Service Bulletin SB-68

V 1.1

Software protocol interface RS-232

Table of Contents

Overview of Service Bulletins for Integrated Drives (ID) **2**

1 Specifications of the Interface **3**

2 Frame Structure **4**

3 Handling of numbers and letters for the RS-232 ..... **4**

4 Communication Protocol Controller Unit to the Integrated Drive **5**

5 Communication Protocol Motorized Capacitor to the Control Unit **7**

6 Examples of Commands **8**

6.1 Example for the command “goToReference”: 8

6.2 Example for the command “doChangesSteps: 1000”: 8

6.3 Example for the command “goToCapacity: 500.0 pF”: 8

7 Transfer of Capacitance Data **9**

Issue date: 02-Jul-2007

Business Unit Vacuum Capacitors

Replaces: 30-Mar-2007

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-70 Software Protocol Interface I^2C
- Available soon

SB-71 Software Protocol Interface SPI
- Available soon

SB-72 Safety Aspects of Integrated Drives
- Capacitor
- Electrical Insulation between ID and Capacitor

SB-73 Test results for Integrated Drives
- Tests of components
- Life time tests

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 kΩ to 7 kΩ
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 / µs
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
Maximum Cable Length	152 m (450 ft)
Data bits	8
Parity	-
Stop bit	1

Fig. 1: Specifications for the RS-232 interface

2 Frame Structure

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

Start 0xAA	Command 1 Byte	Data 1 Byte to 1024 Bytes	Check Sum 2 Bytes

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

Start	One Byte: 0xAA
Command	One Byte
Data	The length depends on the vary command; it can vary from 1 Byte to 1024 Bytes.
Check sum	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 6.1.

3 Handling of numbers and letters for the RS-232

All numbers are stored and transmitted according to the Little Endian[1] format. The representation of numbers is binary

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

Fig. 4: 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.

The serial number of the Motorized Capacitor module consists of one letter and an integer number. The letter represents the year of production, e. g. “M” for 2004.

[1] "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

The letter is coded in one Byte using the ASCII format.

Letter	Code
“A”	0x41
…	…
“Z”	0x5A

Fig. 5: Handling of letters

4 Communication Protocol Controller Unit to the Integrated Drive

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

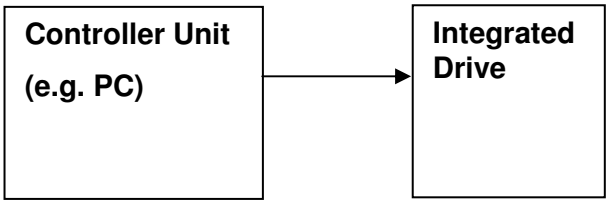

Fig. 6: Communication from the controller unit to the Motorized Capacitor

Command	Code	Data	Description
goToReference	0x10	N/A	Move variable electrode to reference position at C_{min}
goToCapacity	0x20	2 Bytes (capacitance value)	Move variable electrode to the defined capacitance value (only possible when C-curve is downloaded)
goToStepPosition	0x21	2 Bytes (step position)	Move variable electrode to the defined step position (absolute move - only possible when C-curve is downloaded)
doChangesSteps	0x22	2 Bytes (signed, number of steps	Change the actual step position by the defined number of steps (relative move)
goToMinPosition	0x23	N/A	Move variable electrode to C_{min} position
goToMaxPosition	0x24	N/A	Move variable electrode to C_{max} position
getValue	0x40	1 Byte	Read a value from the Motorized Capacitor (see Fig. 8)
setSpeedConfig	0x43	2 Bytes	Configure speed of the Motorized Capacitor

Fig. 7: Commands from the Control Unit to the Motorised Capacitor for RS-232 interface

Description	Data
Actual capacitance	0x01
Actual step position	0x02
Minimum capacitance	0x10
Maximum capacitance	0x11
Minimum step position	0x12
Maximum step position	0x13
Serial number	0x14
Configuration	0x20
Configuration speed	0x21
C-curve	0x30

Fig. 8: Data Bytes of the command 'getValue'

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 go reduces by 1/16 of the maximum speed.

