

# Motorized Capacitor

## Software protocol interface RS-232



### Document Information

---

Authors ..... O. Lehmann / A. Renggli / T. Fenske / T. Fink  
Document ..... SB-68-04.doc  
Created on ..... 25-July-2014  
Revision ..... 05  
Comment ..... Revised version

## Document history

Doc. Rev.	Firmware Rev.	Date	Author(s)	Change(s)	Status
1.0	1.0.x	2007-03-30	O. Lehmann	Initial document	released
1.1	1.1.1	2010-06-25	A. Renggli	Various additions	internal
02	1.2.x and 2.1.x	2013-11-27	T. Fink / T. Fenske	Additional firmware functions	Draft
02	1.2.x and 2.1.x	2013-12-04	T. Fink / T. Fenske	Various commands added; marked commands which are only available for firmware 2.X	preliminary
03	1.2.x and 2.1.x	2013-12-12	T. Fink / T. Fenske	Various modifications due to technical restraints during realization of firmware 2.X	Draft
03	1.2.x and 2.1.x	2014-01-22	T. Fink / T. Fenske	Reviewed with minor modifications / configuration section added	released
04	1.2.x and 2.1.x	2014-01-30	T. Fink / T. Fenske	Added command to move to a stored position (firmware 2.1.1)	released
05	1.2.x, 2.1.x and 2.2.x	2014-07-25	T. Fink / T. Fenske	Added several commands: Micro-stepping Customer Limits (Firmware 2.2.x)	released

## Table of Contents

<b>Overview of Service Bulletins for Integrated Drives (ID)</b>	<b>4</b>
<b>1 Specifications of the Interface</b>	<b>5</b>
<b>2 Firmware Configuration</b>	<b>5</b>
<b>3 Frame Structure</b>	<b>6</b>
3.1 Transmission (TX) protocol	6
3.2 Receiving (RX) protocol	6
<b>4 Handling of numbers and characters</b>	<b>7</b>
<b>5 Communication Protocol from Controller Unit to the Motorized Capacitor</b>	<b>8</b>
5.1 List of commands	8
5.2 Data bytes of the GetValue command 0x40	10
5.3 Data bytes of the SetSpeedConfig command 0x43	10
5.4 Data bytes of the StoreStepPosition command 0x75	11
5.5 Summary of Commands per Firmware	12
<b>6 Communication Protocol from Motorized Capacitor to the Control Unit</b>	<b>13</b>
6.1 Overview of response codes with examples	13
6.2 Detailed syntax of response codes	15
6.3 Data Bytes of the returnValue answer code 0x41	16
6.4 Status data byte returned by command GetValue(0x22)	17
6.4.1 General structure of response data byte	17
6.4.2 Description of the error byte	17
6.4.3 Examples of responses to the GetValue(0x22) command	18
6.5 Examples of error responses	18
<b>7 Examples of Commands</b>	<b>19</b>
7.1 Example for the command "Initialize":	19
7.2 Example for the command "Move-N-Steps: 1000":	19
7.3 Example for the command "Goto-Capacitance: 500.0 pF":	19
<b>8 Transfer of Capacitance Data</b>	<b>20</b>
<b>9 Revision history</b>	<b>21</b>

## Overview of Service Bulletins for Integrated Drives (ID)

- SB-60** Overview of Integrated Drives
- General product description
  - Drive unit product lines and related capacitor series
  - Function levels and configurations
  - Description of module components
  - Technical specifications
  - Overview of the product range
  - Type designation
- SB-61** Drive Unit
- Drive unit product lines
  - Description of the drive unit components
- SB-62** Introduction to Stepping Motors
- Principal function of stepper motors
  - Control system of the drive unit
- SB-63** Electrical Installation ID-400
- Stepping driver control signals and connections for EXPERT ID
- SB-64** Electrical Installation ID-1200
- Stepping driver control signals and connections for EXPERT ID
- SB-65** Electrical Installation ID-2800
- Stepping driver control signals and connections for EXPERT ID
- SB-66** Electrical Installation ID-5400
- Stepping driver control signals and connections for EXPERT ID
- SB-67** Step/Direction; Clockwise / Counter clockwise (CW/CCW) Interface
- Stepping driver signals
  - Stepping driver timing diagram
- SB-68** Software Protocol Interface RS-232
- Specification of the interface
  - Frame structure
  - Communication protocol between the ID and the host system
- SB-69** Software Protocol Interface RS-485
- Specification
  - Frame structure
  - Communication protocol between the ID and the host system
- SB-72** Safety Aspects of Integrated Drives
- Capacitor
  - Electrical Insulation between ID and Capacitor

**DATA SHEETS** are available for each Integrated Drive

## 1 Specifications of the Interface

RS-232 communication involves one master and one slave. The master sends a command to the slave and waits for the slave to reply. The slave can reply to this command at any time. After this reply, the master is ready to send the next command.

