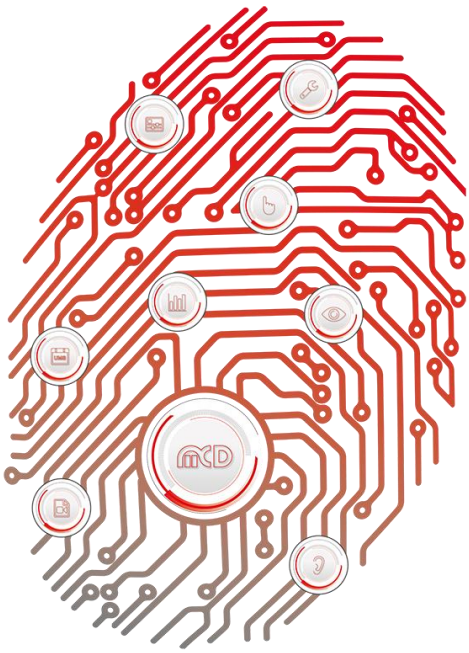
Bedienungsanleitung

Bürstenmotor-Ansteuerung



GET IN touch
WITH SENSITIVE TESTING

Softline

Modline

Conline

Boardline

Avidline

Pixline

Applikation

MCD Elektronik GmbH
Hoheneichstr. 52
75217 Birkenfeld
Telefon 0 72 31/78 405-0
Telefax 0 72 31/78 405-10
info@mcd-elektronik.de
www.mcd-elektronik.de
Sitz: Birkenfeld
Geschäftsführer: Bruno Hörter
Registergericht Mannheim
HRB 505692

Vorlagenversion: 5.4 / 2018-04-09
V2.2 2019-04-23 / MR (JJ)

Inhalt

1. ALLGEMEIN 4

2. LIEFERUMFANG 4

3. ANSCHLUSSBELEGUNG 5

4. EIGENSCHAFTEN 6

5. FUNKTIONEN 7

5.1. MOTORTREIBER 7

5.2. PWM - AUSGANG 7

5.3. KOMMUNIKATIONSEINGANG 8

5.3.1. Baudrate - Erkennung 8

5.3.2. Erzeugung BREAK - Zeichen 9

5.4. PROGRAMMIERGERÄT - ANSCHLUSS 9

5.5. SPANNUNGSVERSORGUNG FÜR HALL - SENSOR 9

5.6. SSI - ANSCHLUSS FÜR POSITIONSGEBER 9

5.7. EINGANG FÜR QUADRATURENCODER ODER IMPULSGEBER MIT RICHTUNGSSIGNAL 9

5.8. ANALOGEINGÄNGE 10

5.8.1. Datenquellen 10

5.8.2. Erfassungssteuerung 10

5.9. PID - REGLER 12

5.10. EEPROM - TABELLE 12

6. TECHNISCHE DATEN 13

7. SCHNITTSTELLENBESCHREIBUNG 14

7.1. ALLGEMEINE FESTLEGUNGEN 14

7.2. ALLGEMEINE BEFEHL STANDARD SYNTAX 14

7.3. ALLGEMEINE ANTWORT 14

7.4. MAXIMALE DATENLÄNGE DER ÜBERTRAGUNG 15

7.5. TYPDEFINITION 15

7.6. ERROR-CODES 15

7.7. BEFEHLSIMPLEMENTIERUNG BEI VERSCHIEDENEN SW - VERSIONEN 16

7.8. BEFEHLSSATZ 17

7.8.1. Befehl 20 Konfigurations - Speicher schreiben / lesen 17

7.8.2. Befehl 22 EEPROM lesen 18

7.8.3. Befehl 23 EEPROM schreiben 18
7.8.4. Befehl 2F Konfigurations - Speicher entsperren 19
7.8.5. Befehl 3F SW VERSION Firmware lesen 19
7.8.6. Befehl 40 Messdaten lesen 20
7.8.7. Befehl 41 Messdatenerfassung auslösen 21
7.8.8. Befehl 50 Encoderzählstand lesen 24
7.8.9. Befehl 52 Encoder konfigurieren 25
7.8.10. Befehl 61 PWM Ausgang einstellen 26
7.8.11. Befehl 64 Kommunikation von DUT lesen 27
7.8.12. Befehl 65 Kommunikation mit DUT schreiben 28
7.8.13. Befehl 66 Baudrate für Kommunikation von / zu DUT lesen 28
7.8.14. Befehl 67 Baudrate für Kommunikation von / zu DUT einstellen 29
7.8.15. Befehl 70 Motorstatus lesen 30
7.8.16. Befehl 71 Motor einstellen 31
7.8.17. Befehl 81 PID-Regler stellen 32
7.8.18. Befehl 82 Sollwert für PID-Regler einstellen 33

1. Allgemein

Dieses Gerät ist Ansteuerungsmodul für Bürstenmotoren und wurde mit dem Ziel der Steuerung und Kontrolle von Klappenstellern im Automobilbau entwickelt. Zudem soll es Anlagen zur Kraft - / Wegemessung vereinfachen. Es stellt dazu vielfältige Schnittstellen und Funktionen zur Verfügung.

Bestellnummer: # 123114

2. Lieferumfang

1 x Bürstenmotor-Ansteuerung

1 x USB - Speicherkarte mit Installationssoftware

3. Anschlussbelegung

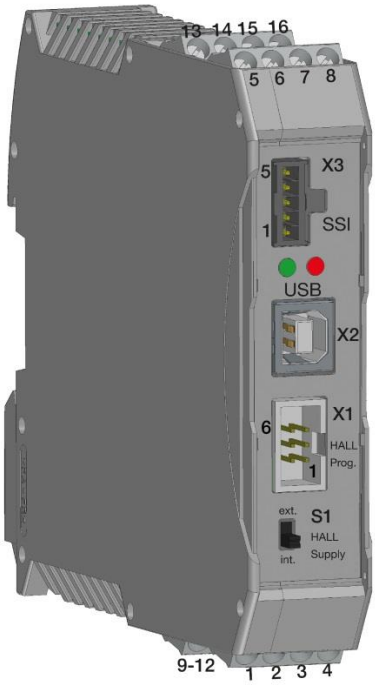


Abbildung 1: Anschlussbelegung

Klemmenbelegungen

Pin - Nr.	**13**	**14**	**15**	**16**
Signal	UART In / AUX / Decoder Z	GND	Decoder A	Decoder B
Pin - Nr.	**5**	**6**	**7**	**8**
Signal	Drehmoment In	GND	GND	PWM Out
Pin - Nr.	**1**	**2**	**3**	**4**
Signal	Supply Hall	Signal Hall	GND Hall	Schirm
Pin - Nr.	**9**	**10**	**11**	**12**
Signal	UB/ KL30 (7 - 18 V)	GND/ KL31	Motor +/ KL30 out	Motor -/ KL31 out

