Manual

Brush Motor Controller

GET IN touch
WITH SENSITIVE TESTING

Softline

Modline

Conline

Boardline

Avidline

Pixline

Application

MCD Elektronik GmbH
Hoheneichstr. 52
75217 Birkenfeld
Tel. +49 (0) 72 31/78 405-0
Fax +49 (0) 72 31/78 405-10
info@mcd-elektronik.de
www.mcd-elektronik.com
HQ: Birkenfeld, Germany
Managing CEO: Bruno Hörter
Register Court Mannheim
HRB 505692

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

Table of Contents

1. General

This device is a controller module for brush motors and was developed for the control of flap actuators in automotive applications. It is also intended to simplify systems for force / displacement measurement. It provides a variety of interfaces and functions.

Order number: # 123114

2. Extent of Delivery

1 x brush motor controller

1 x USB storage card with installation software

3. Pin Assignment

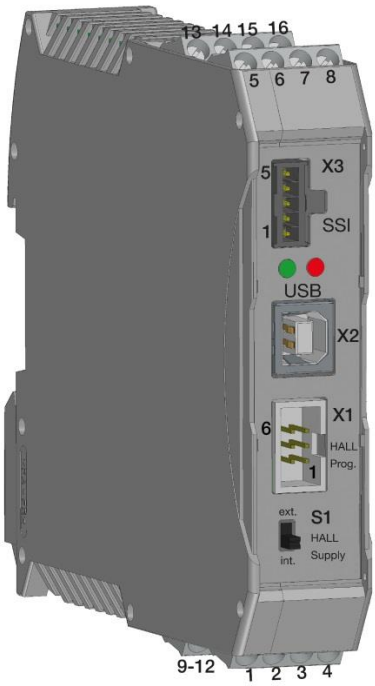


Figure 1: Pin Assignments

Terminal Assignments

Pin No.	**13**	**14**	**15**	**16**
Signal	UART In / AUX / Decoder Z	GND	Decoder A	Decoder B
Pin No.	**5**	**6**	**7**	**8**
Signal	Torque in	GND	GND	PWM Out
Pin No.	**1**	**2**	**3**	**4**
Signal	Supply Hall	Signal Hall	GND Hall	Shield
Pin No.	**9**	**10**	**11**	**12**
Signal	UB/ KL30 (7 - 18 V)	GND/ KL31	Motor +/ KL30 out	Motor -/ KL31 out

Connector Pin Assignments

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. Features

- **Motor driver**
  - Forwards and reverse operation
  - Peak current limitation configurable at four levels
  - Automatic shutoff on short circuit or overheating
  - Symmetrical PWM with adjustable frequency and duty cycle
  - Measurement of motor current between the H - bridge and the motor
    Note: The current through the motor is measured, **not** the current in the supply line!
  - "Open" operation permits open - collector and open - emitter configurations

- **PWM output**
  - Small signal transistor output with short circuit current limitation
  - 2 kΩ pull - up resistance can be connected to operating voltage (from hardware version 1.1)
  - Functions for compensation and communication with flap actuators

- **Programmer connection**
  - Connection for a hall sensor programming unit

- **Power supply for hall sensor**
  - Precision 5 V power supply can be selected with a switch. To be used when no programming unit is connected.

- **SSI connection for position sensor**

- **Input for quadrature encoder or pulse encoder with directional signal**
  - Optionally with reset input
  - Optional prescaler extends the counter range

- **Analog inputs**
  - Torque sensor: Measuring range ± 10 V
  - Power supply for hall sensor: Measuring range 10 V (control measurement)
  - Signal from hall sensor: Measuring range 5 V
  - AUX measurement input: Measuring range: 18 V (from hardware version 1.1)
  - Acquisition of signals with up to 131071 measurement points, optionally synchronous with the quadrature encoder / pulse encoder (force / displacement synchronous)
  - Pre - trigger and post - trigger options; triggering of measured value acquisition on any analog or position input

- **PID controller**
  - Trigger behavior and control range largely configurable
  - Control output can be switched to PWM output or motor driver
  - Control on input signal:
    - Hall sensor
    - Torque sensor
    - Position (quadratur encoder / pulse encoder or SSI)

5. Functions

5.1. Motor Driver

The motor driver has a peak current limiter, that can be set to one of four levels, and a measurement system to measure the momentary motor current. It shuts off on short circuit or overheating. These events are indicated by a red LED on the front panel, and can also be queried via control command from a higher - level device. The frequency and duty cycle of the associated PWM generator can be adjusted. If desired, the PID controller can control the duty cycle. The desired PWM frequency is determined separately.

In the "open" mode, the motor driver is turned off during the inactive time of the PWM signal. This permits the use of open - emitter and open - collector configurations. In this case the motor current may flow through free - wheel or overvoltage protection diodes. This can increase the heat generation in the motor driver and may reduce its maximum load capacity.