Mode of Operation	Single-ended
Maximum Driver Output Voltage	+/- 25 V
Driver Output Signal Level (Loaded Min.)	+/- 5 V to +/- 15 V
Driver Output Signal Level (Unloaded Max)	+/- 25 V
Driver Load Impedance	3 k $\Omega$ to 7 k $\Omega$
Max. Driver Current in High Z State Power On	N/A
Max. Driver Current in High Z State Power Off	+/- 6 mA @ +/- 2 V
Slew Rate (Max.)	30 V / $\mu$ s
Receiver Input Voltage Range	+/- 15 V
Receiver Input Sensitivity	+/- 3 V
Receiver Input Resistance	3k to 7k
Baud rate at the Integrated Drive	9600 Bd (default) (other values are possible upon customer request)
Maximum Cable Length	152 m (450 ft)
Data bits	8
Parity	-
Stop bit	1

Fig. 1: Specifications for the RS-232 interface

## 2 Firmware Configuration

The firmware source is modified as required before compilation. Several adjustable items are configured as needed for the specific product and customer:

Full details on the exact settings are given in the firmware configuration file which can be obtained from COMET upon request. The firmware part number for a specific motorized capacitor product is given on the product datasheet.

## 3 Frame Structure

### 3.1 Transmission (TX) protocol

Fig. 2 shows the principal structure of the RS-232 frame:

Start	Command	Data	Check Sum
0xAA	1 Byte	0 to 1024 Bytes	1 Byte

Fig. 2: Frame of the RS-232 transmission protocol

<b>Start</b>	One Byte: 0xAA
<b>Command</b>	One Byte
<b>Data</b>	The number of data bytes depends on the specific command; it can vary from 0 to 1024 Bytes.
<b>Check sum</b>	The Check sum is the 8-bit addition over all Bytes of the command, also included the start command (0xAA).

Fig. 3: Components of a frame of the RS-232 transmission protocol

The frame structure is expanded by two more levels in the case of uploading the C-curve data from the Motorized Capacitor module to the Control Unit. This case is treated in section 7.1.

### 3.2 Receiving (RX) protocol

Fig. 4 shows the principal structure of the RS-232 frame:

Start	Command	Data	Check Sum
0xAA	1 Byte	0 to 1024 Bytes	1 Byte

Fig. 4: Frame of the RS-232 receiving protocol

<b>Start</b>	One Byte: 0xAA
<b>Answer</b>	One Byte
<b>Data</b>	The number of data bytes depends on the specific command; it can vary from 0 to 1024 Bytes.
<b>Check sum</b>	The Check sum is the 8-bit addition over all Bytes of the command, also included the start command (0xAA).

Fig. 5: Components of a frame of the RS-232 receiving protocol

## 4 Handling of numbers and characters

All numbers are stored and transmitted according to the Little Endian<sup>1</sup> format. The representation of numbers is binary

Decimal value	Hexadecimal value
32767	0x7FFF
...	...
0	0x0000
-1	0xFFFF
...	...
-32768	0x8000

Fig. 6: Handling of numbers

It is important to note that all capacitance values are multiplied by factor of ten before they are encoded as described above. In this way it is possible to achieve a one decimal place resolution of the capacitance while still working with integer numbers. All other values are **not** multiplied by ten.

Until 2010, the serial number of the Motorized Capacitor module consists of one letter and a 5-digit integer number. The letter represents the year of production, e. g. "M" for 2004.

From 2011, the format of the serial number of the Motorized Capacitor module was changed to consist of a 6-digit integer number.

Characters are coded into one Byte using the ASCII format.

Character	Code
"A"	0x41
...	...
"Z"	0x5A
" - "	0x5F
" . "	0x2E

Fig. 7: Handling of letters

<sup>1</sup> "Little Endian" means that the low-order byte of the number is stored in memory at the lowest address, and the high-order byte at the highest address

## 5 Communication Protocol from Controller Unit to the Motorized Capacitor

The following tables show the commands, which are available to control the Integrated Drive with any controller unit.