Steckerbelegungen

X1, Hall - Programmer

X1.1	VDD Sensor
X1.2	nc
X1.3	GND
X1.4	GND
X1.5	Sensor OUT
X1.6	nc

X3, SSI - Encoder

X3.1	Data IN +
X3.2	Data IN -
X3.3	GND
X3.4	CLK OUT +
X3.5	CLK OUT -

4. Eigenschaften

- **Motortreiber**
  - Vorwärts - und Rückwärtslauf
  - In vier Stufen einstellbare Spitzenstrombegrenzung
  - Automatische Abschaltung bei Kurzschluss und Übertemperatur
  - Symmetrische PWM mit einstellbarer Frequenz und Duty - Cycle
  - Messung des Motorstroms zwischen der H - Brücke und dem Motor
    Anmerkung: Es wird der Strom durch den Motor gemessen und **nicht** der Strom in der Zuleitung!
  - „Open“ - Betrieb ermöglicht Open - Collector - und Open - Emitter - Konfigurationen

- **PWM - Ausgang**
  - Kleinsignal - Transistorausgang mit Kurzschlussstrombegrenzung
  - Zuschaltbarer 2 kΩ Pullup - Widerstand zur Betriebsspannung (ab Hardwareversion V1.1)
  - Funktionen für Abgleich und Kommunikation mit Klappenstellern

- **Programmiergerät - Anschluss**
  - Anschluss zum Einschleifen eines Hall - Sensor - Programmiergeräts

- **Spannungsversorgung für Hall - Sensor**
  - Präzisionsspannungsquelle 5 V über Schiebeschalter wählbar. Zu verwenden, wenn kein Programmiergerät angeschlossen ist

- **SSI - Anschluss für Positionsgeber**

- **Eingang für Quadraturencoder oder Impulsgeber mit Richtungssignal**
  - Optional mit Rücksetzeingang
  - Optionaler Prescaler erweitert den Zählbereich

- **Analogeingänge**
  - Drehmomentsensor: Messbereich ± 10 V
  - Spannungsversorgung Hall - Sensor: Messbereich 10 V (Kontrollmessung)
  - Signal des Hall - Sensors: Messbereich 5 V
  - AUX - Messeingang: Messbereich: 18 V (ab Hardwareversion V1.1)
  - Erfassung von Signalen mit bis zu 131071 Messpunkten, wahlweise synchron zum Quadraturencoder / Impulsgeber (Kraft / Weg synchron)
  - Pre - und Post - Triggeroptionen; Triggern der Messwerterfassung auf beliebigen Analog - oder Positionseingang

- **PID - Regler**
  - Ansprechverhalten und Regelbereich weitgehend parametrierbar
  - Regelausgang auf PWM - Ausgang oder Motortreiber schaltbar
  - Regelung auf Eingangssignal:
    - Hall - Sensor
    - Drehmomentsensor
    - Position (Quadraturencoder / Impulsgeber oder SSI)

5. Funktionen

5.1. Motortreiber

Der Motortreiber verfügt über eine vierstufig einstellbare Spitzenstrombegrenzung und eine Messeinrichtung zur Erfassung des momentanen Motorstroms. Bei Kurzschluss oder Überhitzung schaltet er ab. Diese Ereignisse werden durch eine rote LED an der Front angezeigt und können auch via Steuerkommando vom übergeordneten Gerät abgefragt werden. Der dazu gehörende PWM - Generator kann in Frequenz und Duty - Cycle eingestellt werden. Wenn gewünscht, kann der PID - Regler den Duty - Cycle steuern. Die gewünschte PWM - Frequenz wird dazu separat festgelegt.

Im „Open“ - Betrieb wird der Motortreiber während der inaktiven Zeit des PWM - Signals abgeschaltet. Dadurch werden Open - Emitter - und Open - Collector - Konfigurationen ermöglicht. Da dann ggf. der Motorstrom durch Freilauf - bzw. Überspannungsschutzdioden fließt, kann durch die erhöhte Wärmeentwicklung im Motortreiber dessen maximale Belastbarkeit reduziert sein.

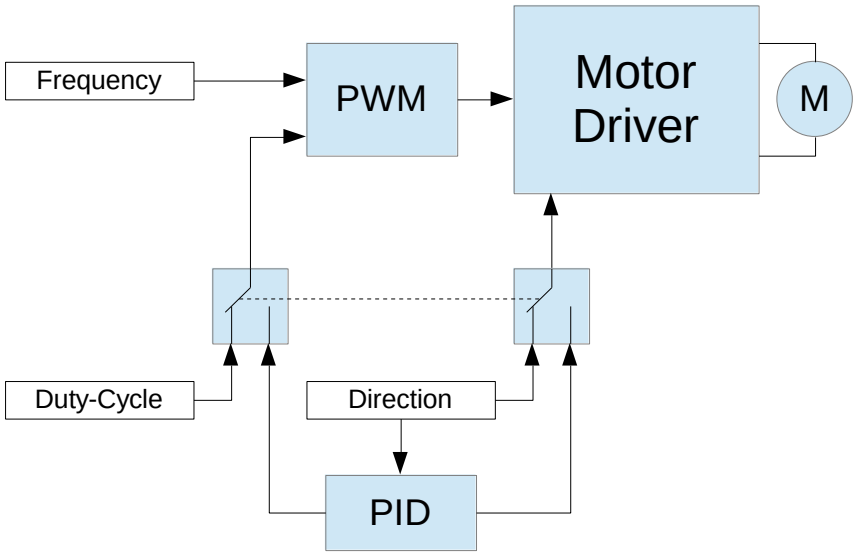


Abbildung 2: Skizze Motortreiber

5.2. PWM - Ausgang

Der PWM - Ausgang ist ein Steuerausgang geringer Leistung mit einem zuschaltbaren 2 kΩ Pullup - Widerstand zur Betriebsspannung. Wie beim Motortreiber ist der PWM - Ausgang in Frequenz und Duty - Cycle parametrierbar. Es können bis zu 40 verschiedene Duty - Cycles mit einer Länge von je bis zu 256 Perioden nacheinander ausgegeben werden, wobei der letzte dann bis zur nächsten Änderung dauerhaft ausgegeben wird. Dies ist eine Funktion, die für manche Klappensteller zum Auslösen von Sonderfunktionen benötigt wird. Die Polarität des Duty - Cycles kann Low oder High sein.

Folgende weitere Funktionen sind möglich:

- Wird das Steuersignal von außen länger als 200 ms auf „Low“ gezogen, so schaltet das PWM - Signal ab und ein Empfänger lauscht auf der Leitung auf asynchrone serielle Signale mit einer parametrierbaren Baudrate. Empfangene Zeichen können per Steuerbefehl vom übergeordneten Gerät abgerufen werden. Die PWM bleibt ausgeschaltet, bis sie per Befehl wieder gestartet wird. Diese Funktionalität kann auch abgeschaltet werden. Der momentane Zustand der PWM - Leitung kann abgefragt werden (siehe Kapitel 5.8).

- Der Ausgang kann auch serielle Daten ausgeben. Dazu muss der Duty - Cycle der PWM auf 0% stehen, da sonst die beiden Signale miteinander interferieren. Die Ausgabe erfolgt mit Polarität des Duty - Cycles.

