARAMETER CODE: 0x7D38

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Getriebeübersetzung bzw. - untersetzung vom Motor zum Abtrieb.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

#### **Übersetzung 2**

PARAMETER CODE: 0x7D39

Zugriffsrechte lesen - schreiben:

*USER - PROFI*

BESCHREIBUNG: Getriebeübersetzung bzw. - untersetzung vom Positionsmesssystem zum Abtrieb. Wird nur benötigt, wenn das *Positionsmesssystem* ▶ 5.3.2.5.3 [91] zwischen Getriebeübersetzung eingebaut ist.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

### 5.3.2.7 Bremse

#### **Seriennummer**

PARAMETER CODE: 0x7D3C

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Seriennummer der Bremse.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

### Typ

PARAMETER CODE: 0x7D3D

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Typ der Bremse. Über die konfigurierte Motorspannung und den Bremsentyp wird die Bremsenspannung per Software angepasst.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (ENUM) ▶ 7.1 [113] BRAKE\_TYPE\_NONE - MAX\_BRAKETYPE-1



### ⚠ GEFÄHR

#### **Schwere Verletzung oder Tod durch nicht ordnungsgemäß wirkende Bremse möglich**

Eine Fehlkonfiguration führt zu einer nicht funktionstüchtigen Bremse

- Fehlerfreiheit der Konfiguration überprüfen

---

### Bremsen Nutzung

PARAMETER CODE: 0x7D3E

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Benutzung der Bremse.

Nur schreibbar wenn Motor stromlos.

- Wird nicht genutzt (0x00)  
Bremse wird nicht benutzt. Fällt nur bei Spannungsausfall ein. Wird sofort nach dem Starten des Moduls einmalig geöffnet.
- Nur im Fehlerfall (0x01)  
Bremse fällt nur im Fehlerfall ein und wird beim ersten Bewegungsbefehl geöffnet. Ansonsten wird der Antrieb dauerhaft geregelt.
- Normal (0x02)  
Bremse fällt im Fehlerfall und am Ende der Bewegung ein

Ist eine Bremse konfiguriert, ist der *Pseudoabsolutwertgeber* ▶ 1.8 [19] möglicherweise aktiv, falls weitere Bedingungen erfüllt sind.

Daten Typ: (ENUM) Auflistung der "ENUMs" BRAKE\_USE\_NO USE - MAX\_BRAKE\_USAGE-1

**Timeout**

PARAMETER CODE: 0x7D3F

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Zeit, welche die Bremse benötigt das Magnetfeld auf- bzw. abzubauen.

Nur schreibbar wenn Motor stromlos.

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

**5.3.2.8 Spannung****Min. Motor Spannung**

PARAMETER CODE: 0x7D32

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*BESCHREIBUNG: Minimal erlaubte Motorspannung. Bei Unterschreitung dieses Wertes wird ein *Unterspannungsfehler* ▶ 4.2.55 [📄 59] ausgelöst.

Daten Typ: (FLOAT) 10 - motorMaxVolt

**Max. Motor Spannung**

PARAMETER CODE: 0x7D33

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*BESCHREIBUNG: Maximal erlaubte Motorspannung. Bei Überschreitung dieses Wertes wird ein *Überspannungsfehler* ▶ 4.2.56 [📄 59] ausgelöst. Tritt dieser Fehler wiederholt auf, wird das Modul gesperrt und ist nur durch einen Servicemitarbeiter wieder in Betrieb zu nehmen.

Daten Typ: (FLOAT) motorMinVolt - 72

---

**HINWEIS****evtl. sollte ein externer Bremshopper eingesetzt werden.**

---

**Min. Logik Spannung**

PARAMETER CODE: 0x7D34

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Minimal erlaubte Logikspannung. Bei Unterschreitung dieses Wertes wird ein **Unterspannungsfehler ▶ 4.2.53 [□ 58]** ausgelöst.

Daten Typ: (FLOAT) 5 - logicMaxVolt

**Max. Logik Spannung**

PARAMETER CODE: 0x7D35

Zugriffsrechte lesen - schreiben:

*USER - ADVANCED*

BESCHREIBUNG: Maximal erlaubte Logikspannung. Bei Überschreitung dieses Wertes wird ein **Überspannungsfehler ▶ 4.2.54 [□ 58]** ausgelöst.