Fig. 8: Communication from the controller unit to the Motorized Capacitor

### 5.1 List of commands

Command	Code	Data Bytes	Description
Initialize (Full reference)	0x10	N/A	This function initializes the Motorized Capacitor: <ul style="list-style-type: none"> <li>- Move variable electrode to reference position at <math>C_{min}</math> (mechanical end-stop)</li> <li>- Detect the hard end-stop</li> <li>- Move variable electrode to <math>C_{max}</math> (mechanical end-stop)</li> <li>- Detect the hard end-stop</li> <li>- Move variable electrode to reference position at <math>C_{min}</math> (mechanical end-stop)</li> <li>- Detect the hard end-stop</li> </ul>
Goto-Capacitance	0x20	2 Bytes (capacitance value) [0.1 pF]	Move variable electrode to the defined capacitance value [0.1 pF] (only possible if C-curve is stored on EPROM)
Goto-StepPosition	0x21	2 Bytes (step position) [full steps]	Move variable electrode to the defined step position [full steps]
Move-N-Steps	0x22	2 Bytes (signed, number of steps) [full steps]	Change the actual step position by the defined number of steps [full steps] (relative move)
Goto-MinPosition	0x23	N/A	Firmware V.1.2.x, V.2.1.1: Move variable electrode to $C_{min}$ position (mechanical end-stop) Firmware V.2.2.x: Move variable electrode to Lower Customer Limit
Goto-MaxPosition	0x24	N/A	Firmware V.1.2.x, V.2.1.1: Move variable electrode to $C_{max}$ position (mechanical end-stop) Firmware V.2.2.x: Move variable electrode to Upper Customer Limit
Goto-MicroStep Position	0x25	4 Bytes (micro step position) [micro steps]	Move variable electrode to the defined micro-step position [micro steps] 1 full step = 16 micro steps
Move-N-MicroSteps	0x25	4 Bytes (micro step position) [micro steps]	Change the actual micro-step position by the defined number of steps [micro steps] (relative move) 1 full step = 16 micro steps



Command	Code	Data Bytes	Description
Goto-Stored-Position (supported from firmware 2.1.1)	0x27	1 Byte (=index)	Move variable electrode to the stored step position with index n (index from 0x00 to 0x09)
Initialize (Reduced reference)	0x33	N/A	This function initializes the Motorized Capacitor: <ul style="list-style-type: none"> <li>- Move variable electrode to reference position at <math>C_{min}</math> (mechanical end-stop)</li> <li>- Detect the hard end-stop</li> </ul>
GetValue	0x40	1 Byte	Read a value from the Motorized Capacitor (see section 5.2)
SetSpeedConfig	0x43	2 Bytes	Configure speed of the Motorized Capacitor First byte acceleration vector value, second byte speed vector value (see section 5.3)
Set Lower Customer Limit	0x7201	2 Bytes (capacitance value) [0.1 pF]	Sets the lower customer limit Customer limit can only be set within the given factory limits
Set Upper Customer Limit	0x7202	2 Bytes (capacitance value) [0.1 pF]	Sets the upper customer limit Customer limit can only be set within the given factory limits
StoreStepPosition (supported from firmware 2.1.1)	0x75	3 Bytes	Store step count position with index in EEPROM (byte 1: index, bytes 2&3: step position)

Fig. 9: Commands from the Control Unit to the Motorized Capacitor for RS-232 interface

## 5.2 Data bytes of the GetValue command 0x40

Description	Data byte 1	Further data bytes
Actual capacitance	0x01	-
Actual full step position	0x02	-
Minimum capacitance	0x10	-
Maximum capacitance	0x11	-
Minimum step position	0x12	-
Maximum step position	0x13	-
Serial number (supported from firmware 2.x)	0x14	-
Firmware p/n and revision (supported from firmware 2.x)	0x15	-
Configuration	0x20	-
Configuration speed	0x21	-
Status (supported from firmware 2.x)	0x22	-
C-curve	0x30	-
Temperature	0x32	-
Overall sum of Full Steps	0x34	-
Overall sum of Initializations	0x35	-
Actual micro step position	0x36	-
Stored step count position (supported from firmware 2.x)	0x75	1 Byte (index from 0x00 to 0x09)
Lower Factory Limit (supported from firmware 2.2.0)	0x76	-
Upper Factory Limit (supported from firmware 2.2.0)	0x77	-
Lower Customer Limit (supported from firmware 2.2.0)	0x78	-
Upper Customer Limit (supported from firmware 2.2.0)	0x79	-