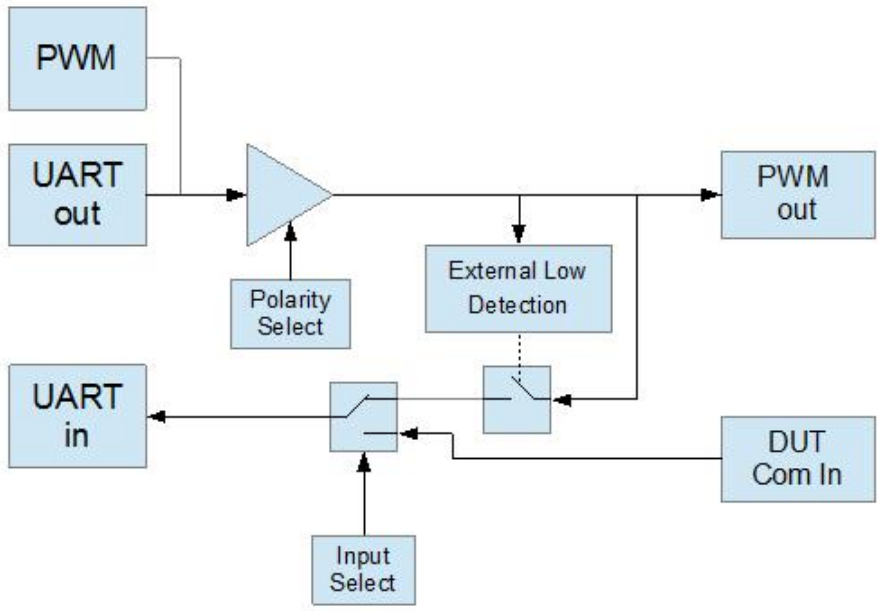


Abbildung 3: Skizze des PWM - Ausgangs

5.3. Kommunikationseingang

Der Kommunikationseingang kann wahlweise die Funktion des seriellen Empfängers im PWM - Ausgang übernehmen. Das 200 ms Warten auf ein „low ziehen“ entfällt hier, jedoch ist der Empfang ist kontinuierlich gegeben.

5.3.1. Baudrate - Erkennung

Der Kommunikationseingang verfügt über eine Funktion zur Baudraten - Erkennung. Ist diese aktiviert, wird einmalig das Startbit vermessen und die Baudrate damit eingestellt. Damit dies funktioniert, muss das LSB des zu sendenden Bytes gesetzt sein (z.B. 0 x 01, 0 x 55 etc.).

Die ermittelte Baudrate kann zur Kontrolle zurückgelesen werden. Es ist möglich, dass das erste übertragene Byte nicht korrekt eingelesen wird, wenn die voreingestellte Baudrate grob von der tatsächlichen abweicht. Das darauf folgende Byte sollte in der Regel korrekt erkannt werden sofern die Abweichung unter 10% liegt. Diese Funktion erlaubt es auch, die Dauer eines einzelnen Pulses (High oder Low) im Bereich von 1 bis 1024 µs zu messen. Dies erlaubt es z.B. auch, über die Abweichung von PWM - Pulsen vom Nominalwert auf Taktabweichungen bei der UART zu schließen und die Baudrate vorab manuell einzustellen. Alternativ kann die Baudrate auch durch Vermessen im Streaming und manuelles Berechnen ermittelt werden.

5.3.2. Erzeugung BREAK - Zeichen

Zur Unterstützung einer LIN - Kommunikation kann die Bürstenmotor-Ansteuerung angewiesen werden, vor jedem Datenpaket einen Break - Impuls zu generieren. Die Länge des Pulses ist abhängig von der eingestellten Baudrate.

5.4. Programmiergerät - Anschluss

An der Frontseite befindet sich ein Anschluss für Hall - Sensor - Programmiergeräte der Firma Micronas. Die Signale werden durchgeschleift und können mittels eines geschirmten Sensorkabels weiter zum Prüfling geführt werden. Die Versorgungsspannung für den Sensor vom Programmiergerät kann zur Kontrolle nachgemessen werden. Das Sensorsignal kann ebenfalls gemessen und dem PID - Regler (siehe Kapitel 5.9) als Ist - Signal zugeführt werden.

5.5. Spannungsversorgung für Hall - Sensor

Für den Fall, dass kein Programmiergerät zur Versorgung des Hall - Sensors zur Verfügung steht, kann mit dem Schiebeschalter an der Front auf eine interne Präzisionsspannungsquelle umgeschaltet werden.

Diese Spannungsquelle kann zudem via Steuerkommando ein - und ausgeschaltet werden.

5.6. SSI - Anschluss für Positionsgeber

Die Synchron - Serielle - Schnittstelle liefert ein differentielles Taktsignal mit einer festen Frequenz und liest das vom angeschlossenen kompatiblen Sensor gelieferte differentielle Signal ein.

5.7. Eingang für Quadraturencoder oder Impulsgeber mit Richtungssignal

Der Eingang für Quadraturencoder ist ein Transistoreingang und spricht auf Eingangssignale über einen weiten Spannungsbereich an. Wenn der Quadraturencoder nur Open - Collector - Ausgänge hat, sind externe Pull - up - Widerstände notwendig. Der Eingang kann per Steuerbefehl so umgeschaltet werden, dass er auch Impulse mit einem Richtungssignal zählen kann. Die Polarität des Richtungssignals ist einstellbar. Der Zählerstand kann jederzeit auf einen gewünschten Wert (zurück -) gesetzt werden.

Zusätzlich gibt es noch einen einstellbaren Vorteiler und einen externen Rücksetzeingang. Für das externe Rücksetzsignal wird der *DUT COM In* Eingang genutzt. Auf Wunsch setzt er den Zähler bei steigender und / oder fallender Flanke auf den Startwert zurück. Zur Unterstützung von endlos rotierenden Bewegungen kann zudem ein Maximalzählwert festgelegt werden, bei dessen Überschreitung der Zähler auf Null gesetzt bzw. bei Unterschreitung von Null auf den maximalen Wert gesetzt wird.

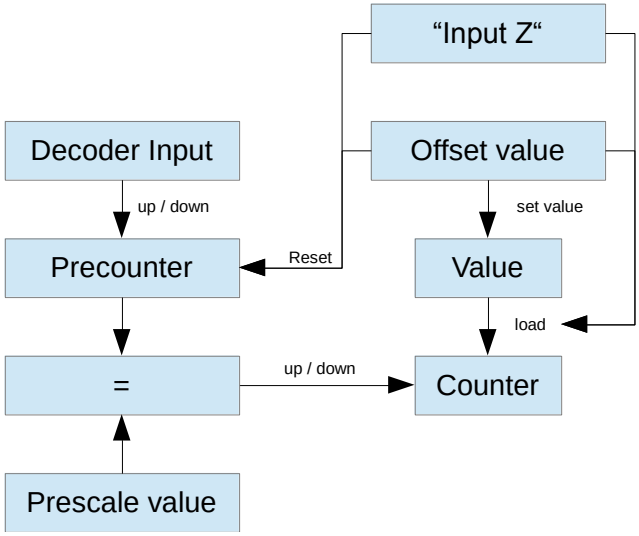


Abbildung 4: Eingang für Quadraturencoder oder Impulsgeber

5.8. Analogeingänge

5.8.1. Datenquellen

Es gibt vier Analogeingänge: Drehmoment - oder Kraftsensor, Motorstrom, Hall - Signal und Spannungsversorgung für den Hall - Sensor. Nur der Eingang für den Drehmoment - oder Kraftsensor hat einen dedizierten Anschluss. Alle anderen sind intern mit den entsprechenden Leitungen verbunden.