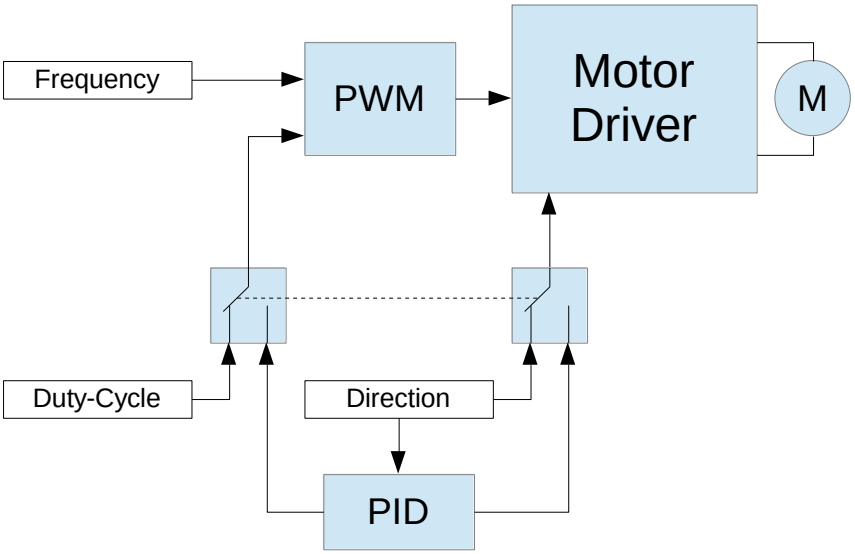


Figure 2: Illustration of the Motor Driver

5.2. PWM Output

The PWM output is a low - power control output with a 2 kΩ pull - up resistor that can be connected to the operational power. Just as for the motor driver, the frequency and duty cycle of the PWM output are configurable. Up to 40 different duty cycles with a length of up to 256 periods each can be output in succession. The last one is then issued permanently until the next change. This is a function required for some flap actuators to trigger special functions. The polarity of the duty cycle can be low or high.

The following additional functions are also supported:

- If the external control signal is pulled low for longer than 200 ms, the PWM signal shuts off and a receiver listens to the line for asynchronous serial signals with a configurable baud rate. Characters received can be retrieved by the higher - level device via control command. The PWM stays off until it is started again by command. This functionality can also be disabled. The momentary state of the PWM line can be queried (see chapter 5.8).

- The output can also output serial data. To do this, the duty cycle of the PWM must be at 0%, otherwise the two signals would interfere. The output uses the polarity of the duty cycle.

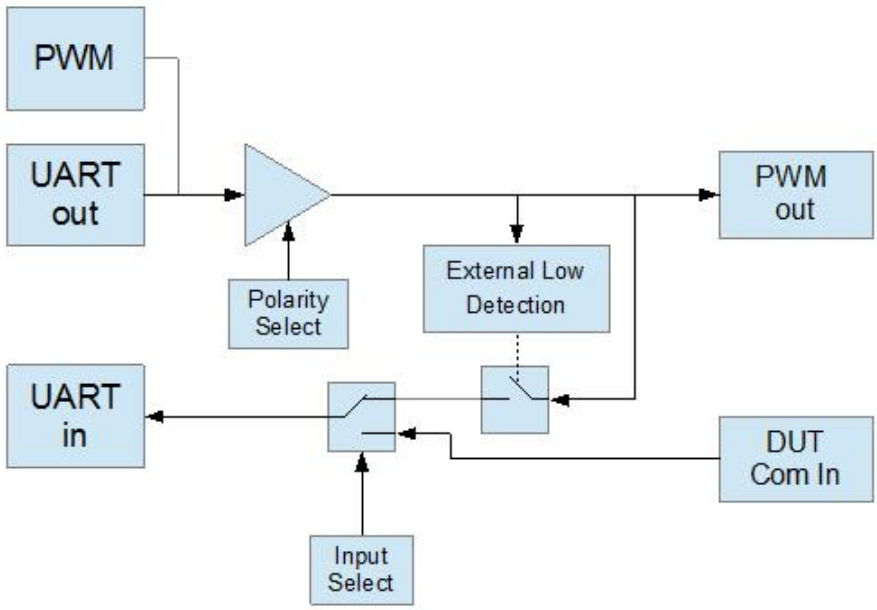


Figure 3: Illustration of the PMW Output

5.3. Communication Input

The communication input can optionally take over the function of the serial receiver in the PWM output. The 200 ms wait on “pulling low” is omitted here, however receiving is continuous.

5.3.1. Baud Rate Detection

The communication input has a function to auto detect the baud rate. Is this function activated, the start bit is measured once and the baud rate will be set. In order for this to work, the LSB of the byte to be sent must be set (e.g. 0 x 01, 0 x 55 etc.).

The calculated baud rate can be read back for a plausibility check. It is possible that the first transferred byte may not be read correctly. The following byte should usually be correctly detected if the deviation is less than 10%. This function can also be used to measure the duration of a single pulse (high or low) in the range of 1 to 1024 µs. This also makes it possible, for example, to close pulses from the nominal value to clock deviations at UART via the deviation of PWM pulses and to set the construction rate manually in advance. Alternatively, the baud rate can also be determined by measuring streaming and manual calculation.

5.3.2. Generation BREAK Symbol

To support LIN communication, the brush motor control is able to generate a BREAK symbol at the start of each data packet. The length of the BREAK depends on the selected baud rate.

5.4. Programming Unit Connector

On the front side there is a connection for hall sensor programming units from the company Micronas. The signals are passed through and can be forwarded to the DUT using a shielded sensor cable. The power supply for the sensor from the programming unit can be measured for inspection. The sensor signal can also be measured and sent to the PID controller (see chapter 5.9) as the actual signal.

5.5. Power Supply for Hall Sensor

If no programming unit is available to provide power to the hall sensor, the sliding switch on the front panel can be used to enable an internal precision power supply.

This power source can also be turned on and off by control command.