Fig. 10: Data Byte of the command 'GetValue'

## 5.3 Data bytes of the SetSpeedConfig command 0x43

Nibble	Description
UpperByteUpperNibble	No data
UpperByteLowerNibble	Acceleration; 0 (lowest) ... 15 (highest); we recommend an acceleration of 5
LowerByteUpperNibble	Start speed; 0 (lowest) ... 15 (highest) Start speed must be smaller than driving speed
LowerByteLowerNibble	Driving speed; 0 (lowest) ... 15 (highest); The maximum speed in steps per second is shown on the Data-Sheet. At each lower step, the speed goes reduces by 1/16 of the maximum speed.

Fig. 11: Data Bytes of the command 'SetSpeedConfig'

## 5.4 Data bytes of the StoreStepPosition command 0x75

This command is only supported by firmware 2.x and higher.

Byte Number	Description
Byte 1	Index from 0 to 9 (0x00 ... 0x09)
Bytes 2 and 3	Step count position

Fig. 12: Data Bytes of the command 'StoreStepPosition'.

## 5.5 Summary of Commands per Firmware

Command	Code	Firmware				
		V.1.2.x	V.2.1.1	V.2.2.x		
Initialize (Full reference)	0x10	x	x	x		
Goto-Capacitance	0x20	x	x	x		
Goto-StepPosition	0x21	x	x	x		
Move-N-Steps	0x22	x	x	x		
Goto-MinPosition	0x23	x	x	x		
Goto-MaxPosition	0x24	x	x	x		
Goto-MicroStep Position	0x25	x	x	x		
Move-N-MicroSteps	0x25	x	x	x		
Goto-Stored-Position	0x27		x	x		
Initialize (Reduced reference)	0x33	x	x	x		
Actual capacitance	0x4001	x	x	x		
Actual step position	0x4002	x	x	x		
Minimum capacitance	0x4010	x	x	x		
Maximum capacitance	0x4011	x	x	x		
Minimum step position	0x4012	x	x	x		
Maximum step position	0x4013	x	x	x		
Serial number	0x4014		x	x		
Firmware with revision	0x4015		x	x		
Configuration	0x4020	x	x	x		
Configuration speed	0x4021	x	x	x		
Status	0x4022		x	x		
C-curve	0x4030	x	x	x		
Temperature	0x4032	x	x	x		
Overall sum of Full Steps	0x4034	x	x	x		
Overall sum of Initializations	0x4035	x	x	x		
Stored step count position	0x4075		x	x		
Lower Factory Limi	0x4076			x		
Upper Factory Limit	0x4077			x		
Lower Customer Limit	0x4078			x		
Upper Customer Limit	0x4079			x		
SetSpeedConfig	0x43	x	x	x		
Set Lower Customer Limit	0x7201			x		
Set Upper Customer Limit	0x7202			x		
StoreStepPosition	0x75		x	x		

## 6 Communication Protocol from Motorized Capacitor to the Control Unit

The following tables show the responses (RX), that will be sent from the Integrated Drive to the controller unit after a command (TX) from the controller unit to the Integrated Drive has been received.