Es können sowohl einzelne als auch bis zu 131071 Werte erfasst werden. Wenn mehr als eine Signalquelle erfasst werden sollen verringert sich entsprechend die maximale Anzahl an Werten, die auf in einem Strom erfasst werden können:

$$Max.Werteanzahl = \frac{131071}{Anzahl\ Kanäle}$$

Werden also alle vier Analogeingänge erfasst, so können maximal 32767 Werte - Quartetts aufgenommen werden.

Zusätzlich zu den vier analogen Eingängen stehen noch folgende Kanäle bereit:

Zählerstand Quadratureingang, der Wert des SSI - Eingangs und der Zustand aller digitalen Eingangsleitungen (inklusive dem PWM - Ausgang) und der Motor - Steuersignale.

Eine Tabelle aller erfassten Digitalsignale gibt es bei der Beschreibung für Befehl 41 (Kapitel 7.8.7).

5.8.2. Erfassungssteuerung

Die Messdaten können zeitabhängig mit einer konfigurierbaren Samplerate oder wegabhängig synchron zum Quadraturencoder - /Impulszählereingang erfasst werden. Die maximale Zählfrequenz darf dabei 100 kHz nicht überschreiten, sonst gehen Impulse verloren.

Achtung: Bei einem Quadraturencoder bedeutet dies eine maximale Signalfrequenz von 25 kHz!

.

Zudem kann eine Bedingung definiert werden, bei deren Erfüllung die Messdatenerfassung erfolgt (Triggering). Typischerweise ist dies die steigende oder fallende Flanke eines sich ändernden Eingangssignals bei einem bestimmten Spannungswert. Der Trigger kann sich dabei vor, nach oder an einer bestimmten Position im Messdatenstrom befinden. Es kann außerdem bestimmt werden, ob erst die spezifizierte Anzahl an Messwerten erfasst worden sein muss, ehe eine Triggerbedingung ausgewertet wird, oder ob es wichtiger ist, die erste erfüllte Triggerbedingung zu erfassen. Wird die Triggerbedingung nicht erreicht oder die Erfassung findet wegen fehlender oder zu weniger Zählimpulse kein Ende, so wird die Messung durch ein einstellbares Timeout beendet und die bis dahin erfassten Messdaten zurückgegeben.

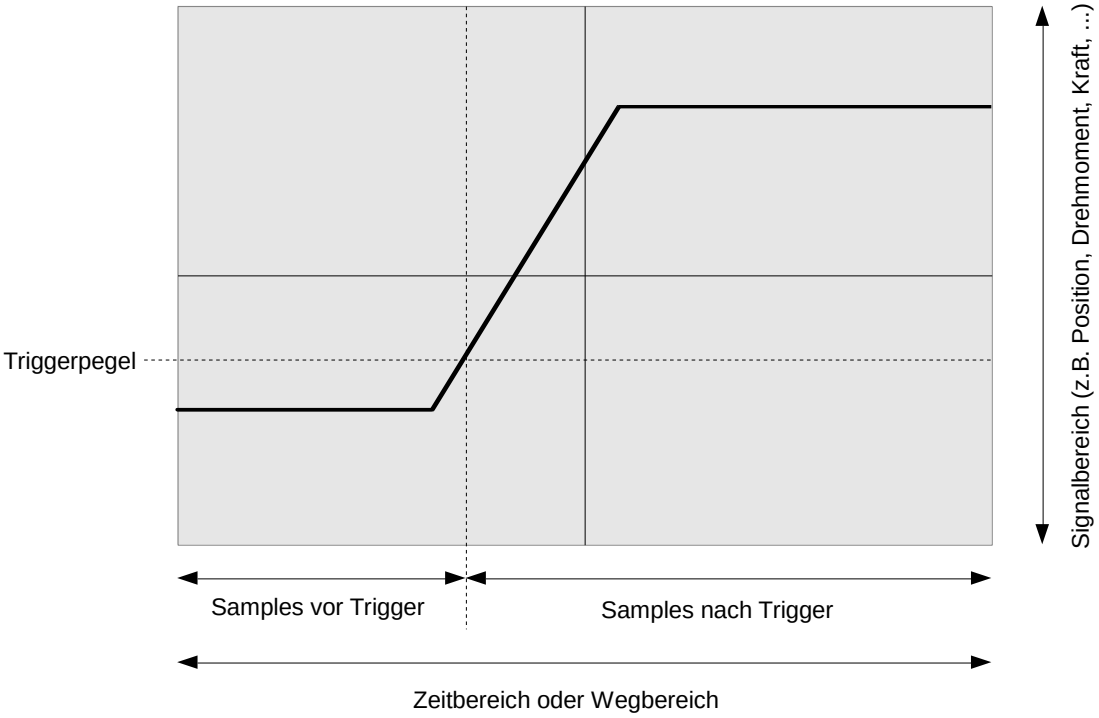
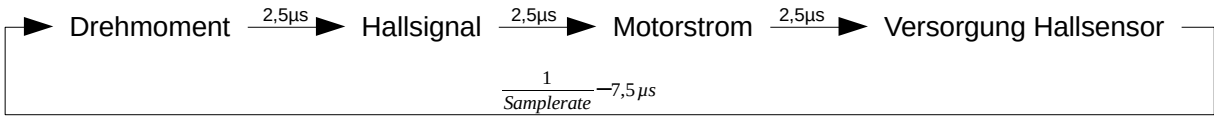
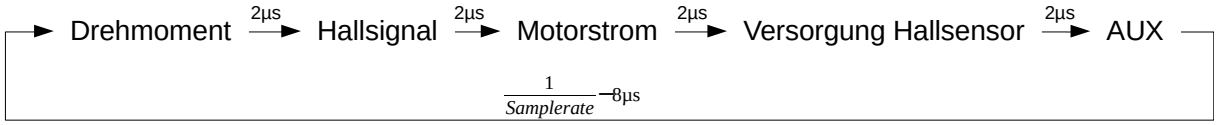


Abbildung 5: Erfassungssteuerung

Die Erfassung der analogen Messwerte erfolgt im Hintergrund in einer festen Reihenfolge und in einem definierten zeitlichen Bezug:



Bei Modellen mit AUX - IN:



5.9. PID - Regler

Der PID - Regler kann neben den drei Koeffizienten auch in seinem zeitlichen Ansprechverhalten der I - und D - Komponente beeinflusst werden. Als Eingangswert können die analogen Messwerte für Drehmoment / Kraft und Hall - Signal sowie der Zählwert des Quadraturencoders und des SSI - Eingangs herangezogen werden. Der I - Anteil und der endgültige Ausgangswert können begrenzt werden. Der Ausgangswert wird als Duty - Cycle auf die PWM des Motortreibers und / oder des PWM - Ausgangs gelegt. Wenn negative Ausgangswerte erlaubt sind, so treibt der Motortreiber aktiv in beide Richtungen. Für den PWM - Ausgang sollten negative Werte nicht erlaubt werden, da mit einem PWM - Signal keine Polarität dargestellt werden kann. Bei einem negativen Ausgangswert steigt daher die PWM wieder an.