5.6. SSI Connection for Position Sensor

The synchronous serial interface provides a different clock signal at a fixed frequency and reads the differential signal provided by the connected, compatible sensor.

5.7. Input for Quadrature Encoder or Pulse Encoder with Directional Signal

The input for quadrature encoders is a transistor input and triggers on input signals over a wide range of voltages. If the quadrature encoder only has open - collector outputs, external pull - up resistors are required. The input can be switched using a control command so that it can also count pulses with a directional signal. The polarity of the directional signal is configurable. The counter value can be set / reset to a desired value at any time.

There is also a configurable prescaler and an external reset input. *DUT COM In* is used for the external reset signal. It can reset the counter to the starting value on either a rising or falling edge. To support continuous rotating movements, a maximum counter value can also be determined which resets the counter to zero when exceeded, or resets the counter to the maximum when it falls below zero.

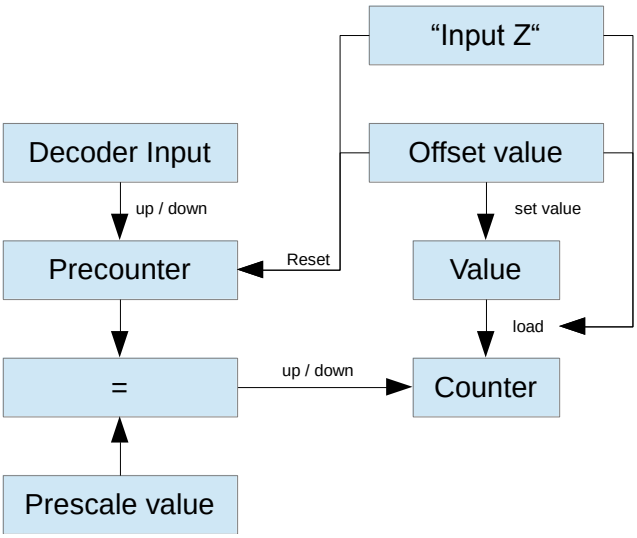


Figure 4: Illustration of the Input for Quadrate Encoder or Pulse Encoder

5.8. Analog Inputs

5.8.1. Data Sources

There are four analog inputs: Torque or force sensor, motor current, hall signal and power supply for hall sensor. Only the input for the torque or force sensor has a dedicated connection. All others are connected internally to the corresponding lines.

Both individual values or up to 131071 values can be acquired. If more than one signal source will be acquired, the maximum number of values that can be entered in a stream is reduced correspondingly:

$$Max.\ Number\ of\ Values = \frac{131071}{Number\ of\ Channels}$$

If all four analog signals are acquired, then at most 32727 value quartets will be recorded.

In addition to the four analog inputs, the following channels are also available:

Counter state of quadrature input, the value of the SSI input and the state of all digital input lines (including the PWM output) and motor controlling signals.

A table with all acquired digital signals can be found in the description of Command 41(chapter 7.8.7).

5.8.2. Data Acquisition Control

Measured data can be recorded on a time - dependent basis with a configurable sampling rate or on a displacement - dependent basis synchronous with the quadrature encoder / pulse counter input. The maximum counter frequency may not exceed 100 kHz. Otherwise, pulses will be lost.

Warning: For a quadrature encoder, this means a maximum signal of 25 kHz!

A condition can also be defined which triggers the acquisition of measured values (Triggering). This is typically the rising or falling edge of a changing input signal at a certain voltage. The trigger can come before, after, or at a certain position in the measured data stream. It can also be configured whether first the specified number of measured values must be acquired before a trigger condition is evaluated, or it is more important to record the first trigger condition. If the trigger condition is not met or the acquisition does not terminate due to missing or insufficient counter pulses, the measurement is terminated by a configurable timeout and the measured data recorded to that point are returned.