Daten Typ: (FLOAT) logicMinVolt - 30

**5.3.2.9 Info**

**EEPROM Version**

PARAMETER CODE: 0x7D9B

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Versionsnummer des EEPROM-Datensatzes.

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**EEPROM CRC**

PARAMETER CODE: 0x7D9C

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Checksumme über alle EEPROM-Daten.

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**Daten CRC**

PARAMETER CODE: 0x7D9D

Zugriffsrechte lesen - schreiben:

*USER - ROOT*

BESCHREIBUNG: Checksumme über alle EEPROM-Daten welche nicht Modul spezifisch sind.

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**Fehlercode 0**

PARAMETER CODE: 0x7DA0

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Zuletzt aufgetretenen Fehler. (n-ter Fehler)

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 1**

PARAMETER CODE: 0x7DA1

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Vorletzter aufgetretener Fehler. (n-1 Fehler)

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 2**

PARAMETER CODE: 0x7DA2

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: Vor-Vorletzter aufgetretener Fehler. (n-2 Fehler)

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 3**

PARAMETER CODE: 0x7DA3

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-3 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 4**

PARAMETER CODE: 0x7DA4

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-4 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 5**

PARAMETER CODE: 0x7DA5

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-5 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 6**

PARAMETER CODE: 0x7DA6

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-6 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 7**

PARAMETER CODE: 0x7DA7

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-7 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 8**

PARAMETER CODE: 0x7DA8

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-8 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 9**

PARAMETER CODE: 0x7DA9

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-9 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 10**

PARAMETER CODE: 0x7DAA

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-10 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 11**

PARAMETER CODE: 0x7DAB

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-11 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 12**

PARAMETER CODE: 0x7DAC

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-12 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 13**

PARAMETER CODE: 0x7DAD

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-13 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 14**

PARAMETER CODE: 0x7DAE

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-14 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 15**

PARAMETER CODE: 0x7DAF

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-15 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 16**

PARAMETER CODE: 0x7DB0

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-16 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 17**

PARAMETER CODE: 0x7DB1

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-17 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 18**

PARAMETER CODE: 0x7DB2

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-18 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**Fehlercode 19**

PARAMETER CODE: 0x7DB3

Zugriffsrechte lesen - schreiben:

*PROFI - ROOT*

BESCHREIBUNG: n-19 Fehler.

Daten Typ: (UINT8) MIN\_UINT8 - MAX\_UINT8

**5.3.3 Asynchron****5.3.3.1 Aktuelle Temperatur Hauptplatine**

PARAMETER CODE: 0x7DB9

Zugriffsrechte lesen - schreiben:

*PROFI - Disabled*

BESCHREIBUNG: gemessene Temperatur des ersten Temperatursensors.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**5.3.3.2 Aktuelle Temperatur Motor**

PARAMETER CODE: 0x7DBA

Zugriffsrechte lesen - schreiben:

*PROFI - Disabled*

BESCHREIBUNG: gemessene Temperatur des Motortemperatursensors.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**5.3.3.3 Aktuelle Temperatur Opt. Komm.**

PARAMETER CODE: 0x7DBB

Zugriffsrechte lesen - schreiben:

*PROFI - Disabled*

BESCHREIBUNG: gemessene Temperatur des zweiten Temperatursensors.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.3.4 Fehler Zeile

PARAMETER CODE: 0x7DBC

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Code Zeile in welche der letzte Fehler ausgelöst wurde.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

#### 5.3.3.5 Fehler Wert

PARAMETER CODE: 0x7DBD

Zugriffsrechte lesen - schreiben:

*PROFI - Disabled*

BESCHREIBUNG: Detail welches zum letzten Fehler geführt hat.

(Für Servicezwecke wichtige Information.)

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.3.6 Fehler Datei

PARAMETER CODE: 0x7DBE

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: File Name in welchem der letzte Fehler ausgelöst wurde.

(Für Servicezwecke wichtige Information.)

Daten Typ: (CHAR\_ARRAY) -

#### 5.3.3.7 Firmware Typ

PARAMETER CODE: 0x7DBF

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Software Typ des Moduls

(Für Servicezwecke wichtige Information.)

Daten Typ: (ENUM) ▶ [7.1](#) [[□ 113](#)] SW\_TYPE\_PTA - MAX\_SW\_TYPE-1

**5.3.3.8 Bestellnummer**

PARAMETER CODE: 0x7DC0

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Bestellnummer des Moduls

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.9 Modultyp**

PARAMETER CODE: 0x7DC1

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Modultyp

(Für Servicezwecke wichtige Information.)