Fig. 9: Data Bytes of the command 'setSpeedConfig'

5 Communication Protocol Motorized Capacitor to the Control Unit

The following tables show the commands, that will sent from the Integrated Drive to the controller unit after a request or command from any controller unit to the Integrated Drive.

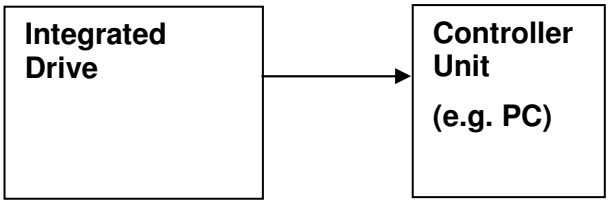

Fig. 10: Communication from the controller unit to the Motorized Capacitor

Command	Code	Data	Description
returnValue	0x41	2 Bytes	Return the value of the query value from the ID
MovementStarted	0x50	N/A	The motor has started to move the variable electrode
MovementFinished	0x51	N/A	The motor has stopped to moving the variable electrode
ReferenceStop	0xF0	N/A	The reference drive has been completed

Fig. 11: Commands from the Motorized Capacitor to the Control Unit

Description	Data
Actual capacitance	0x01 & 2 Bytes
Actual step position	0x02 & 2 Bytes
Minimum capacitance	0x10 & 2 Bytes
Maximum capacitance	0x11 & 2 Bytes
Minimum step position	0x12 & 2 Bytes
Maximum step position	0x13 & 2 Bytes
Serial number	0x14 & 4 Bytes
Configuration	0x20 & 2 Bytes
Configuration speed	0x21 & 2 Bytes
C-curve	0x30 & see chapter 6.1

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

6 Examples of Commands

This chapter shows some examples of ID commands.

6.1 Example for the command “goToReference”:

Part of the Frame	Command	
Start	0xAA	Refer to Fig. 3
Command	0x10	Refer to Fig. 7
Data		Dec(1000) = Hex(3E8)
Check sum	0xBA	Check sum = Hex(AA + 10) = Hex (BA)

Fig. 13: 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**

6.2 Example for the command “doChangesSteps: 1000”:

Part of the Frame	Command	
Start	0xAA	Refer to Fig. 3
Command	0x22	Refer to Fig. 7
Data	High Byte: 0x03 Low Byte: 0xE8	Dec(1000) = Hex(3E8)
Check sum	0xB7	Check sum = Hex(AA + 22 + 03 + E8) = Hex (1B7)

Fig. 14: Example for a Data Frame “doChangesSteps”

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

6.3 Example for the command “goToCapacity: 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	
Start	0xAA	Refer to Fig. 3
Command	0x20	Refer to Fig. 7
Data	High Byte: 0x13 Low Byte: 0x88	Dec(5000) = Hex(1388)
Check sum	0x65	Check sum = Hex(AA + 20 + 13 + 88) = Hex (165)

Fig. 15: Example for a Data Frame “goToCapacity”

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

7 Transfer of Capacitance Data

The structure of the data for the transmission of the C-curve data is shown

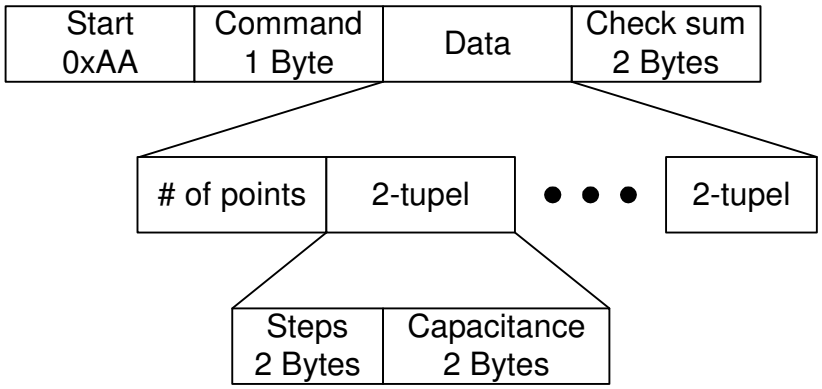

Fig. 16: Data structure for the transmission of the C-curve

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

Fig. 17: Data Bytes when reading the C-curve