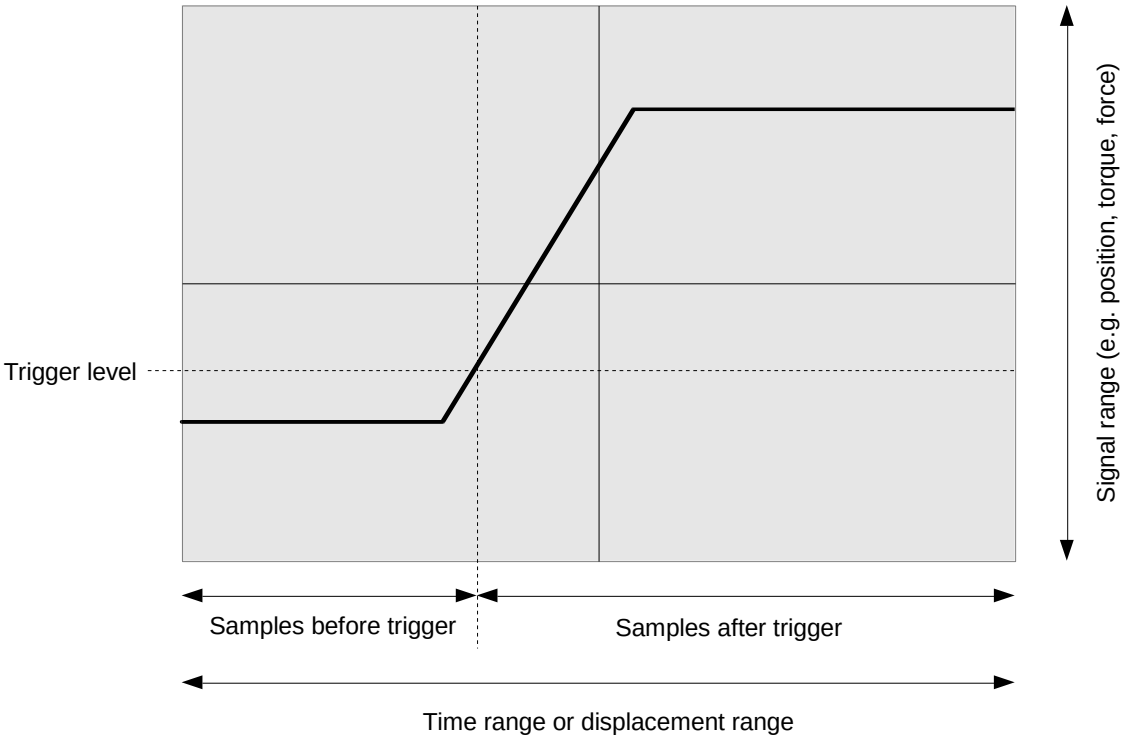
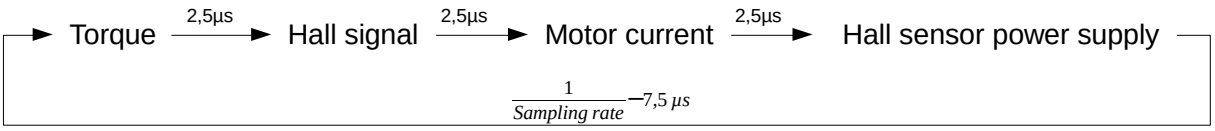
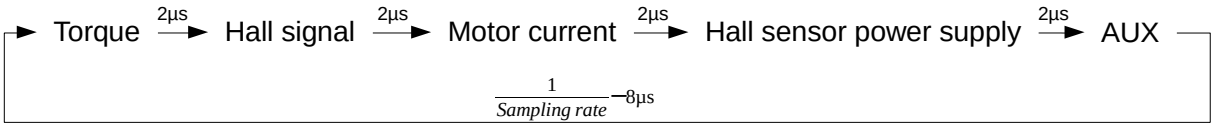


Figure 5: Data Acquisition Control

The analog measured values are recorded in the background in a fixed order with a defined time reference:



For models with AUX - IN:



5.9. PID Controller

The PID controller can be influenced not only by its three coefficients, but also the temporal triggering vehicle of the I - and D - components. The analog measured values for torque / force and the hall signal and the counter value of the quadrature encoder and the SSI input can be used as input values. The I - component and the final output value can be limited. The output value is used as the duty cycle on the PWM of the motor driver and / or the PWM output. If negative output values are permitted, the motor driver actively drives in both directions. Negative values should not be permitted on the PWM output, since a PWM signal cannot represent polarity. For a negative output value, the PWM therefore starts to rise again.

5.10. EEPROM Table

Usage of the EEPROM:

EEPROM address	Firmware bis V1.4x	Firmware V1.5x
00	unused	Offset alignment for current measurement
01…63	unused	unused

Preallocated memory positions should not be overwritten. Infringement can lead to malfunction (measurement failure) which may not always be noticed.

6. Technical Data

Electrical Characteristics		
Operating voltage	7 – 18 VDC	The absolute maximum is 19 V For a correct reset the voltage must be interrupted for ≥ 1 s
Power consumption	< 1 W	Typical current consumption < 60 mA
Power supply for hall sensor	5 V ± 0,2% max.; 20 mA max.	Warning: a short circuit on the output can have a negative influence on the specified value until the controller has cooled down again!
PWM Output	Open collector output with 100 kOhm pull - up resistor to the operating voltage	Pull - down current at 12 V: about 13 mA External pull - up depending on the application
SSI: synchronous serial interface	Beat: 500 kHz Resolution: 16 Bit Termination resistance: 10 kOhm	
Analog input	Maximum sample rate: 100 kSps	Input resistance: 100 kOhm
Analog measuring ranges	Hall signal: 0…5,12 V Hall supply: 0…10,24 V Force sensor: ± 10,24 V Current: 0…5,12 V AUX (UART In) see below	Nominal values without calibration 0,048…4,048 V ≙ - 8 A…+ 8 A
Quadrature encoder input	Input impedance: 20 kOhm Low level: < 0,8 V High level: > 2,5 V Input frequency: max. 100 kHz max. 50 kHz	 Max. 30 V At TTL level At 10 V_p (At higher voltage levels it is possible to use an input resistor of approx. 20 kΩ / 5 V to increase the limiting frequency)
Input UART In / Decoder - Z	Input impedance: 1 MOhm Digital: Low level: < 3,3 V High level: > 3,3 V Analog: 0,1 V…18,7 V	 Max. 30 V Gain error ± 2% max. (uncalibrated)
Mechanical Characteristics		
Dimensions (H x W x D)	116 mm x 22 m x 115 mm	Without plug connected
Connections	Screw terminals	
	1 x USB - B 2.0	Control Input
	Header 2 x 3 pins	Connection to programming unit for hall sensor
	Header Samtec IPL1 1 x 5 pins	SSI: Interface for position sensor
Other Characteristics		
Control	Virtual serial port via USB Galvanically isolated	3 MBaud 1 Startbit 1 Stopbit No handshake
Display	1 x LED green	Operation / activity
	1 x LED red	Motor driver warning / indicator
Operating temperature	- 40 – 85°C	Motor driver up to 125°C
Weight	130 g	