Daten Typ: (CHAR\_ARRAY) -

**5.3.3.10 Erstelldatum Firmware Hauptplatine**

PARAMETER CODE: 0x7DC2

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Kompilier-Datum der Software auf Hauptplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.11 Erstellzeit Firmware Hauptplatine**

PARAMETER CODE: 0x7DC3

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Kompilier-Zeit der Software auf Hauptplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.12 Hardware Version Hauptplatine**

PARAMETER CODE: 0x7DC4

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Hardwareversion der Hauptplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.13 Firmware Version Hauptplatine**

PARAMETER CODE: 0x7DC5

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Softwareversion der Hauptplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.14 Erstelldatum Firmware Opt. Komm.**

PARAMETER CODE: 0x7DC6

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Kompilier-Datum der Software auf Kommunikationsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.15 Erstellzeit Firmware Opt. Komm.**

PARAMETER CODE: 0x7DC7

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Kompilier-Zeit der Software auf Kommunikationsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.16 Hardware Version Opt. Komm.**

PARAMETER CODE: 0x7DC8

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Hardwareversion der Kommunikationsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.17 Firmware Version Opt. Komm.**

PARAMETER CODE: 0x7DC9

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Softwareversion der Kommunikationsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.18 Seriennummer Opt. Komm.**

PARAMETER CODE: 0x7DCA

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Seriennummer der Kommunikationsplatine.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.19 Erstelldatum Firmware Opt. 1**

PARAMETER CODE: 0x7DCB

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Kompilier-Datum der Software auf der Erweiterungsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.20 Erstellzeit Firmware Opt. 1**

PARAMETER CODE: 0x7DCC

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Kompilier-Zeit der Software auf Erweiterungsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.21 Hardware Version Opt. 1**

PARAMETER CODE: 0x7DCD

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Hardwareversion der Erweiterungsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.22 Firmware Version Opt. 1**

PARAMETER CODE: 0x7DCE

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Softwareversion der Erweiterungsplatine.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.23 Seriennummer Opt. 1**

PARAMETER CODE: 0x7DCF

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Seriennummer der Erweiterungsplatine.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.24 Erstelldatum Firmware Opt. 2**

PARAMETER CODE: 0x7DD0

Zugriffsrechte lesen - schreiben:

*PROFI - Disabled*

BESCHREIBUNG: Kompilier-Datum der Software auf Erweiterungslatine 2.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.25 Erstellzeit Firmware Opt. 2**

PARAMETER CODE: 0x7DD1

Zugriffsrechte lesen - schreiben:

*PROFI - Disabled*

BESCHREIBUNG: Kompilier-Zeit der Software der Erweiterungslatine 2.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

**5.3.3.26 Hardware Version Opt. 2**

PARAMETER CODE: 0x7DD2

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Hardwareversion der Erweiterungslatine 2.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

**5.3.3.27 Firmware Version Opt. 2**

PARAMETER CODE: 0x7DD3

Zugriffsrechte lesen - schreiben:

*USER - Disabled*

BESCHREIBUNG: Softwareversion der Erweiterungslatine 2.

(Für Servicezwecke wichtige Information.)

Daten Typ: (UINT16) MIN\_UINT16 - MAX\_UINT16

#### 5.3.3.28 **Seriennummer Opt. 2**

PARAMETER CODE: 0x7DD4

Zugriffsrechte lesen - schreiben:

USER - SCHUNK

BESCHREIBUNG: Seriennummer der Erweiterungsplatine 2.

Daten Typ: (UINT32) MIN\_UINT32 - MAX\_UINT32

#### 5.3.3.29 **Motor Spannung**

PARAMETER CODE: 0x7DD6

Zugriffsrechte lesen - schreiben:

USER - Disabled

BESCHREIBUNG: Aktuell gemessene Motorspannung [V].

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

#### 5.3.3.30 **Logik Spannung**

PARAMETER CODE: 0x7DD7

Zugriffsrechte lesen - schreiben:

USER - Disabled

BESCHREIBUNG: Aktuell gemessene Logikspannung [V].

Daten Typ: (FLOAT) 0 - MAX\_FLOAT

#### 5.3.3.31 **Max. Software Endanschlag**

PARAMETER CODE: 0x7DD8

Zugriffsrechte lesen - schreiben:

PROFI - PROFi

BESCHREIBUNG: Temporärer Softwareendanschlag max. Wert.