5.10. EEPROM - Tabelle

Belegung des EEPROMs:

EEPROM - Adresse	Firmware bis V1.4x	Firmware V1.5x
00	frei	Abgleich Offset - Strommessung
01…63	frei	frei

Vorbelegte Speicherstellen dürfen vom Anwender nicht überschrieben werden. Zuwiderhandlung führt zu einer Fehlfunktion (Messfehler), die unbemerkt bleiben kann.

6. Technische Daten

Elektrische Eigenschaften		
Betriebsspannung	7 – 18 VDC	Absolutes Maximum ist 19 V Für einen korrekten Reset muss die Spannung ≥ 1 s unterbrochen sein
Leistungsaufnahme	< 1 W	Stromaufnahme typisch < 60 mA
Versorgung Hall - Sensor	5 V ± 0,2% max.; 20 mA max.	Achtung: Ein Kurzschluss am Ausgang kann die Genauigkeit über den spezifizierten Wert beeinträchtigen, bis der Regler wieder abgekühlt ist!
PWM - Ausgang	Open - Collector - Ausgang mit 100 kOhm Pullup - Widerstand zur Betriebsspannung	Pulldown - Strom bei 12 V: ca. 13 mA Externer Pullup je nach Applikation
SSI: Synchron - Serielle - Schnittstelle	Takt: 500 kHz Auflösung: 16 Bit Abschlusswiderstand: 10 kOhm	
Analogeingang	Maximale Samplerate: 100 kSps	Eingangswiderstand: 100 kOhm
Analoge Messbereiche	Hall - Signal: 0…5,12 V Hall - Supply: 0…10,24 V Kraftsensor: ± 10,24 V Strom: 0…5,12 V AUX (UART In) siehe unten	Nominalwerte ohne Kalibrierung 0,048…4,048 V ≙ - 8 A…+ 8 A
Eingang Quadraturencoder	Eingangsimpedanz: 20 kOhm Pegel Low: < 0,8 V Pegel High: > 2,5 V Eingangsfrequenz: max. 100 kHz max. 50kHz	 Max. 30 V Bei TTL - Pegel Bei 10 V_s (Bei höherer Spannung kann zur Erhöhung der Grenzfrequenz ein Vorwiderstand von ca. 20 kΩ / 5 V verwendet werden)
Eingang UART In / Decoder-Z	Eingangsimpedanz: 1 MOhm Digital: Pegel Low: < 3,3 V Pegel High: > 3,3 V Analog: 0,1 V…18,7 V	 Max. 30V Gain - Fehler ± 2% max. (unkalibriert)
Mechanische Eigenschaften		
Baugröße (H x B x T)	116 mm x 22 m x 115 mm	Ohne angeschlossene Stecker
Anschlüsse	Schraubklemmen	
	1 x USB - B 2.0	Steuereingang
	Stiftleiste 2 x 3polig	Einschleifung Programmiergerät für Hall - Sensor
	Stiftleiste Samtec IPL1 1 x 5polig	SSI: Schnittstelle für Positionsgeber
Sonstige Eigenschaften		
Ansteuerung	Virtueller serieller Port über USB galvanisch getrennt	3 MBaud 1 Startbit 1 Stopbit Kein Handshake
Anzeige	1 x LED grün	Betrieb / Aktivität
	1 x LED rot	Warnung / Meldung Motortreiber
Betriebstemperatur	- 40 – 85°C	Motortreiber bis 125°C
Gewicht	130 g	

7. Schnittstellenbeschreibung

7.1. Allgemeine Festlegungen

- Als Startzeichen wird $12 gesendet
- Längenbyte ist die Anzahl der gesendeten ASCII Zeichen (Char) beginnend mit dem CMD-Byte
- Als Ende und Terminierungszeichen wird $0D gesendet

7.2. Allgemeine Befehl Standard Syntax

Befehl: „?“ entspricht einem ASCII Zeichen

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			Triggerzeichen für Interface
??	LENbyte	Längenbyte [u08]		2..255	Anzahl der ASCII Zeichen beginnend mit CMD - Byte bis Daten - Byte n
??	CMDbyte	Befehlscode [u08]			Befehlscode siehe Befehl
??	Dbyte1				
$0D	Term.	Terminierung			Abschlusszeichen

7.3. Allgemeine Antwort

Antwort: bei korrekter Befehls - Parametrisierung

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			Triggerzeichen für Interface
??	CMDbyte	Befehlscode [u08]			Wiederholung des Befehlscode in der Antwort
??	Dbyte1				
$0D	Term.	Terminierung			Abschlusszeichen

Antwort: bei Fehler in Befehls - Parametrisierung

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			Triggerzeichen für Interface
FF	CMDbyte	ERROR [u08]			Default „FF“ für Fehler
??	Fehlerbyte	Errorcode [u08]			Errorcode
$0D	Term.	Terminierung			Abschlusszeichen

7.4. Maximale Datenlänge der Übertragung

Bei Befehlen ohne feste Länge dürfen nicht mehr als **127 Datenbytes** (254 ASCII Zeichen high Byte / low Byte) nach dem Befehlsbyte (CMDbyte) folgen.

Eine Ausnahme bildet der Befehl mit Messdaten (Befehl 40 – Kapitel 7.8.6). Nach Abschluss des Befehlsstrings werden die Messdaten binär übertragen.

7.5. Typdefinition

Datentyp	*Abkürzung*
unsigned char	u08
unsigned short	u16
signed short	s16
unsigned long	u32

7.6. ERROR-Codes

Datentype	*Abkürzung*
ERROR	0x0F
NOCMD	0x01
PARAMS	0x03
WERTEBEREICH	0x04
CMDLEN	0x05

7.7. Befehlsimplementierung bei verschiedenen SW - Versionen

Befehl	Beschreibung	Firmware - Version							
		1.00	1.10	1.20	1.30	1.31	1.32	1.4	1.6
20	Konfigurations - Speicher schreiben / lesen	X	X	X	X	X	X	X	X
2F	Konfigurations - Speicher entsperren	X	X	X	X	X	X	X	X
22	EEPROM lesen	X	X	X	X	X	X	X	X
23	EEPROM schreiben	X	X	X	X	X	X	X	X
3F	SW - Version abfragen	X	X	X	X	X	X	X	X
40	Messdaten auslesen	X	X	X	X	X	X	X	X
41	Messdatenerfassung auslösen	X	X	X	X[2)]	X	X	X[1)]	X
50	Encoderzählstand lesen	X	X	X	X	X	X	X	X
51	Encoder einstellen	X	X						
52	Encoder konfigurieren	X	X[1)]	X[1)]	X	X[1)]	X	X	X
61	PWM - Ausgang einstellen	X	X[1)]	X	X[2)]	X	X	X	X
64	Kommunikation von DUT lesen	X	X	X	X	X	X	X	X
65	Kommunikation zu DUT schreiben	X	X	X	X	X	X	X	X
66	Baudrate für Kommunikation von / zu DUT lesen								X
67	Baudrate für Kommunikation von / zu DUT einstellen	X	X	X	X	X	X	X	X
70	Status Motortreiber lesen	X	X	X	X	X[1)]	X	X	X
71	Motortreiber einstellen	X	X[1)]	X	X	X	X[1)]	X	X
81	PID - Regler einstellen	X	X	X	X	X	X	X	X
82	Sollwert für PID - Regler einstellen	X	X	X	X	X	X	X	X

[1)] Befehl erweitert
[2)] Befehl erweitert, erfordert zur Nutzung Hardwareversion 1.1

7.8. Befehlssatz

7.8.1. Befehl 20 Konfigurations - Speicher schreiben / lesen

Dieser Befehl ist ausschließlich für Firmware - Updates vorgesehen und sollte im normalen Betrieb niemals verwendet werden!