7. Interface Description

7.1. General Definitions

- Start character is sent as $12
- Length byte is the number of transmitted ASCII characters (Char) starting with the CMD byte
- As end and termination signal $0D is sent

7.2. General Standard Command Syntax

Command: „?“ corresponds to ASCII signs

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Triggersign			Trigger sign for interface
??	LENbyte	Length byte [u08]		2..255	Number of ASCII signs beginning with CMD byte to data byte n
??	CMDbyte	Command code [u08]			Command code see command
??	Dbyte1				
$0D	Term.	Termination			End sign

7.3. General Response

Response: with correct command parameters

Data	*Parameters*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			Trigger sign for interface
??	CMDbyte	Command code [u08]			Repetition of command code in response
??	Dbyte1				
$0D	Term.	Termination			Final sign

Response: with error in command parameters

Data	*Parameters*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			Trigger sign for interface
FF	CMDbyte	ERROR [u08]			Default "FF" for errors
??	Errorbyte	Error code [u08]			Error code
$0D	Term.	Termination			End sign

7.4. Maximum Data Length of Transmission

For commands with no fixed length, no more than **127 data bytes** (254 ASCII signs high byte / low byte) can follow the command byte (CMDbyte).

A command with measurement data (Command 40 – chapter 7.8.6) is an exception to this rule. After completion of the command string, the measurement data are transmitted in binary form.

7.5. Type Definitions

Data Type	*Abbreviation*
unsigned char	u08
unsigned short	u16
signed short	s16
unsigned long	u32

7.6. ERROR Codes

Data Type	*Abbreviation*
ERROR	0x0F
NOCMD	0x01
PARAMS	0x03
VALUERANGE	0x04
CMDLEN	0x05

7.7. Command Implementation for Different Software Versions

Command	Description	Firmware Version							
		1.00	1.10	1.20	1.30	1.31	1.32	1.4	1.6
20	Write / read configuration memory	X	X	X	X	X	X	X	X
2F	Unlock configuration memory	X	X	X	X	X	X	X	X
22	Read EEPROM	X	X	X	X	X	X	X	X
23	Write EEPROM	X	X	X	X	X	X	X	X
3F	Read version of firmware	X	X	X	X	X	X	X	X
40	Read out measurement data	X	X	X	X	X	X	X	X
41	Trigger measured data acquisition	X	X	X	X[2)]	X	X	X[1)]	X
50	Read encoder counter value	X	X	X	X	X	X	X	X
51	Set encoder	X	X						
52	Configure encoder	X	X[1)]	X[1)]	X	X[1)]	X	X	X
61	Set PWM output	X	X[1)]	X	X[2)]	X	X	X	X
64	Read communication from DUT	X	X	X	X	X	X	X	X
65	Write communication to DUT	X	X	X	X	X	X	X	X
66	Read baud rate for communication from / to DUT								X
67	Set baud rate for communication from / to DUT	X	X	X	X	X	X	X	X
70	Read motor driver status	X	X	X	X	X[1)]	X	X	X
71	Set motor driver	X	X[1)]	X	X	X	X[1)]	X	X
81	Set PID controller	X	X	X	X	X	X	X	X
82	Set setpoint for PID controller	X	X	X	X	X	X	X	X

[1)] Command extended
[2)] Command extended, requires hardware version 1.1 for use

7.8. Set of Command

7.8.1. Command 20 Write / Read Configuration Memory

This command is provided exclusively for firmware updates and should never be used in normal operation!

If the memory is blocked, error code FF04 is returned.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
??	LENbyte	Length byte [u08]		2..255	
20	CMDbyte	Command code [u08]			
??	Dbyte 1	DATA [u08]		0..255 = 00h..FFh	Data to send
??	…	DATA [u08]		0..255 = 00h..FFh	
??	Dbyte x	DATA [u08]		0..255 = 00h..FFh	
$0D	Term.	Termination			

Response:

Data	*Parameters*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
20	CMDbyte	Command code			
??	Dbyte 1	DATA [u08]		0..255 = 00h..FFh	Data read
??	…	DATA [u08]		0..255 = 00h..FFh	
??	Dbyte x	DATA [u08]		0..255 = 00h..FFh	
$0D	Term.	Termination			

7.8.2. Command 22 Read EEPROM

Reads a 16 - bit word from the EEPROM. If an invalid address is given, error code FF04 is returned.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
04	LENbyte	Length byte [u08]		2..255	
22	CMDbyte	Command code [u08]			
??	Dbyte	Address [u08]		0..63 = 00h..3Fh	
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
22	CMDbyte	Command code			
??	Dbyte 1	DATA [u08]		0..255 = 00h..FFh	Data read
??	Dbyte 2	DATA [u08]		0..255 = 00h..FFh	
$0D	Term.	Termination			

7.8.3. Command 23 Write EEPROM

Writes a 16 - bit word to the EEPROM. If an invalid address is given, error code FF04 is returned.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
08	LENbyte	Length byte [u08]		2..255	
23	CMDbyte	Command code [u08]			
??	Dbyte 1	DATA [u08]		0..63 = 00h..3Fh	Address
??	Dbyte 2	DATA [u16]	High Byte	0..65535 = 0000h..FFFFh	Data word
??	Dbyte 3		Low Byte		
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
23	CMDbyte	Command code			
$0D	Term.	Termination			