Hiermit können die Softwareendanschläge geändert werden, ohne diese fest im EEPROM zu speichern.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

#### 5.3.3.32 **Min. Software Endanschlag**

PARAMETER CODE: 0x7DD9

Zugriffsrechte lesen - schreiben:

PROFI - PROFi

BESCHREIBUNG: Temporärer Softwareendanschlag min. Wert.

Hiermit können die Softwareendanschläge geändert werden, ohne diese fest im EEPROM zu speichern.

Daten Typ: (FLOAT) -MAX\_FLOAT - MAX\_FLOAT

**5.3.3.33 User**

PARAMETER CODE: 0x7DDA

Siehe Kapitel „Benutzerverwaltung“ ▶ [1.7 \[18\]](#)

BESCHREIBUNG: Bei lesendem Zugriff wird der aktuelle Nutzer zurückgemeldet. Bei schreibendem Zugriff muss ein String mit dem passenden Passwort übergeben werden.

Daten Typ: (*ENUM*) ▶ [7.1 \[113\]](#) USER\_USER - MAX\_USER-1

## 6 Bemerkungen

### 6.1 Schnell-Test

Sind weder USB-Device noch USB-Host aktiv, kann über die Schalter Test1 und Test2 (siehe Hardwareanleitung) ein Schnell-Test durchgeführt werden. Folgende Kommandos werden unterstützt:

- Test1 - OFF Test2 - OFF : Warte auf neues Kommando
- Test1 - ON Test2 - OFF: *CMD ACK* ▶ 4.1.4 [ 50]
- Test1 - OFF Test2 - ON: Relativfahrt um 1.0[mm] bzw 1.0[Grad] mit 10% maximal Geschwindigkeit, 10% maximal Beschleunigung, 50% nominal Strom und 50% maximal Ruck.
- Test1- ON Test2- ON: *CMD REFERENCE*

Um den Testmodus zu Beenden ist das Modul neu zu starten (Logikspannung abschalten und wieder zuschalten).

### 6.2 Module

<b>Anschlussbeschreibung zum Modul</b>	Da verschiedenste <i>Schunk</i> Module mit verschiedenen Anschlussbelegungen existieren, sehen Sie bitte hierfür in mitgelieferten Dokumentation zu Ihrem Modul nach.
<b>Modul kann nicht aus jeder Lage referenzieren</b>	Der maximal zugelassene Referenzstrom reicht nicht aus um das Modul zu bewegen. Den <i>max. Referenzstrom</i> ▶ 5.3.2.4.5 [ 88] erhöhen.

### 6.3 Protokoll

<b>Fragmentierung ist nicht möglich</b>	Das komplette Modul kann vollständig bedient werden, ohne das eine Fragmentierung der Daten notwendig ist. Dies gilt für alle unterstützten Kommunikationsschnittstellen.
---	---

### 6.4 Profibus

<b>Daten können nicht konsistent übertragen werden</b>	<ol style="list-style-type: none"> <li>1. D-Len (erstes Byte) auf 0x00 setzen, alle Daten auffüllen und zuletzt D-Len auf gewünschten Wert setzen.</li> <li>2. SYNC, FREEZE Mechanismus verwenden.</li> </ol>
--	---

## 7 Anhang

### 7.1 Auflistung der Enumerationen

Wert	MotorType	BrakeType	BrakeUsage
0	MOTOR DC	BRAKE TYPE NONE	BAKE USE NO USE
1	MOTOR BLDC	BRAKE TYPE MAGNETIC 12V	BRAKE USE ERROR ONLY
2	MOTOR PMSM	BRAKE TYPE MAGNETIC 24V	BRAKE USE NORMAL
3	not used	BRAKE TYPE MAGNETIC 48V	MAX BRAKE USAGE
4	not used		
5	MAX MOTOR TYPE		

Wert	ReferenceType	User	Controller Mode
0	REF SWITCH INTERN LEFT	USER	CTRL CURRENT SPEED
1	REF SWITCH INTERN RIGHT	reserved	CTRL CASCADE
2	REF SWITCH EXTERN LEFT	PROFI	CTRL SPEED CUR LIMIT
3	REF SWITCH EXTERN RIGHT	ADVANCED	CTRL SPPED PWM LIMIT
4	REF VEL LEFT	ROOT	CTRL POS CASCADE
5	REF VEL RIGH	reserved	MAX CTL MODE
6	REF VEL LEFT DIST	MAX USER	
7	REF VEL RIGHT DIST		
8	REF CURRENT LEFT		
9	REF CURRENT RIGHT		
10	REF CURRENT LEFT DIST		
11	REF CURRENT RIGHT DIST		
12	REF NONE		
13	not used		
14	MAX REFTYPE		

