Motorized Capacitor

Interface Protocol (RS232 / RS485)

Document Information

Authors ..... Th. Fenske
Document ..... SB-69_Universal-ID_Interface-Protocol-RS232-RS485.docx
Created on 01-June-2018
Revision 1.1.1
Comment ..... Updated version

Document history

Doc. Rev.	Firmware Rev.	Date	Author(s)	Change(s)	Status
1.0.0	none	2017-11-28	Th. Fenske	Initial document	released
1.1.0	none	2017-11-28	Th. Fenske	Added chapter “Get-Status (0x4022)”	released
1.1.1	none	2018-06-01	Th. Fenske	Added chapter 1.1 (Interface Parameters)	released

Table of Contents

1 Specification of RS232 / RS485 Interface

RS-232 communication involves one master and one slave.
RS-485 communication involves one master and up to 32 slaves.
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.

Specifications	RS232	RS485
Mode of Operation	SINGLE-ENDED	DIFFERENTIAL
Total Number of Drivers and Receivers on One Line (One driver active at a time for RS485 networks)	1 DRIVER 1 RECVR	32 DRIVER 32 RECVR
Maximum Cable Length	15 m	1220 m
Maximum Data Rate (15m – 1220m for RS422/RS485)	20kb/s	10Mb/s-100Kb/s
Maximum Driver Output Voltage	+/-25V	-7V to +12V
Driver Output Signal Level (Loaded Min.)	+/-5V to +/-15V	+/-1.5V
Driver Output Signal Level (Unloaded Max)	+/-25V	+/-6V
Driver Load Impedance	3 kΩ to 7 kΩ	54 Ω
Max. Driver Current in High Z State (Power on)	N/A	+/-100uA
Max. Driver Current in High Z State	+/-6mA @ +/-2v	+/-100uA
Slew Rate (Max.)	30V/uS	n/a
Receiver Input Voltage Range	+/-15V	-7V to +12V
Receiver Input Sensitivity	+/-3V	+/-200mV
Receiver Input Resistance, (1 Standard Load for RS485)	3 kΩ to 7 kΩ	>=12 kΩ

Table 1: Specifications for the RS232 / RS485 interface[1]

1.1 Interface Parameters

The following default interface parameters apply if not stated different (e.g. in datasheet)

Parameter	Value
Baud rate	115’200
Parity	none
Data bits	8
Stop bit (s)	1

[1] http://www.rs485.com/rs485spec.html

2 Frame Structure

2.1 General

Both RS232 and RS485 protocol use the same frame structure.

- Every TX command gets (strictly) one RX response (ACK resp. answer or NAK)
- First byte of every TX command must be 0xAA – otherwise command is ignored
- For RS232 and RS485 a time interrupt of at least 10 ms between each TX command has to be applied (to allow the slaves to distinguish between the commands)
- RS232 protocol also contains the address byte – a dummy address of 0x21 shall be applied which will be ignored by the RS232 receiver

2.2 Transmission (TX) protocol

Table 2 shows the principal structure of the RS232 / RS485 frame (TX):

Start	Address	Command	Data	Check Sum
0xAA	**1 Byte**	**2 Bytes** (Fix for all commands)	**0 to 8 Bytes** (Dependent on command)	**1 Byte**

Table 2: Frame of the RS232 / RS485 transmission protocol

Start	One Byte: 0xAA (fixed)
Address	One Byte Host address is 0x20 (fix). For RS232: Fix address of 0x21 For RS485: Address of slaves can be set between 0x21 and 0x40
Command	Two Bytes (for all commands)
Data	The number of data bytes depends on the specific command; it can vary from 0 to 8 Bytes
Check sum	Check sum is the 8-bit addition over all Bytes of the Start, Address, Command and Data bytes (no special identifier).