7.8.4. Command 2F Unlock Configuration Memory

Enables access to the configuration memory. Any command other than command 20 will block the access again.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
04	LENbyte	Length byte [u08]		2..255	
2F	CMDbyte	Command code [u08]			
??	Dbyte	DATA [u08]		0..255 = 00h..FFh	55h = Configuration memory is unlocked Any other value blocks the memory
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
2F	CMDbyte	Command code			
$0D	Term.	Termination			

7.8.5. Command 3F Read SW VERSION of Firmware

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
02	LENbyte	Length byte [u08]		2..255	
3F	CMDbyte	Command code [u08]			
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
3F	CMDbyte	Command code			
??	Dbyte 1	DATA [u08]			Text string of the version
??	…	DATA [u08]			
??	Dbyte x	DATA [u08]			
$0D	Term.	Termination			

7.8.6. Command 40 Read Measurement Data

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
04	LENbyte	Length byte [u08]		2..255	
40	CMDbyte	Command code [u08]			
??	Dbyte 1	Number of records [u08]		0..255 = 00h..FFh	Bit 0: 0 = Check for end of measurement and return data (If not completed, error 04h is returned) 1 = Cancel measurement and return the existing data Bit [7:1] = Not used – must always be 0
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
40	CMDbyte	Command code			
??	Dbyte 1	DATA [u17]	High Byte	0..131071 = 000000h..01 FFFFh	Number of record before the trigger
??	Dbyte 2				
??	Dbyte 3		Low Byte		
??	Dbyte 4	DATA [u17]	High Byte	0..131071 = 000000h..01 FFFFh	Number of record after the trigger
??	Dbyte 5				
??	Dbyte 6		Low Byte		
$0D	Term.	Intermediate termination			
$??	Dbyte 7	First record, first value	High Byte	0..65535 = 0000h..FFFF h	
$??	Dbyte 8		Low Byte		
…					
$??	Dbyte n-1	Last record; last value	High Byte	0..65535 = 0000h..FFFF h	
$??	Dbyte n		Low Byte		
$0D	Term.	Termination			

7.8.7. Command 41 Trigger Measurement Data Acquisition

At most 131071 data words can be acquired. Every measurement source requires one data word per sample. The sampled data remains in storage until they are read out with command 40 or a new acquisition run is started.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
1A	LENbyte	Length byte [u08]		2..255	
41	CMDbyte	Command code [u08]			
??	Dbyte 1	Number of records [u17]	High Byte	0..131071 = 000000h..01 FFFFh	Number of records to read before trigger

??	Dbyte 2				
??	Dbyte 3		Low Byte		
??	Dbyte 4	Number of records [u17]	High Byte	0..131071 = 000000h..01FFFFh	Number of records to read after trigger
	Dbyte 5				
??	Dbyte 6		Low Byte		
??	Dbyte 7	Trigger threshold [u16] / s[16]	High Byte	0..65535 = 0000h..FFFFh -32768..32767 = 8000h..7FFFh	Trigger threshold, with or without sign depending on the chosen trigger source: Input hall sensor signal and the hall sensor power supply are unsigned, all others are signed Value is ignored for PWM inputs
??	Dbyte 8		Low Byte		
??	Dbyte 9	Data source [u08]		0..255 = 00h..FFh	Bit 0 = Read Input of torque sensor Bit 1 = Read motor current Bit 2 = Read hall sensor signal Bit 3 = Read hall sensor power supply Bit 4 = Read encoder input Bit 5 = Read data SSI interface Bit 6 = State of digital inputs and outputs Bit 7 = Acquire analog value from PWM / UART / Decoder - Z input (Pin 13)[2)]
??	Dbyte 10	Trigger source [u08]		0..255 = 00h..FFh	00h = Torque sensor input 01h = Motor current input 02h = Hall sensor signal input 03h = Hall sensor power supply input 04h = Encoder input 05h = SSI interface input 06h = PWM input / output 07h = PWM / UART / Decoder - Z input (Pin 13) analog value [2)]

??	Dbyte 11	Trigger configuration [u08]		0..255 = 00h..FFh	Bit 0 = 0/1: Start on trigger / start immediately Bit 1 = 0/1: Trigger edge falling / rising Bit 2 = Trigger cannot fire before the specified quantitiy of data has been recorded Bit 3 = Not used – must always be 0 Bit 4 = Synchronize data acquisition with encoder input Bit [7:5] = Not used – must always be 0
??	Dbyte 12	Sample rate [u08]		0..255 = 00h..FFh	Bit [4:0] = Divider für sample rate -1 [3)] Bit [7:5] = Not used – must always be 0
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
41	CMDbyte	Command code			
$0D	Term.	Termination			

[2)] Command extended in V1.30 and up, requires hardware version 1.1 for use
[3)] Divider extended from 4 bit to 5 bit in V1.40 and up

Bit assignment of digital inputs and outputs:

Bit 0	Encoder input B
Bit 1	Encoder input A
Bit 2	SSI input
Bit 3	PWM_IO
Bit 4	DUT Com In
Bit 5	Motor PWM
Bit 6	Motor direction
Bit 7	Motor on /off
Bit [15:8]	Not used

Please note that the digital signals may change more quickly than they are read in, which can lead to falsification of the signal (undersampling).