Ist der Speicher gesperrt, so wird der Fehlercode FF04 zurückgegeben.

Befehl:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
??	LENbyte	Längenbyte [u08]		2..255	
20	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	DATA [u08]		0..255 = 00h..FFh	Zu sendende Daten
??	…	DATA [u08]		0..255 = 00h..FFh	
??	Dbyte x	DATA [u08]		0..255 = 00h..FFh	
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
20	CMDbyte	Befehlscode			
??	Dbyte 1	DATA [u08]		0..255 = 00h..FFh	Gelesene Daten
??	…	DATA [u08]		0..255 = 00h..FFh	
??	Dbyte x	DATA [u08]		0..255 = 00h..FFh	
$0D	Term.	Terminierung			

7.8.2. Befehl 22 EEPROM lesen

Liest ein 16 Bit - Wort aus dem EEPROM. Wird eine ungültige Adresse angegeben, so wird der Fehlercode FF04 zurückgegeben.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
04	LENbyte	Längenbyte [u08]		2..255	
22	CMDbyte	Befehlscode [u08]			
??	Dbyte	Adresse [u08]		0..63 = 00h..3Fh	
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
22	CMDbyte	Befehlscode			
??	Dbyte 1	DATA [u08]		0..255 = 00h..FFh	Gelesene Daten
??	Dbyte 2	DATA [u08]		0..255 = 00h..FFh	
$0D	Term.	Terminierung			

7.8.3. Befehl 23 EEPROM schreiben

Schreibt ein 16 Bit - Wort in das EEPROM. Wird eine ungültige Adresse angegeben, so wird der Fehlercode FF04 zurückgegeben.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
08	LENbyte	Längenbyte [u08]		2..255	
23	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	DATA [u08]		0..63 = 00h..3Fh	Adresse
??	Dbyte 2	DATA [u16]	High Byte	0..65535 = 0000h..FFFFh	Datenwort
??	Dbyte 3		Low Byte		
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
23	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

7.8.4. Befehl 2F Konfigurations - Speicher entsperren

Entsperrt den Zugriff auf den Konfigurationsspeicher. Jeder andere Befehl außer Befehl 20 sperrt den Zugriff wieder.

Befehl:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
04	LENbyte	Längenbyte [u08]		2..255	
2F	CMDbyte	Befehlscode [u08]			
??	Dbyte	DATA [u08]		0..255 = 00h..FFh	55h = Konfigurations - Speicher entsperrt Jeder andere Wert sperrt den Speicher
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
2F	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

7.8.5. Befehl 3F SW VERSION Firmware lesen

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
02	LENbyte	Längenbyte [u08]		2..255	
3F	CMDbyte	Befehlscode [u08]			
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
3F	CMDbyte	Befehlscode			
??	Dbyte 1	DATA [u08]			Textstring der Version
??	…	DATA [u08]			
??	Dbyte x	DATA [u08]			
$0D	Term.	Terminierung			

7.8.6. Befehl 40 Messdaten lesen

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
04	LENbyte	Längenbyte [u08]		2..255	
40	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Anzahl Datensätze [u08]		0..255 = 00h..FFh	Bit 0: 0 = Auf Ende der Messung prüfen und Daten zurückgeben (wenn nicht fertig, wird ein Fehler 04h zurückgegeben) 1 = Messung abbrechen und vorhandene Daten zurückgeben Bit [7:1] = Nicht benutzt – muss immer 0 sein
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
40	CMDbyte	Befehlscode			
??	Dbyte 1	DATA [u17]	High Byte	0..131071 = 000000h..01FFFFh	Anzahl Datensätze vor Trigger
??	Dbyte 2				
??	Dbyte 3		Low Byte		
??	Dbyte 4	DATA [u17]	High Byte	0..131071 = 000000h..01FFFFh	Anzahl Datensätze nach Trigger
??	Dbyte 5				
??	Dbyte 6		Low Byte		
$0D	Term.	Zwischenterminierung			
$??	Dbyte 7	Erster Datensatz; erster Wert	High Byte	0..65535 = 0000h..FFFFh	
$??	Dbyte 8		Low Byte		
…					
$??	Dbyte n-1	Letzter Datensatz; letzter Wert	High Byte	0..65535 = 0000h..FFFFh	
$??	Dbyte n		Low Byte		
$0D	Term.	Terminierung			

7.8.7. Befehl 41 Messdatenerfassung auslösen

Maximal können 131071 Datenworte erfasst werden. Jede Messquelle benötigt je Sample ein Datenwort. Die gesampelten Daten bleiben bestehen, bis sie mit Befehl 40 ausgelesen werden oder eine neue Erfassung gestartet wird.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
1A	LENbyte	Längenbyte [u08]		2..255	
41	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Anzahl Datensätze [u17]	High Byte	0..131071 = 000000h..01FFFFh	Anzahl der zu lesenden Datensätze vor Trigger

??	Dbyte 2				
??	Dbyte 3		Low Byte		
??	Dbyte 4	Anzahl Datensätze [u17]	High Byte	0..131071 = 000000h..01FFFFh	Anzahl der zu lesenden Datensätze nach Trigger
	Dbyte 5				
??	Dbyte 6		Low Byte		
??	Dbyte 7	Triggerschwelle [u16] / s[16]	High Byte	0..65535 = 0000h..FFFFh - 32768..32767 = 8000h..7FFFh	Triggerschwelle, je nach Wahl der Triggerquelle mit oder ohne Vorzeichen: Eingang Hall - Sensorsignal und Spannungsversorgung Hall - Sensor sind vorzeichenlos, alle anderen mit Vorzeichen Wert wird bei PWM - Eingängen ignoriert
??	Dbyte 8		Low Byte		
??	Dbyte 9	Datenquellen [u08]		0..255 = 00h..FFh	Bit 0 = Eingang des Drehmomentsensors lesen Bit 1 = Motorstrom lesen Bit 2 = Hall - Sensorsignal lesen Bit 3 = Spannungsversorgung Hall - Sensor lesen Bit 4 = Encoder - Eingang lesen Bit 5 = Daten SSI - Schnittstelle lesen Bit 6 = Zustand der Digitalein - und ausgänge Bit 7 = Analogwerterfassung PWM / UART / Decoder - Z - Eingang (Pin 13)[2)]
??	Dbyte 10	Triggerquelle [u08]		0..255 = 00h..FFh	00h = Eingang des Drehmomentsensors 01h = Eingang Motorstrom 02h = Eingang Hall - Sensorsignal 03h = Eingang Spannungsversorgung Hall - Sensor 04h = Eingang Encoder - Eingang 05h = Eingang SSI - Schnittstelle 06h = PWM - Ein - / Ausgang 07h = Analogwert PWM / UART / Decoder - Z - Eingang (Pin 13)[2)]