Table 3: Components of a frame of the RS232 / RS485 transmission protocol

2.3 Receiving (RX) protocol

Table 4 shows the principal structure of the RS232 / RS485 frame (RX):

Start	Address	Command	Data	Check Sum
0xAA	**1 Byte**	**1 or 2 Bytes** (Dependent on command)	**0 to 11 Bytes** (Dependent on command)	**1 Byte**

Table 4: Frame of the RS232 / RS485 receiving protocol

Start	One Byte: 0xAA (fixed)
Address	One Byte Host address is 0x20 (fix). For RS232: Fix address of 0x21 For RS485: Address of slaves can be set between 0x21 and 0x40
Command	One or two Bytes
Data	The number of data bytes depends on the specific command; it can vary from 0 to 11 Bytes
Check sum	Check sum is the 8-bit addition over all Bytes of the Start, Address, Command and Data bytes (no special identifier).

Table 5: Components of a frame of the RS232 / RS485 receiving protocol

3 Handling of numbers and characters

All numbers are stored and transmitted according to the Little Endian[2] format. The representation of numbers is binary.

3.1 Capacitance values (2 Bytes)

All capacitance values are multiplied by factor of ten before encoding them. 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.

All capacitance values (multiplied by 10) have two bytes (0x0000 to 0xFFFF) which is equivalent to unsigned integer values U16 (0 to 65'535)

3.2 Full-Step values (2 Bytes)

Absolute full-step positions (e.g. "Go-To-Full-Step-Position" 0x21) have a length of two bytes (0x0000 to 0xFFFF) which is equivalent to unsigned integer values U16 (0 to 65'535).
For a standard stepper motor with 200 full-steps per resolution this equals 327.68 turns.

Relative full-step positions (e.g. "Move-Full-Steps" 0x22) have a length of two bytes as well (0x8000 to 0x7FFF) which is equivalent to signed integer values I16 (−32'768 to 32'767).
For a standard stepper motor with 200 full-steps per resolution this equals +/-163.385 turns.

Full-Step position "0" is at Cmin-mech-endstop always, which is the reference point for step counting.

3.3 Micro-Step values (4 Bytes)

Absolute micro-step positions (e.g. "Go-To-Micro-Step-Position" 0x25) have a length of four bytes (0x0000 0000 to 0xFFFF FFFF) which is equivalent to unsigned integer values U32 (0 to 4'294'967'295). With 256 micro-steps per full-step, this equals 16'777'216 full-steps.

Relative full-step positions (e.g. "Move-Micro-Steps" 0x26) have a length of four bytes as well (0x8000 0000 to 0x7FFF FFFF) which is equivalent to signed integer values I32 (-2147483648 to 2'147'483'647)

Micro-Step position "0" is at Cmin-mech-endstop always, which is the reference point for step counting.

3.4 ASCII characters (1 Byte per character)

Characters are coded into one Byte using the ASCII format. Following table shows some examples:

Character	"A"	"Z"	"_"	"."	"1"	"9"		
Code	0x41	0x5A	0x5F	0x2E	0x31	0x39		

[2] "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

4 Specification of commands triggering any cap movement

4.1 Full reference run (0x10)

4.1.1 Triggering the “full reference run” (0x1000)

Following command triggers a full reference run (→ Preset → Cmin → Cmax → Cmin → Preset)

Preset value is a fix position (between Cmin-nom and Cmax-nom) which is a fix parameter in firmware.
Preset capacitance and preset full-step position can be readout - see commands 0x70 resp. 0x71).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x1000	none	1 Byte		0xAA21 1000 DB
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B

4.1.2 Check if “full reference run” is completed (0x1001)

Following command checks if full reference run (→ Preset → Cmin → Cmax → Cmin → Preset) is completed
Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x1001	none	1 Byte		0xAA21 1001 DC
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	“Command approved & Movement completed”	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	“Command approved & Movement not yet completed”	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	“Movement has not been triggered”	0xAA21 531E

4.2 Reduced-Reference-Run (0x33)

4.2.1 Triggering the "Reduced-Reference-Run" (0x3300)

Following command triggers a reduced reference run (→ Preset → Cmin → Preset)

Preset value is a fix position (between Cmin-nom and Cmax-nom) which is a fix parameter in firmware.
Preset capacitance and preset full-step position can be readout - see commands 0x70 resp. 0x71).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x3300	none	1 Byte		0xAA21 3300 FE
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B