7.8.8. Command 50 Read Encoder Value

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
02	LENbyte	Length byte [u08]		2..255	
50	CMDbyte	Command code [u08]			
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
50	CMDbyte	Command code			
??	Dbyte 1	DATA [u16]	High Byte	0..65535 = 0000h..FFFFh	Counter value of quadrature encoder / pulse counter
??	Dbyte 2		Low Byte		
??	Dbyte 3	DATA [u16]	High Byte	0..65535 = 0000h..FFFFh	Value of SSI
??	Dbyte 4		Low Byte		
$0D	Term.	Termination			

7.8.9. Command 52 Configure Encoder

The encoder input can work as a quadrature encoder or a pulse counter with direction signal. A hardware reset can optionally be defined for the counter value. The value is then reset to the offset value (the standard value is zero). The counter can also be manually set to the offset value.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
0E	LENbyte	Length byte [u08]		2..255	
52	CMDbyte	Command code [u08]			
??	Dbyte 1	Configuration [u08]		0..255 = 00h..FFh	Bit 0 = 0/1: Quadrature enoder / pulse counter with direction signal Bit 1 = Not used – must always be 0 Bit 2 = 0 / 1: Counting direction normal / reversed Bit 3 = Encoder returns 0000h until an edge comes in on connection 13 (Bit 4 and / or Bit 5 must be set in this case) [1)]. Bit 4 = Encoder to offset value on falling edge on connection 13 Bit 5 = Encoder to offset value on rising edge on connection 13 Bit 6 = Set encoder to offset value Bit 7 = Not used – must always be 0
?? [1)]	Dbyte 2	Divider [u08]		0..15 = 00h..0Fh	Bit [3:0] = Predivider - 1 Bit [7:4] = Not used – msut always be 0
??	Dbyte 3	Offset value [u16]	High Byte	0..65535 = 0000h..FFFFh	Offset value to which the counter is set when Bit 6 in the first byte is set or an external reset pulse is received (with Bit 4 or Bit 5 set).
??	Dbyte 4		Low Byte		
??	Dbyte 5	Maximum value [u16]	High Byte	0..65535 = 0000h..FFFFh	Maximum permissible value at which the counter jumps to 0 or which it is assigned if the values fall below 0. The default after power is applied is FFFFh.
??	Dbyte 6		Low Byte		
$0D	Term.	Termination			

Respsone:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
52	CMDbyte	Command code			
$0D	Term.	Termination			

[1)] Command extended (V1.31 and up)

7.8.10. Command 61 Set PWM Output

At least 5 data bytes must be transmitted (flags, period, first duty cycle). The subsequent parameters are optional.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
??	LENbyte	Length byte [u08]		2..255	
61	CMDbyte	Command code [u08]			
??	Dbyte 1	Configuration [u08]		0..255 = 00h..FFh	Bit 0 = 0/1 Duty cycle low / high Bit 1 = 0/1 Pull - up to operating voltage off / on [1)] Bit 2 = PWM source is PID controller (first duty cycle omitted) Bit 3 = Not used – must always be 0 Bit 4 = 0/1 Polarity of UART receiver is low active / high active Bit 5 = 0/1: UART receiver on PWM output / PWM input Bit 6 = Not used – must always be 0 Bit 7 = Permit detection of ground samples on PWM output
??	Dbyte 2	Period length [u16]	High Byte	0..65535 = 0000h..FFFFh	Period length in 1 µs - 1
??	Dbyte 3		Low Byte		
??	Dbyte 4	1. Duty cycle [u16]	High Byte	0..65535 = 0000h..FFFFh	1. Duty cycle in 1 µs
??	Dbyte 5		Low Byte		

??	Dbyte 6	Duration [u08]		0..255 = 00h..FFh	Duration of the first duty cycle in periods -1
…					
??	Dbyte n-1	n. duty cycle [u16]	High Byte	0..65535 = 0000h..FFFFh	Last duty cycle in 1 µs
??	Dbyte n		Low Byte		
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
61	CMDbyte	Command code			
$0D	Term.	Termination			

[2)] Command extended in V1.30 and up; requires hardware version 1.1 for use

7.8.11. Command 64 Read Communication from DUT

Returns the characters stored in the input buffer.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
02	LENbyte	Length byte [u08]		2..255	
64	CMDbyte	Command code [u08]			
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
64	CMDbyte	Command code			
??	Dbyte n	DATA [u08]			
$0D	Term.	Termination			

7.8.12. Command 65 Write Communication to DUT

Outputs data serially on the PWM output with the selected PWM polarity. PWM must be inactive during this (PWM 0%).

If bit 14 was placed in the command 67, a LIN - compatible BREAK signal is preceded by the message.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
06	LENbyte	Length byte [u08]		2..255	
65	CMDbyte	Command code [u08]			
??	Dbyte 1	Data [u08]			
??	Dbyte n	Data [u08]			
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
65	CMDbyte	Command code			
$0D	Term.	Termination			

7.8.13. Command 66 Read Baud Rate for Communication from / to DUT

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigge sign			
02	LENbyte	Length byte [u08]		2..255	
66	CMDbyte	Command code [u08]			
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
66	CMDbyte	Command code			
??	Dbyte 1	Auto baud Bit duration [u10]	High Byte	0..1023 = 0000h..03FFh	Bit 15: 0/1 = Auto baud inactive / active Bit 14: 1 = Generate BREAK for each message Bit [13:10]: reserved Bit [9:0]: Bit time in 1 µs – 1
??	Dbyte 2		Low Byte		
$0D	Term.	Termination			