Wert	PosSystemType	PosSystemMount	PosSystemResolver Freq
0	POS SYSTEM ENCODER	MOUNT INPUT SIDE	RESOLVER FREQ 1kHz
1	POS SYSTEM ENCODER INDEX	MOUNT OUTPUT SIDE	RESOLVER FREQ 2kHz
2	POS SYSTEM RESOLVER	MOUNT MIDDLE SIDE	RESOLVER FREQ 4kHz

Wert	PosSystemType	PosSystemMount	PosSystemResolver Freq
3	reserved	reserved	RESOLVER FREQ 8kHz
4	reserved	MAX POS MOUNT	MAX RESOLVER FREQ
5	reserved		
6	POS SYSTEM ENCODER DIFF		
7	POS SYSTEM ENCODER DEF INDEX		
8	reserved		
9	reserved		
10	reserved		
11	reserved		
12	MAX POS SYSTEM TYPE		

Wert	RampModeType	DigitalInType	DigitalOutType
0	RAMP TRAPEZ	DIGITAL_IN_NORMAL	DIGITAL_OUT_NORMAL
1	RAMP JERK	reserved	reserved
2	RAMP TRAPEZ SRU	reserved	reserved
3	RAMP NO RAMP	reserved	reserved
4	RAMP USER	reserved	reserved
5	RAMP LINEAR	reserved	reserved
6	MAX RAMP MODE	reserved	MAX DIGITAL OUT TYPE
7		MAX DIGITAL IN TYPE	
8			
9			
10			
11			
12			
13			
14			

Wert	Daten Typ		
0	DT_UNKOWN		
1	DT_UINT8		
2	DT_INT8		
3	DT_UINT16		
4	DT_INT16		
5	DT_UINT32		

Wert	Daten Typ		
6	DT_INT32		
7	DT_UINT64		
8	DT_INT64		
9	DT_FLOAT		
10	DT_DOUBLE		
11	DT_CHAR_ARRAY		
12	DT_BOOL		
13	DT_BINARY		
14	DT_ENUM		
15	MAX_DATA_TYPE		