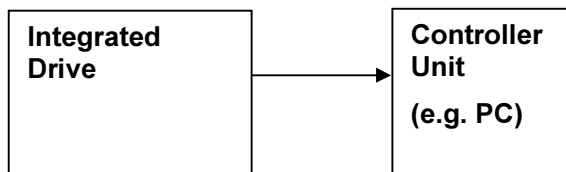


Fig. 13: Communication from the controller unit to the Motorized Capacitor

### 6.1 Overview of response codes with examples

Command (TX)	Response (RX)	Example
Initialize Full Reference (0x10)	<u>Firmware 1.x:</u> 0xF0 (Completion) <u>Firmware 2.x:</u> 0x50 (Acknowledge) 0xF0 (Completion)	TX: AA10 BA RX: AA50 FA (immediate answer to indicate movement) RX: AAF0 9A (answer after completion of referencing)
Goto-Capacitance (0x20)	0x50 (Acknowledge) 0x51 (Completion)	TX: AA20 1770 51 (go to 600 pF) RX: AA50 FA (immediate answer to indicate movement) RX: AA51 FB (answer after completion of movement)
Goto-StepPosition (0x21)  (Limit feature supported from firmware V.2.2.0)	0x50 (ACK: Limits ok) 0x93 (NAK: Limits nok) 0x51 (Completion)	TX: AA21 0258 25 (go to step count 600) RX: AA50 FA (immediate answer to indicate movement & target position is within customer limits) RX: AA93 3D (immediate answer to indicate movement but target position exceeds customer limits) RX: AA51 FB (answer after completion of movement)
Move-N-Steps (0x22)  (Limit feature supported from firmware V.2.2.0)	0x50 (ACK: Limits ok) 0x93 (NAK: Limits nok) 0x51 (Completion)	TX: AA22 0258 26 (move 600 steps) RX: AA50 FA (immediate answer to indicate movement & target position is within customer limits) RX: AA93 3D (immediate answer to indicate movement but target position exceeds customer limits) RX: AA51 FB (answer after completion of movement)
Goto-MinPosition (0x23)	0x50 (Acknowledge) 0x51 (Completion)	TX: AA23 CD RX: AA50 FA (immediate answer to indicate movement) RX: AA51 FB (answer after completion of movement)
Goto-MaxPosition 0x24	0x50 (Acknowledge) 0x51 (Completion)	TX: AA24 CE RX: AA50 FA (immediate answer to indicate movement) RX: AA51 FB (answer after completion of movement)

Command (TX)	Response (RX)	Example
Goto-MicroStepPosition (0x25)  (Limit feature supported from firmware V.2.2.0)	0x50 (ACK: Limits ok) 0x93 (NAK: Limits nok) 0x51 (Completion)	TX: AA25 0000 1F40 2E (go to micro-step count 8000) RX: AA50 FA (immediate answer to indicate movement & target position is within customer limits) RX: AA93 3D (immediate answer to indicate movement but target position exceeds customer limits) RX: AA51 FB (answer after completion of movement)
Move-N-MicroSteps (0x26)  (Limit feature supported from firmware V.2.2.0)	0x50 (ACK: Limits ok) 0x93 (NAK: Limits nok) 0x51 (Completion)	TX: AA26 0000 0C80 5C (move 3200 micro-steps) RX: AA50 FA (immediate answer to indicate movement & target position is within customer limits) RX: AA93 3D (immediate answer to indicate movement but target position exceeds customer limits) RX: AA51 FB (answer after completion of movement)
Goto-Stored-Position 0x27 (supported from firmware 2.1.1) (Limit feature supported from firmware V.2.2.0)	0x50 (ACK: Limits ok) 0x93 (NAK: Limits nok) 0x51 (Completion)	TX: AA27 04D5 (go to stored position with index 4) RX: AA50 FA (immediate answer to indicate movement & target position is within customer limits) RX: AA93 3D (immediate answer to indicate movement but target position exceeds customer limits) RX: AA51 FB (answer after completion of movement)
Initialize Reduced Reference (0x33)	<u>Firmware 1.x:</u> 0xF0 (Completion) <u>Firmware 2.x:</u> 0x50 (Acknowledge) 0xF0 (Completion)	TX: AA33 DD RX: AA50 FA (immediate answer to indicate movement) RX: AAF0 9A (answer after completion of referencing)
GetValue (0x40)	0x41 (Acknowledge)	TX: AA40 01 EB (request capacitance value) RX: AA41 01 07 0C FF (answer with capacitance value)
SetSpeedConfig (0x43)	<u>Firmware 1.x:</u> no answer <u>Firmware 2.x:</u> 0x8F (Acknowledge)	TX: AA43 0F 0F 0B (set speed=15, acceleration=15) RX: AA8F 39 (acknowledge)
StoreStepPosition (0x75) (supported from firmware 2.x)	0x8F (Acknowledge)	TX: AA75 03 02 58 7C (set step count pos 3 to 600) RX: AA8F 39 (acknowledge)

Fig. 14: Overview of commands and corresponding answer codes from the Motorized Capacitor to the Control Unit