4.2.2 Check if "Reduced-Reference-Run" is completed (0x3301)

Following command checks if full reference run (→ Preset → Cmin → Preset) is completed.
Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x3301	none	1 Byte		0xAA21 3301 FF
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	"Command approved & Movement completed"	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	"Command approved & Movement not yet completed"	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	"Movement has not been triggered"	0xAA21 531E

4.3 Go-To-Capacitance (0x20)

4.3.1 Triggering the movement “Go-To-Capacitance” (0x2000)

Following command triggers the motorized capacitor to run to the given capacitance

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2000	2 Bytes: C [0.1 pF]	1 Byte	e.g. goto150 pF	0x AA21 2000 05DC CC
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.3.2 Check if “Go-To-Capacitance” is completed (0x2001)

Following command checks if movement “Go-To-Capacitance” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2001	none	1 Byte		0x AA21 2001 EC
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.4 Go-To-Full-Step-Position (0x21)

4.4.1 Triggering the movement “Go-To-Full-Step-Position” (0x2100)

Following command triggers the motorized capacitor to run to the given full-step position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2100	2 Bytes: Full-steps	1 Byte	e.g. goto 600 full-steps	0x AA21 2100 0258 46
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.4.2 Check if “Go-To-Full-Step-Position” is completed (0x2101)

Following command checks if movement “Go-To-Full-Step-Position” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2101	none	1 Byte		0x AA21 2101 ED
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.5 Move-Full-Steps (0x22)

4.5.1 Triggering the movement “Move-Full-Steps” (0x2200)

Following command triggers the motorized capacitor to move the given number of full-steps.
Positive full-steps run towards Cmax, negative towards Cmin.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2200	2 Bytes: Full-steps	1 Byte	e.g. move -150 full-steps	0x AA21 2200 FF6A 56
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.5.2 Check if “Move-Full-Steps” is completed (0x2201)

Following command checks if movement “Move-Full-Steps” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2201	none	1 Byte		0x AA21 2201 EE
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.6 Go-To-Cmin (0x23)

4.6.1 Triggering the movement “Go-To-Cmin” (0x2300)

Following command triggers the motorized capacitor to run to the Lower-Customer-Limit.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2300	none	1 Byte		0x AA21 2300 EE
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.6.2 Check if “Go-To-Cmin” is completed (0x2301)

Following command checks if movement “Go-To-Cmin” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2301	none	1 Byte		0x AA21 2301 EF
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.7 Go-To-Cmax (0x24)

4.7.1 Triggering the movement “Go-To-Cmax” (0x2400)

Following command triggers the motorized capacitor to run to the Upper-Customer-Limit.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2400	none	1 Byte		0xAA21 2400 EF
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.7.2 Check if “Go-To-Cmax” is completed (0x2401)

Following command checks if movement “Go-To-Cmax” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2401	none	1 Byte		0x AA21 2401 F0
F0One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.8 Go-To-Micro-Step-Position (0x25)

4.8.1 Triggering the movement “Go-To-Micro-Step-Position” (0x2500)

Following command triggers the motorized capacitor to run to the given micro-step position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2500	4 Bytes: Micro-steps	1 Byte	e.g. goto 16000 micro-steps	0x AA21 2500 0000 3E80 AE
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.8.2 Check if “Go-To-Micro-Step-Position” is completed (0x2501)

Following command checks if movement “Go-To-Micro-Step-Position” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2501	none	1 Byte		0x AA21 2501 F1
One of the following RX (ACK) will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.9 Move-Micro-Steps (0x26)

4.9.1 Triggering the movement “Move-Micro-Steps” (0x2600)

Following command triggers the motorized capacitor to move the given number of micro-steps.
Positive micro-steps run towards Cmax, negative towards Cmin.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2600	4 Bytes: Micro-steps	1 Byte	e.g. move -1600 micro-steps	0x AA21 2600 FFFF F9C0 A8
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E

4.9.2 Check if “Move-Micro-Steps” is completed (0x2601)

Following command checks if movement “Move-Full-Steps” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2601	none	1 Byte		0x AA21 2601 F2
One of the following RX will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