7.8.14. Command 67 Set Baud Rate for Communication from / to DUT

Sets the baud rate in µs. If desired, a LIN compatible BREAK signal can be generated prior to each message (command 65). The autobaud function measures the length of the first start bit received after bit 15 has been set. For this, the LSB of the next received byte must be set. After reception, bit 15 is automatically cleared (the measurement takes place only once). A plausibility check does not take place.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
06	LENbyte	Length byte [u08]		2..255	
67	CMDbyte	Command code [u08]			
??	Dbyte 1	Auto baud Bit duration [u10]	High Byte	0..1023 = 0000h..03FFh	Bit 15: 0/1 = Auto baud cancel / start self - clearing bit after executing auto baud Bit 14: 1 = Generate BREAK for each message Bit [13:10]: no used – must always be 0 Bit [9:0]: Bit time in 1 µs – 1 (e.g. 1A0h für 2398 Baud; standard value after reset)
??	Dbyte 2		Low Byte		
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
67	CMDbyte	Command code			
$0D	Term.	Termination			

7.8.15. Command 70 Read Motor Status

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
02	LENbyte	Length byte [u08]		2..255	
70	CMDbyte	Command code [u08]			
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
70	CMDbyte	Command code			
??	Dbyte 1	DATA [u16]	High Byte		Bit 15: Short circuit of the motor + connection to V_{batt} in the state OFF Bit 14: Short circuit of the motor + Connection to GND in the state OFF Bit 13: Device was disconnected from power since last query (power - on reset) Bit 12: Not used Bit [8:11]: Overcurrent in an output transistor
??	Dbyte 2		Low Byte		Bit 7: Bridge active Bit 6: Excessive temperature switch - off Bit 5: Temperature warning (Information: current will be limited increasingly with rising temperature up to 160°C driver temperature (2,5 A @ 175°C) Bit 4: Current limit is reached Bit 3: Not used Bit 2: Not used Bit 1: Not used Bit 0: No load in the state OFF
$0D	Term.	Termination			

7.8.16. Command 71 Set Motor

"Open" operation permits open - collector or open - emitter configuration of the motor driver. The motor driver is then turned off during the low phase of the PWM.

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
0C	LENbyte	Length byte [u08]		2..255	
71	CMDbyte	Command code [u08]			
??	Dbyte 1	Period duration [u16]	High Byte	0..65535 = 0000h..FFFFh	Period duration in 2 µs - 1
??	Dbyte 2		Low Byte		
??	Dbyte 3	Duty cycle [u16]	High Byte	0..65535 = 0000h..FFFFh	Duty cycle in 2 µs
??	Dbyte 4		Low Byte		
??	Dbyte 5	Configuration [u08]		0..255 = 00h..FFh	Bit [1:0] = Current limit: 00 = 2,5 A 01 = 4 A 10 = 6,6 A 11 = 8,6 A Bit 2 = „Kick start“: Overcurrent protection for capacitive loads for 50 ms after enabled. [3] Bit 3 = “Open“ operation [1] Bit 4 = PWM source is PID controller Bit 5 = Motor rotation direction forwards Bit 6 = Motor enabled Bit 7 = 5 V power supply for hall sensor
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
71	CMDbyte	Command code			
$0D	Term.	Termination			

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

7.8.17. Command 81 Set PID Controller

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
20	LENbyte	Length byte [u08]		2..255	
81	CMDbyte	Command code [u08]			
??	Dbyte 1	Configuration		0..255 = 00h..FFh	Bit [1:0] = Input source for PID controller: 00: Torque 01: Hall signal 10: SSI Input 11: Encoder input Bit [6:2] = Not used – must always be 0 Bit 7 = Reset controller and turn it off
	Dbyte 2	Proportional coefficient [u16]	High Byte	0.. 65535 = 0000h..FFFFh	
??	Dbyte 3		Low Byte		
??	Dbyte 4	Integral coefficient [u16]	High Byte	0..65535 = 0000h..FFFFh	
??	Dbyte 5		Low Byte		
??	Dbyte 6	Derivative coefficient [u16]	High Byte	0..65535 = 0000h..FFFFh	
??	Dbyte 7		Low Byte		
??	Dbyte 8	Upper integral limit [u16]	High Byte	0..65535 = 0000h..FFFFh	Greatest possible I part in output value / I - coefficient
??	Dbyte 9		Low Byte		
??	Dbyte 10	Lower total limit [s16]	High Byte	-32768..32767 = 8000h..7FFFh	Minimum output value of the PID controller / 2
??	Dbyte 11		Low Byte		
??	Dbyte 12	Upper total limit [s16]	High Byte	-32768..32767 = 8000h..7FFFh	Maximum output value of the PID controller / 2
??	Dbyte 13		Low Byte		
??	Dbyte 14	Regular period [u14]	High Byte	0..16383 = 0000h..3FFFh	Regular period for I - and D - parts in 10 µs - 1
??	Dbyte 15		Low Byte		
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
81	CMDbyte	Command code			
$0D	Term.	Termination			

7.8.18. Command 82 Set Setpoint for PID Controller

Command:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
06	LENbyte	Length byte [u08]		2..255	
82	CMDbyte	Command code [u08]			
??	Dbyte 1	Setpoint [u16]	High Byte	0..65535 = 0000h..FFFFh	
??	Dbyte 2		Low Byte		
$0D	Term.	Termination			

Response:

Data	*Parameter*	*Description*	*Value*	*Value Range*	*Information*
$12	Trig.	Trigger sign			
82	CMDbyte	Command code			
$0D	Term.	Termination			