## 6.2 Detailed syntax of response codes

Answer	Data Bytes	Description
returnValue	0x41 + n Bytes	Return the value of the query value from the ID
Serial Number (supported from FW 2.x)	0x41 14 + 8 Bytes	8 ASCII character (e. g. M13452__)
Firmware (supported from FW 2.x)	0x41 15 + 11 Bytes	11 ASCII characters (e. g. 20042324.03)
Status (supported from FW 2.x)	0x41 22 + 2 Bytes	Byte 1 = error bits, Byte 2 = status code
Stored position (supported from FW 2.x)	0x41 75 + 3 Bytes	Byte 1 = index of stored position (0x00 ... 0x09), Bytes 2 and 3 = step count position
other return values	0x41 + n Bytes	see section 5.2
SpeedConfig	0x43 + 2 Bytes	Return the speed and acceleration value
Movement started	0x50	The motor has started to move the variable electrode
Movement completed	0x51	The motor has stopped to moving the variable electrode
Initialization completed	0xF0	Initialization of Motorized Capacitor has been completed
Acknowledged (supported from FW 2.x)	0x8F	Command was accepted
Not acknowledged / Unknown command (supported from FW 2.x)	0x90	The command code is not defined
Not acknowledged / Frame error (supported from FW 2.x)	0x91	Frame format (start with 0xAA) is not correct or wrong number of data bytes
Not acknowledged / Checksum error (supported from FW 2.x)	0x92	Checksum is not correct
Not acknowledged / requested movement exceeds customer limit (supported from FW 2.2.x)	0x93	Target position of requested movement is beyond the lower / upper customer limit Motorized Capacitor will only run to the customer limit and stop there

Fig. 15: Answer codes from the Motorized Capacitor to the Control Unit

### 6.3 Data Bytes of the returnValue answer code 0x41

Description	Data Bytes	Description of additional Data Bytes
Actual capacitance	0x01 & 2 Bytes	[0.1 pF]
Actual step position	0x02 & 2 Bytes	[full steps]
Minimum capacitance	0x10 & 2 Bytes	[0.1 pF]
Maximum capacitance	0x11 & 2 Bytes	[0.1 pF]
Minimum step position	0x12 & 2 Bytes	[full steps]
Maximum step position	0x13 & 2 Bytes	[full steps]
Serial number (supported from FW 2.x)	0x14 & 8 Bytes	
Firmware with revision (supported from FW 2.x)	0x15 & 11 Bytes	
Configuration	0x20 & 2 Bytes	
Configuration speed	0x21 & 2 Bytes	Byte 1 = acceleration code; Byte 2 = speed code
Status (supported from FW 2.x)	0x22 & 1 Byte	
C-curve	0x30 & n Bytes	see chapter 8
Temperature	0x32 & 2 Bytes	[0.1 °C]
Overall sum of Full Steps	0x34 & 8 Bytes	[full steps]
Overall sum of Initializations	0x35 & 8 Bytes	
Stored step count position (supported from FW 2.x)	0x75 & 3 Bytes	Byte 1 = index of stored position (0x00 ... 0x09), Bytes 2 and 3 = step count position
Lower Factory Limit (supported from FW 2.2.0)	0x76 & 2 Bytes	[0.1 pF]
Upper Factory Limit (supported from FW 2.2.0)	0x77 & 2 Bytes	[0.1 pF]
Lower Customer Limit (supported from FW 2.2.0)	0x78 & 2 Bytes	[0.1 pF]
Upper Customer Limit (supported from FW 2.2.0)	0x79 & 2 Bytes	[0.1 pF]

Fig. 16: Data Bytes of the command 'returnValue'



## 6.4 Status data byte returned by command GetValue(0x22)

This command is only supported by firmware 2.x and higher.

### 6.4.1 General structure of response data byte

The response of the GetValue(0x22) consists of 1 byte:

Byte Number	Description
Byte 1	Error byte (8 bits which are individually set to 1 to indicate various error states. The bits are 0 in the absence of the respective error state)

Fig. 17: Response byte as returned by command GetValue(0x22).

### 6.4.2 Description of the error byte

The individual bits in the error byte have the following meaning:

Bit	Name	Function	Remark
0	OCA	overcurrent bridge A low side (ok: 0, error: 1)	Driver: 3PWM cycles with overcurrent within 64PWM cycles.
1	OCB	overcurrent bridge B low side (ok: 0, error: 1)	Driver: 3PWM cycles with overcurrent within 64PWM cycles.
2	OCHS	overcurrent high side (ok: 0, error: 1)	Driver: 3PWM cycles with overcurrent within 64PWM cycles.
3	UV	driver undervoltage (ok: 0, error: 1)	Driver undervoltage on VS. (Driver voltage < 5.9V typical [min.: 5.5V, max.: 6.2V])
4	OT	Overtemperature (ok: 0, overtemp: 1)	Temperature: tbd
5	RESET	Reset indicator (ok: 0, reset actuated: 1)	Reset indicator flag will be reset after reading.
6	-	Default (0)	Reserved
7	-	Default (0)	Reserved

Fig. 18: Description of the individual bits in the error byte which is returned by command GetValue(0x22).