??	Dbyte 11	Triggerkonfiguration [u08]		0..255 = 00h..FFh	Bit 0 = 0/1: Start mit Trigger / Start sofort Bit 1 = 0/1: Triggerflanke fallend / steigend Bit 2 = Trigger kann nicht auslösen, ehe die spezifizierte Menge an Daten erfasst wurde Bit 3 = Nicht benutzt – muss immer 0 sein Bit 4 = Datenerfassung mit Encodereingang synchronisieren Bit [7:5] = Nicht benutzt – muss immer 0 sein
??	Dbyte 12	Samplerate [u08]		0..255 = 00h..FFh	Bit [4:0] = Teiler für Samplerate - 1[3)] Bit [7:5] = Nicht benutzt – muss immer 0 sein
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
41	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

[2)] Befehl erweitert ab V1.30, Nutzung erfordert Hardwareversion 1.1
[3)] Teiler erweitert von 4Bit auf 5Bit ab V1.40

Bitbelegung der Digitalein - und Ausgänge:

Bit 0	Encodereingang B
Bit 1	Encodereingang A
Bit 2	Eingang SSI
Bit 3	PWM_IO
Bit 4	DUT Com In
Bit 5	Motor PWM
Bit 6	Motor Richtung
Bit 7	Motor Ein / Aus
Bit [15:8]	nicht benutzt

Bitte beachten, dass die Digitalsignale gegebenenfalls schneller wechseln als eingelesen wird, was dann zu Verfälschungen des Signals führt (Undersampling).

7.8.8. Befehl 50 Encoderzählstand lesen

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
02	LENbyte	Längenbyte [u08]		2..255	
50	CMDbyte	Befehlscode [u08]			
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
50	CMDbyte	Befehlscode			
??	Dbyte 1	DATA [u16]	High Byte	0..65535 = 0000h..FFFFh	Zählerstand Quadraturencoder / Impulszähler
??	Dbyte 2		Low Byte		
??	Dbyte 3	DATA [u16]	High Byte	0..65535 = 0000h..FFFFh	Wert von SSI
??	Dbyte 4		Low Byte		
$0D	Term.	Terminierung			

7.8.9. Befehl 52 Encoder konfigurieren

Der Encodereingang kann als Quadraturencoder oder als Impulszähler mit Richtungssignal arbeiten. Optional kann ein Hardware - Reset für den Zählerstand festgelegt werden. Der Wert wird dann auf den Offsetwert zurückgesetzt (Standardwert ist Null). Zudem kann der Zähler manuell auf den Offsetwert gestellt werden.

Befehl:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
0E	LENbyte	Längenbyte [u08]		2..255	
52	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Konfiguration [u08]		0..255 = 00h..FFh	Bit 0 = 0/1: Quadraturencoder / Pulszähler mit Richtungssignal Bit 1 = Nicht benutzt – muss immer 0 sein Bit 2 = 0/1: Zählrichtung normal / umgekehrt Bit 3 = Encoder gibt 0000h zurück, bis eine Flanke bei Anschluss 13 eingeht (Bit 4 und / oder Bit 5 muss dazu gesetzt sein)[1]. Bit 4 = Encoder auf Offsetwert bei fallender Flanke auf Anschluss 13 Bit 5 = Encoder auf Offsetwert bei steigender Flanke auf Anschluss 13 Bit 6 = Encoder auf Offsetwert stellen Bit 7 = Nicht benutzt – muss immer 0 sein
?? [1]	Dbyte 2	Teiler [u08]		0..15 = 00h..0Fh	Bit [3:0] = Vorteiler - 1 Bit [7:4] = Nicht benutzt – muss immer 0 sein
??	Dbyte 3	Offsetwert [u16]	High Byte	0..65535 = 0000h..FFFFh	Offsetwert, auf den der Zähler gesetzt wird, wenn Bit 6 im ersten Byte gesetzt ist oder ein externer Resetimpuls eingeht (mit Bit 4 oder 5 gesetzt).
??	Dbyte 4		LowByte		
??	Dbyte 5	Maximalwert [u16]	High Byte	0..65535 = 0000h..FFFFh	Maximal zulässiger Wert bei dem der Zähler auf 0 zurückspringt bzw. den er bei Werten kleiner 0 annimmt. Standard nach dem Einschalten ist FFFFh.
??	Dbyte 6		LowByte		
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
52	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

[1)]Befehl erweitert (ab V1.31)

7.8.10. Befehl 61 PWM Ausgang einstellen

Es müssen mindestens 5 Datenbytes übertragen werden (Flags, Periode, erster Duty - Cycle). Die danach folgenden Parameter sind optional.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
??	LENbyte	Längenbyte [u08]		2..255	
61	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Konfiguration [u08]		0..255 = 00h..FFh	Bit 0 = 0/1: Duty - Cycle Low / High Bit 1 = 0/1 Pullup zur Betriebsspannung aus / ein [1)] Bit 2 = PWM - Quelle ist PID - Regler (erster Duty - Cycle entfällt) Bit 3 = Nicht benutzt – muss immer 0 sein Bit 4 = 0/1: Polarität des UART - Empfangs ist Low - aktiv / High - aktiv Bit 5 = 0/1: UART - Empfang auf PWM - Ausgang / PWM - Eingang Bit 6 = Nicht benutzt – muss immer 0 sein Bit 7 = Erkennung Massetasten auf PWM - Ausgang erlauben
??	Dbyte 2	Periodendauer [u16]	High Byte	0..65535 = 0000h..FFFFh	Periodendauer in 1µs - 1
??	Dbyte 3		Low Byte		
??	Dbyte 4	1. Duty-Cycle [u16]	High Byte	0..65535 = 0000h..FFFFh	1. Duty-Cycle in 1µs
??	Dbyte 5		Low Byte		
??	Dbyte 6	Dauer [u08]		0..255 = 00h..FFh	Dauer des 1. Duty - Cycle in Perioden - 1

…					
??	Dbyte n-1	n. Duty-Cycle [u16]	High Byte	0..65535 = 0000h..FFFFh	Letzter Duty - Cycle in 1µs
??	Dbyte n		Low Byte		
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
61	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

[1)] Befehl erweitert ab V1.30, Nutzung erfordert Hardwareversion 1.1

7.8.11. Befehl 64 Kommunikation von DUT lesen

Gibt die im Eingangsbuffer befindlichen Zeichen zurück.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
02	LENbyte	Längenbyte [u08]		2..255	
64	CMDbyte	Befehlscode [u08]			
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
64	CMDbyte	Befehlscode			
??	Dbyte n	DATA [u08]			
$0D	Term.	Terminierung			

7.8.12. Befehl 65 Kommunikation mit DUT schreiben

Gibt die Daten seriell mit der gewählten PWM - Polarität auf dem PWM - Ausgang aus. Die PWM muss dazu inaktiv sein (PWM 0%).