4.10 Go-To-Stored-Position (0x27)

4.10.1 Triggering movement “Go-To-Stored-Position” (0x2700)

Following command triggers the motorized capacitor to move to the given stored position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2700	1 Bytes: Index of stored position [0 .. 9]	1 Byte	e.g. goto indexed position “4”	0xAA21 2700 04F6
← **RX (ACK)**	0xAA	1 Byte	0x50	none	1 Byte	Command approved Movement started	0xAA21 501B
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command approved, Movement started but target position outside of customer limits, Cap will run till customer limit	0xAA21 935E
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved, Index is out of range	0xAA21 945F

4.10.2 Check if “Go-To-Stored-Position” is completed (0x2701)

Following command checks if movement “Go-To-Stored-Position” is completed. Command can be send while capacitor is still moving.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x2701	none	1 Byte		
One of the following RX will be received							
← **RX (ACK)**	0xAA	1 Byte	0x51	none	1 Byte	Command approved & Movement completed	0xAA21 511C
← **RX (ACK)**	0xAA	1 Byte	0x52	none	1 Byte	Command approved & Movement not yet completed	0xAA21 521D
← **RX (ACK)**	0xAA	1 Byte	0x53	none	1 Byte	Movement has not been triggered	0xAA21 531E

5 Specification of commands writing or setting parameters

5.1 Set-Acceleration-Speed-Index (0x4300)

Following command selects the indexed value within the given 16-value vector [0...15] (e.g. index = 4 sets the fifth vector value as speed value).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4300	2 Bytes: See table below	1 Byte	e.g. acceleration = 5 speed = 15	0xAA21 4300 050F 22
← **RX (ACK)**	0xAA	1 Byte	0x4300	none	1 Byte	Command approved	

Nibble	Description
Upper Byte, Upper Nibble	No data
Upper Byte, Lower Nibble	Acceleration; 0 (lowest) …15 (highest); It's recommended to use default acceleration of 5
Lower Byte, Upper Nibble	Start speed; 0 (lowest) …15 (highest) Start speed must be smaller than driving speed
Lower Byte, Lower Nibble	Driving speed; 0 (lowest) …15 (highest); Maximum speed in steps per second is shown on the Data-Sheet. At each lower step, the speed reduces by 1/16 of the maximum speed.

5.2 Set-Current (0x4500)

Following command sets idle and drive current for stepper motor.
Idle current can be set within 0 (lowest) and an upper limit I-idle-max.
I-idle-max is a fix parameter in firmware between [1...255].
Drive current can be set within 0 (lowest) and an upper limit I-drive-max.
I-drive-max is a fix parameter in firmware between [1…255].

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4500	2 Bytes: Byte 1: Idle current [0..I-idle-max] Byte 2: Drive current [0..I-drive-max]	1 Byte	e.g. set Idle-current = 40 Drive-current = 110	0xAA21 4500 286E A6
← **RX (ACK)**	0xAA	1 Byte	0x4500	none	1 Byte	Command approved	0xAA21 4500 10
→ **TX**	0xAA	1 Byte	0x4500	2 Bytes: Byte 1: I-Idle > I-idle-max and / or Byte 2: I-drive > I-drive-max	1 Byte	At least one current is set above its specified upper limit	0xAA21 4500 286F A7
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved: Index out of range	0xAA21 945F

5.3 Set-Slave-Address (0x6000)

Following command sets the address of the connected slave(s).
Slave address can be set within 0x21 (lowest) to 0x41 (highest).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x6000	1 Byte: Address [0x21 … 0x41]	1 Byte	e.g. address = 24	0xAA21 6000 244F
← **RX (ACK)**	0xAA	1 Byte	0x6000	none	1 Byte	Command approved	0xAA21 6000 2B
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved: Address out of range	0xAA21 945F

5.4 Set-Baud-Rate (0x6100)

Following command sets the baud rate (RS232 / RS485) of the connected module.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x6100	2 Bytes: Byte 1: Baud rate index RS232 [0 … 7] Byte 2: Baud rate index RS485 [0 … 7]	1 Byte	e.g. baud rate for RS232 and RS485 are set to 9'600	0xAA21 6100 0000 2C
← **RX (ACK)**	0xAA	1 Byte	0x6100	none	1 Byte	Command approved	0xAA21 6100 2C
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved: Address out of range	0xAA21 945F