Individual error states (error bit ON = set to 1) will result in the following error byte values

Error bit ON	Error byte value	Error type	Description
none	0x00	no error	All error bits 0 = no error
0	0x01	OCA error	overcurrent bridge A low side
1	0x02	OCB error	overcurrent bridge A low side
2	0x04	OCHS error	overcurrent high side
3	0x08	UV error	driver undervoltage
4	0x16	OT error	Overtemperature
5	0x20	RESET actuated	Reset indicator (bit 5 will be set to 0 after reading)
6	0x40	reserved	Per default, bit 6 is set to 0 always
7	0x80	reserved	Per default, bit 7 is set to 0 always

Fig. 19: Interpretation of error byte values in hexadecimal notation.

### 6.4.3 Examples of responses to the GetValue(0x22) command

The status byte contains a code of 1-byte length which indicates the status of the motorized capacitor after the previous command has been received and processed:

Command (TX string)	Response (RX string)	Description
AA 40 22 0C	AA 41 22 04 11	Command: get status value Response: Error byte 0x04 = OCHS error
AA 40 22 0C	AA 41 22 00 0C	Command: get status value Response: Error byte 0x00 = no error

### 6.5 Examples of error responses

This following table shows various examples of incorrect TX strings and the resulting error responses of the firmware:

Command	Command (TX string)	Response (RX string)
Goto-Capacitance (0x20) (example value: 600 pF)	AA20 1770 51 (correct command)	AA50FA (@ movement started) AA51FB (@ movement completed)
	AA20 1770 52 (wrong checksum)	AA92 3C (= Not acknowledged / wrong checksum)
	AA20 BB 85 (1 Data byte missing, checksum ok)	AA91 3B (= Not acknowledged / frame error)
	AA20 1770 0051 (1 Data byte too much, checksum ok)	AA92 3CAA 913B (AA92 3C: Not acknowledged / wrong checksum; AA 91 3B: Not acknowledged / frame error) <u>Note:</u> firmware expects checksum at 5 <sup>th</sup> byte of TX string. The extra "00" byte at pos. 5 causes a checksum error; byte 6 is interpreted as a new request which is incomplete and causes a frame error

## 7 Examples of Commands

This chapter shows some examples of ID commands.

### 7.1 Example for the command “Initialize”:

Part of the Frame	Command	Further information
Start	0xAA	Refer to Fig. 3
Command	0x10	Refer to Fig. 9
Data		Dec(1000) = Hex(3E8)
Check sum	0xBA	Check sum = Hex(AA + 10) = Hex (BA)

Fig. 20: Example for a Data Frame “goToReference”

The reference run will be initialized when you send the following command to the serial interface of the PC: **AA10BA**

### 7.2 Example for the command “Move-N-Steps: 1000”:

Part of the Frame	Command	Further information
Start	0xAA	Refer to Fig. 3
Command	0x22	Refer to Fig. 9
Data	High Byte: 0x03 Low Byte: 0xE8	Dec(1000) = Hex(3E8)
Check sum	0xB7	Check sum = Hex(AA + 22 + 03 + E8) = Hex (1B7)

Fig. 21: Example for a Data Frame “Move-N-Steps”

The ID will move +1000 steps when you send the following command to the serial interface of the PC: **AA2203E8B7**

### 7.3 Example for the command “Goto-Capacitance: 500.0 pF”:

Don't forget that all capacitance values have to be multiplied by ten before they are included into the command string.

Part of the Frame	Command	Further information
Start	0xAA	Refer to Fig. 3
Command	0x20	Refer to Fig. 9
Data	High Byte: 0x13 Low Byte: 0x88	Dec(5000) = Hex(1388)
Check sum	0x65	Check sum = Hex(AA + 20 + 13 + 88) = Hex (165)

Fig. 22: Example for a Data Frame “Goto-Capacitance”

The ID will go to 500.0 pF, when you send the following command to the serial interface of the PC: **AA20 1388 65**

## 8 Transfer of Capacitance Data

The structure of the data for the transmission of the C-curve data is shown in the figure below. This is the answer to command GetValue(0x30).