## 7.2 Info- und Fehlercodes

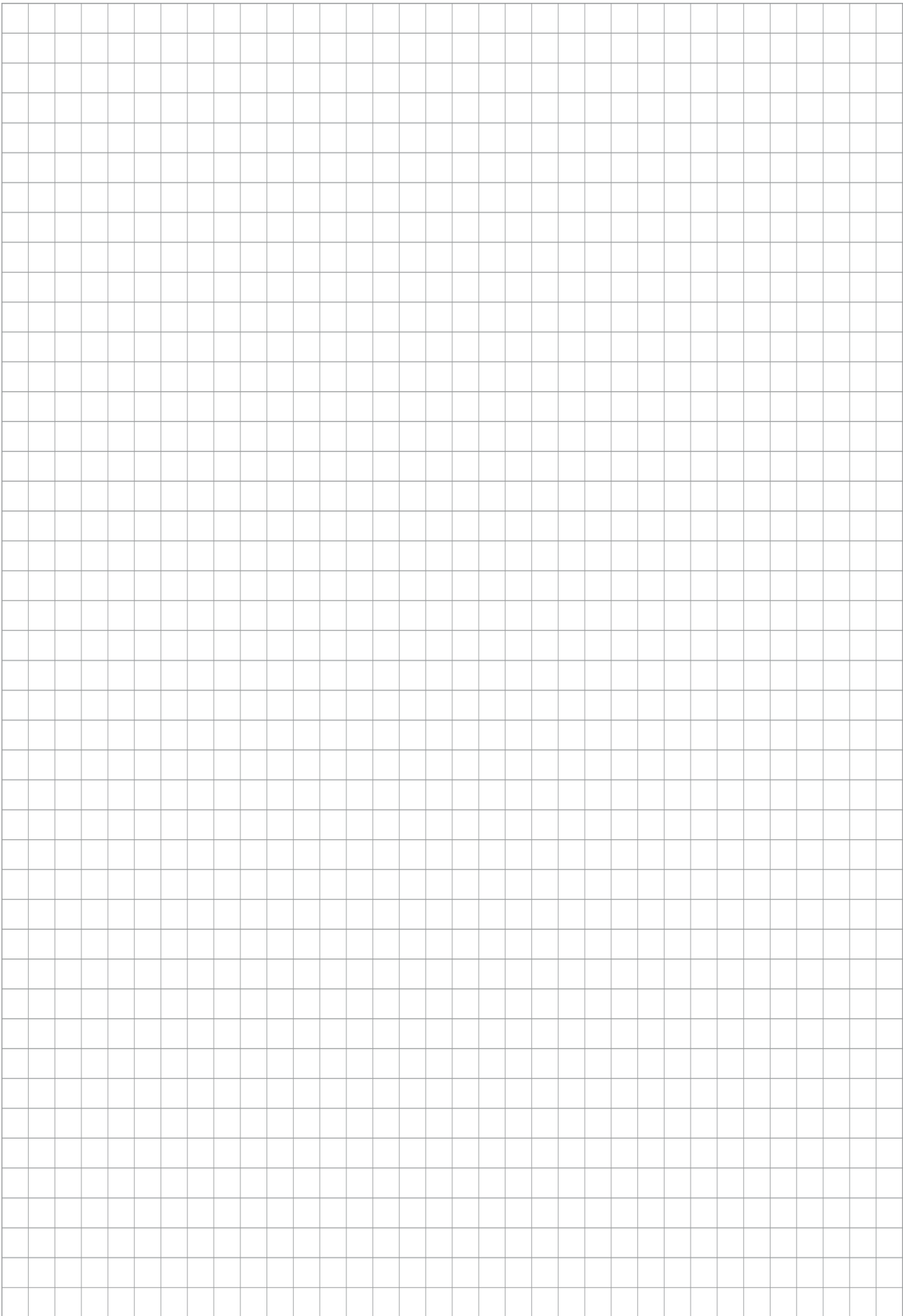
Hex	Dez	Name	Seite
0x0001	1	INFO BOOT	▶ 4.2.1 [□ 51]
0x03	3	INFO NO RIGHTS	▶ 4.2.2 [□ 51]
0x04	4	INFO UNKNOWN COMMAND	▶ 4.2.3 [□ 51]
0x05	5	INFO FAILED	▶ 4.2.4 [□ 51]
0x06	6	NOT REFERENCED	▶ 4.2.5 [□ 51]
0x0007	7	INFO SEARCH SINE VECTOR	▶ 4.2.6 [□ 51]
0x0008	8	INFO NO ERROR	▶ 4.2.7 [□ 52]
0x09	9	INFO COMMUNICATION ERROR	▶ 4.2.8 [□ 52]
0x10	16	INFO TIMEOUT	▶ 4.2.9 [□ 52]
0x12	18	INFO WRONG DATA TYPE	▶ 4.2.10 [□ 52]
0x19	25	INFO CHECKSUM	▶ 4.2.12 [□ 52]
0x1B	27	INFO VALUE LIMIT MAX	▶ 4.2.13 [□ 52]
0x1C	28	INFO VALUE LIMIT MIN	▶ 4.2.14 [□ 53]
0x1D	29	INFO MESSAGE LENGTH	▶ 4.2.15 [□ 53]
0x1E	30	INFO WRONG PARAMETER	▶ 4.2.16 [□ 53]
0x23	35	INFO UNKNOWN PARAMETER	▶ 4.2.17 [□ 53]
0x60	96	ERROR FILE NOT FOUND	▶ 4.2.18 [□ 53]
0x61	97	ERROR FILE IS CORRUPT	▶ 4.2.19 [□ 53]
0x62	98	ERROR FILE TYPE WRONG	▶ 4.2.20 [□ 53]
0x64	99	ERROR FILE SYSTEM WRONG	▶ 4.2.21 [□ 53]
0x65	100	ERROR FILE READ	▶ 4.2.22 [□ 53]
0x66	101	ERROR FILE IS NOT CREATED	▶ 4.2.23 [□ 54]
0x67	102	ERROR FILE WRITE	▶ 4.2.24 [□ 54]
0x6A	106	ERROR CONNECTION TEMP LOW	▶ 4.2.47 [□ 57]
0x6B	107	ERROR CONNECTION TEMP HIGH	▶ 4.2.48 [□ 57]
0x6C	108	ERROR MOTOR TEMP LOW	▶ 4.2.49 [□ 58]
0x6D	109	ERROR MOTOR TEMP HIGH	▶ 4.2.50 [□ 58]
0x6E	110	ERROR TEMP LOW OPTION	
0x6F	111	ERROR TEMP HIGH OPTION	
0x70	112	ERROR TEMP LOW	▶ 4.2.51 [□ 58]
0x71	113	ERROR TEMP HIGH	▶ 4.2.52 [□ 58]
0x72	114	ERROR LOGIC LOW	▶ 4.2.53 [□ 58]
0x73	115	ERROR LOGIC HIGH	▶ 4.2.54 [□ 58]
0x74	116	ERROR MOTOR VOLTAGE LOW	▶ 4.2.55 [□ 59]
0x75	117	ERROR MOTOR VOLTAGE HIGH	▶ 4.2.56 [□ 59]
0x76	118	ERROR CABLE BREAK	▶ 4.2.57 [□ 59]
0x7A	122	ERROR LIFE SIGN	▶ 4.2.58 [□ 60]