Wurde im Befehl 67 Bit 14 gesetzt, so wird der Nachricht ein LIN - kompatibles BREAK - Signal vorangesetzt.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
06	LENbyte	Längenbyte [u08]		2..255	
65	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Daten [u08]			
??	Dbyte n	Daten [u08]			
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
65	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

7.8.13. Befehl 66 Baudrate für Kommunikation von / zu DUT lesen

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
02	LENbyte	Längenbyte [u08]		2..255	
66	CMDbyte	Befehlscode [u08]			
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
66	CMDbyte	Befehlscode			
??	Dbyte 1	Autobaud Bitdauer [u10]	High Byte	 0..1023 = 0000h..03FFh	Bit 15: 0/1 = Autobaud beendet / aktiv Bit 14: 1 = BREAK wird vor jeder Nachricht gesendet Bit [13:10]: Reserviert Bit [9:0]: Bitzeit in 1 µs – 1
??	Dbyte 2		Low Byte		
$0D	Term.	Terminierung			

7.8.14. Befehl 67 Baudrate für Kommunikation von / zu DUT einstellen

Setzt die Baudrate in µs. Auf Wunsch kann vor jeder Nachricht (Befehl 65) ein LIN - kompatibles BREAK - Signal erzeugt werden. Die Autobaud - Funktion misst die Länge des ersten Startbits das empfangen wird, nachdem Bit 15 gesetzt wurde. Dazu muss das LSB des nächsten empfangenen Bytes gesetzt sein. Nach dem Empfang wird das Bit 15 automatisch gelöscht (die Messung findet nur einmal statt). Eine Plausibilitätskontrolle erfolgt nicht.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
06	LENbyte	Längenbyte [u08]		2..255	
67	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Autobaud Bitdauer [u10]	High Byte	 0..1023 = 0000h..03FFh	Bit 15: 0/1 = Autobaud abbrechen / ausführen. Bit löscht sich selbst, nachdem Autobaud ausgeführt wurde. Bit 14: 1 = BREAK vor jeder Nachricht senden Bit [13:10]:Nicht benutzt – muss immer 0 sein Bit [9:0]: Bitzeit in 1µs – 1 (z.B. 1A0h für 2398 Baud; Standardwert nach Reset)
??	Dbyte 2		Low Byte		
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
67	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

7.8.15. Befehl 70 Motorstatus lesen

Befehl:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
02	LENbyte	Längenbyte [u08]		2..255	
70	CMDbyte	Befehlscode [u08]			
$0D	Term.	Terminierung			

Antwort:

Data	*Parameter*	*Beschreibung*	*Wertigkeit*	*Wertebereich*	*Infos*
$12	Trig.	Triggerzeichen			
70	CMDbyte	Befehlscode			
??	Dbyte 1	DATA [u16]	High Byte		Bit 15: Kurzschluss des Motor + Anschlusses zu U_{batt} im Zustand AUS Bit 14: Kurzschluss des Motor + Anschlusses zu GND im Zustand AUS Bit 13: Gerät war seit der letzten Abfrage stromlos (Power - on Reset) [2)] Bit 12: Nicht benutzt Bit [8:11]: Überstrom in einem Ausgangstransistor
??	Dbyte 2		Low Byte		Bit 7: Brücke eingeschaltet Bit 6: Übertemperatur - Abschaltung Bit 5: Temperaturwarnung (Information: Strom wird ab Treibertemperatur 160°C mit steigender Temperatur zunehmend begrenzt (2,5 A @ 175°C) Bit 4: Strombegrenzung erreicht Bit 3: Nicht benutzt Bit 2: Nicht benutzt Bit 1: Nicht benutzt Bit 0: Keine Last im Zustand AUS
$0D	Term.	Terminierung			

7.8.16. Befehl 71 Motor einstellen

Der „Open“ - Betrieb ermöglicht eine Open - Collector oder Open - Emitter Konfiguration des Motortreibers. Dabei wird während der Low - Phase der PWM der Motortreiber ausgeschaltet.

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
0C	LENbyte	Längenbyte [u08]		2..255	
71	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Periodendauer [u16]	High Byte	0..65535 = 0000h..FFFFh	Periodendauer in 2µs - 1
??	Dbyte 2		Low Byte		
??	Dbyte 3	Duty-Cycle [u16]	High Byte	0..65535 = 0000h..FFFFh	Duty-Cycle in 2µs
??	Dbyte 4		Low Byte		
??	Dbyte 5	Konfiguration [u08]		0..255 = 00h..FFh	Bit [1:0] = Strombegrenzung: 00 = 2,5 A 01 = 4 A 10 = 6,6 A 11 = 8,6 A Bit 2 = „Kickstart“: Überstromabschaltung für kapazitive Last für 50 ms nach Freigabe [3)] Bit 3 = „Open“ - Betrieb [1)] Bit 4 = PWM - Quelle ist PID - Regler Bit 5 = Motor Drehrichtung vorwärts Bit 6 = Motor freigegeben Bit 7 = 5V - Versorgung für Hall - Sensor
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
71	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

[1)] Befehl erweitert (ab V1.10)
[2)] Befehl erweitert (ab V1.31)
[3)] Befehl erweitert (ab V1.33)

7.8.17. Befehl 81 PID-Regler stellen

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
20	LENbyte	Längenbyte [u08]		2..255	
81	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Konfiguration		0..255 = 00h..FFh	Bit [1:0] = Eingangsquelle für PID - Regler: 00: Drehmoment 01: Hallsignal 10: SSI - Eingang 11: Encoder - Eingang Bit [6:2] = Nicht benutzt – muss immer 0 sein Bit 7 = Regler zurücksetzen und abschalten
	Dbyte 2	Proportional - Koeffizient [u16]	High Byte	0.. 65535 = 0000h..FFFFh	
??	Dbyte 3		Low Byte		
??	Dbyte 4	Integral - Koeffizient [u16]	High Byte	0..65535 = 0000h..FFFFh	
??	Dbyte 5		Low Byte		
??	Dbyte 6	Derivativ - Koeffizient [u16]	High Byte	0..65535 = 0000h..FFFFh	
??	Dbyte 7		Low Byte		
??	Dbyte 8	Oberes Integral - Limit [u16]	High Byte	0..65535 = 0000h..FFFFh	Größter möglicher I - Anteil am Ausgangswert / I - Koeffizient
??	Dbyte 9		Low Byte		
??	Dbyte 10	Unteres Gesamt - Limit [s16]	High Byte	-32768..32767 = 8000h..7FFFh	Minimaler Ausgangswert des PID-Reglers / 2
??	Dbyte 11		Low Byte		
??	Dbyte 12	Oberes Gesamt - Limit [s16]	High Byte	-32768..32767 = 8000h..7FFFh	Maximaler Ausgangswert des PID-Reglers / 2
??	Dbyte 13		Low Byte		
??	Dbyte 14	Regelperiode [u14]	High Byte	0..16383 = 0000h..3FFFh	Regelperiode für I - und D - Anteil in 10 µs - 1
??	Dbyte 15		Low Byte		
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
81	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			

7.8.18. Befehl 82 Sollwert für PID-Regler einstellen

Befehl:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
06	LENbyte	Längenbyte [u08]		2..255	
82	CMDbyte	Befehlscode [u08]			
??	Dbyte 1	Sollwert [u16]	High Byte	0..65535 = 0000h..FFFFh	
??	Dbyte 2		Low Byte		
$0D	Term.	Terminierung			

Antwort:

Data	Parameter	Beschreibung	Wertigkeit	Wertebereich	Infos
$12	Trig.	Triggerzeichen			
82	CMDbyte	Befehlscode			
$0D	Term.	Terminierung			