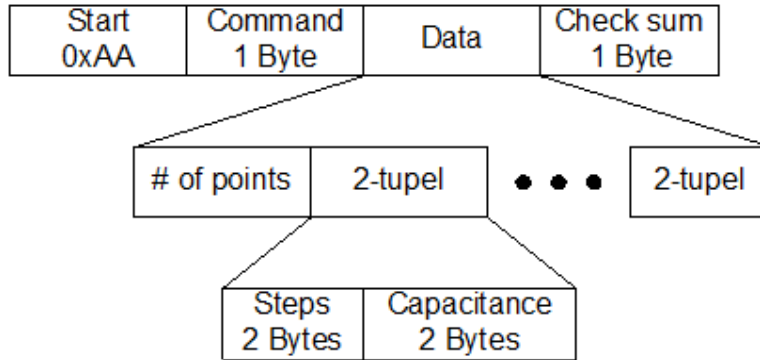


Fig. 23: Data structure for the transmission of the C-curve

Description	Data
# of points	Total number of points of the C-curve
C-curve	The complete C-curve is transmitted within one frame.

Fig. 24: Data Bytes when reading the C-curve

## 9 Revision history

Changes from document SB-68 V.1.1 to SB-68 Rev. 02:

Section	Description of change
General	Updated document to cover firmware version V.2.X in addition to V.1.X
2.1	Transmission (TX) protocol – wording improved
2.2	Receiving (RX) protocol – new section added
3	Handling of numbers and characters – wording improved; added hex codes for frequently used characters “_” and “.”
4.1	List of commands – new command “StoreStepPosition” added
4.2	Data bytes of the GetValue command 0x40 – marked byte value 0x14 (serial number) as supported from firmware 2.x; added new byte values 0x14 (firmware), 0x22 (status) and 0x60 (stored step count position) which are supported from firmware 2.x
4.4	Data bytes of the StoreStepPosition command 0x60 – new section added
5.1	Overview of response codes with examples – new section added
5.2	Detailed syntax of response codes – response code syntax for new commands and not acknowledged codes added which are supported from firmware 2.x
5.3	Data bytes of the returnValue answer code 0x41 – added information for data values which are supported from firmware 2.x
5.4	Status data byte returned by command GetValue(0x22) – new section added
8	Revision history – new section added

Changes from document SB-68 Rev. 02 to Rev. 03:

Section	Description of change
2	Firmware configuration – new section added
3.1	Transmission (TX) protocol – Error corrected: number of check sum bytes is 1, not 2
3.2	Receiving (RX) protocol – Error corrected: number of check sum bytes is 1, not 2
5.1	List of commands – changed code for command “StoreStepPosition” from 0x60 to 0x75 (code 0x60 was already used internally in the firmware); added units to commands Goto-Capacitance, Goto-StepPosition, and Move-N-Steps.
5.2	Data bytes of the GetValue command 0x40 – changed data byte value for stored step count position from 0x60 to 0x75 (code 0x60 was already used internally in the firmware); changed “firmware” to “firmware p/n” for clarification
5.4	Data bytes of the StoreStepPosition command 0x75 – changed command code from 0x60 to 0x75 (code 0x60 was already used internally in the firmware)
6.1	Overview of response codes with examples – corrected error in example TX string for command GetValue; changed StoreStepPosition command code to 0x75; changed acknowledge response for command StoreStepPosition and updated RX and TX example strings accordingly
6.2	Detailed syntax of response codes – changed answer code for “Stored position” from 0x60 to 0x75 (code 0x60 was already used internally in the firmware); added Acknowledged code 0x8F

Section	Description of change
6.3	Data Bytes of the returnValue answer code 0x41 – changed stored step count position code from 0x60 to 0x75 (code 0x60 was already used internally in the firmware); added units in column “Further information”; added further information for “Configuration speed”
6.5	Examples of error responses – new section added
8	Transfer of capacitance data – added information for clarification; error corrected: number of check sum bytes is 1, not 2

Changes from document SB-68 Rev. 03 to Rev. 04:

Section	Description of change
Title	Title format adapted to new layout
5.1	List of commands – added command Goto-Stored-Position (0x27)
6.1	Overview of response codes with examples – added command Goto-Stored-Position (0x27)

Changes from document SB-68 Rev. 04 to Rev. 05:

Section	Description of change
Title	Title format adapted to new layout
5.1	Updated description and added new commands
5.2	Added new commands
6.1	Added new commands and updated response codes
6.2	Updated response codes
6.3	Added new commands