Hex	Dez	Name	Seite
0x7B	123	ERROR CUSTOM DEFINED	▶ 4.2.59 [□ 60]
0x7C	123	ERROR REBOOT	▶ 4.2.25 [□ 54]
0x7D	123	ERROR MOTOR PHASE	▶ 4.2.26 [□ 54]
0x82	130	ERROR OVERSHOOT	▶ 4.2.60 [□ 60]
0x83	131	ERROR HARDWARE VERSION	▶ 4.2.61 [□ 60]
0x84	132	ERROR SOFTWARE VERSION	▶ 4.2.62 [□ 60]
0xC8	200	ERROR WRONG RAMP TYPE	▶ 4.2.27 [□ 54]
0xD1	210	ERROR WRONG DIRECTION	▶ 4.2.28 [□ 54]
0xD2	210	ERROR CONFIG MEMORY	▶ 4.2.29 [□ 54]
0xD5	213	ERROR SOFT LOW	▶ 4.2.30 [□ 54]
0xD6	214	ERROR SOFT HIGH	▶ 4.2.31 [□ 55]
0xD8	216	ERROR SERVICE	▶ 4.2.32 [□ 55]
0xD9	217	ERROR FAST STOP	▶ 4.2.33 [□ 55]
0xDA	218	ERROR TOW	▶ 4.2.34 [□ 55]
0xDB	219	ERROR VPC3	▶ 4.2.35 [□ 55]
0xDC	220	ERROR FRAGMENTATION	▶ 4.2.36 [□ 55]
0xDD	221	ERROR COMMUTATION	▶ 4.2.37 [□ 56]
0xDE	222	ERROR CURRENT	▶ 4.2.39 [□ 56]
0xDF	223	ERROR I2T	▶ 4.2.38 [□ 56]
0xE0	224	ERROR INITIALIZE	▶ 4.2.45 [□ 57]
0xE1	225	ERROR INTERNAL	▶ 4.2.46 [□ 57]
0xE4	228	ERROR TOO FAST	▶ 4.2.40 [□ 56]
0xE5	228	ERROR POS SYSTEM	▶ 4.2.41 [□ 56]
0xEB	229	ERROR RESOLVER CHECK FAILED	▶ 4.2.42 [□ 56]
0xEC	236	ERROR MATH	▶ 4.2.43 [□ 56]
0xEE	236	ERROR CALIB CURRENT	▶ 4.2.44 [□ 57]

### 7.3 Firmware Update

Ein Firmwareupdate kann nur über USB erfolgen.

- USB Device: über das Software-Tool *MTS* (siehe Softwareanleitung "Motion Tool Schunk") kann ein Firmwareupdate ausgeführt werden.
- USB Host:
  - gewünschte Firmware-Datei auf einen USB-Stick kopieren
  - in "firmware.bin" umbenennen
  - USB-Stick mit Modul verbinden
  - USB-Freigabe betätigen (Anleitung Hardware)
  - Update Kommando über entsprechenden Schalter (Anleitung Hardware) freigeben
  - Update Prozess abwarten
  - Modul neu starten



**SCHUNK GmbH & Co. KG**  
**Spann- und Greiftechnik**

Bahnhofstr. 106 - 134  
D-74348 Lauffen/Neckar  
Tel. +49-7133-103-0  
Fax +49-7133-103-2399  
info@de.schunk.com  
schunk.com

Folgen Sie uns | *Follow us*