Baud rate can be set as follow:

Baud rate index	0	1	2	3	4	5	6	7
Baud rate	9'600	14'400	19'200	28'800	38'400	57'600	'76'800	115'200

5.5 Set-Lower-Customer-Limit (0x7201)

5.5.1 Illustration of factory and customer limits

Following command sets “Lower-Customer-Limit” of capacitor. Lower-Customer-Limit can only be set within Factory-Limits.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x7201	2 Bytes: C [0.1 pF]	1 Byte	e.g. set limit to 25 pF	0xAA21 7201 00FA 38
← **RX (ACK)**	0xAA	1 Byte	0x7201	none	1 Byte	Command approved (requested limit inside factory limits)	0xAA21 7201 3E
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command not approved (requested limit outside of factory limits) ->>br>Command will be ignored	0xAA21 935E

5.6 Set-Upper-Customer-Limit (0x7202)

Following command sets “Upper-Customer-Limit” of capacitor. Upper-Customer-Limit can only be set within Factory-Limits.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x7202	2 Bytes: C [0.1 pF]	1 Byte	e.g. set limit to 250 pF	0xAA21 7202 09C4 0C
← **RX (ACK)**	0xAA	1 Byte	0x7202	none	1 Byte	Command approved	0xAA21 7202 3F
← **RX (NAK)**	0xAA	1 Byte	0x93	none	1 Byte	Command not approved (requested limit outside of factory limits) -> Command will be ignored	0xAA21 935E

5.7 Store-Indexed-Position (0x7500)

Following command sets “Indexed-Position” of capacitor.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x7500	3 Bytes: Byte 1: Index [0 .. 9] Bytes 2&3: Full-Step-Position	1 Byte	e.g. store index position idx=3, pos=800	0xAA21 7500 0303 2066
← **RX (ACK)**	0xAA	1 Byte	0x7500	none	1 Byte	Command approved	0xAA21 7500 40
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved: Index out of range	0xAA21 945F

6 Specification of commands reading parameters

6.1 Get-Actual-Capacitance (0x4001)

Following command returns the actual capacitance [0.1 pF].

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4001	none	1 Byte		0xAA21 4001 0C
← **RX (ACK)**	0xAA	1 Byte	0x4101	2 Bytes: C [0.1 pF]	1 Byte	Command approved e.g. C = 80.0 pF	0xAA21 4101 0320 30

6.2 Get-Actual-Full-Step-Position (0x4002)

Following command returns the actual full-step-position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4002	none	1 Byte		0xAA21 4002 0D
← **RX (ACK)**	0xAA	1 Byte	0x4102	2 Bytes: Full-steps	1 Byte	Command approved e.g. Pos = 650 full-steps	0xAA21 4102 028A 9A

6.3 Get-Minimum-Capacitance (0x4010)

Following command returns the minimum capacitance [0.1 pF] (as per stored C-curve).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4010	none	1 Byte		0xAA21 4010 1B
← **RX (ACK)**	0xAA	1 Byte	0x4110	2 Bytes: C [0.1 pF]	1 Byte	Command approved e.g. Cmin = 18.0 pF	0xAA21 4110 00B4 D0

6.4 Get-Maximum-Capacitance (0x4011)

Following command returns the maximum capacitance [0.1 pF] (as per stored C-curve).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4011	none	1 Byte		0xAA21 4011 1C
← **RX (ACK)**	0xAA	1 Byte	0x4111	2 Bytes: C [0.1 pF]	1 Byte	Command approved e.g. Cmax = 257.6 pF	0xAA21 4111 0A10 37

6.5 Get-Minimum-Full-Step-Position (0x4012)

Following command returns the minimum full-step-position (as per last initialization).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4012	none	1 Byte		0xAA21 4012 1D
← **RX (ACK)**	0xAA	1 Byte	0x4112	2 Bytes Full-steps	1 Byte	Command approved e.g. Pos-min = 0	0xAA21 4112 0000 1E

6.6 Get-Maximum- Full-Step-Position (0x4013)

Following command returns the maximum full-step-position (as per last initialization).

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4013	none	1 Byte		0xAA21 4013 1E
← **RX (ACK)**	0xAA	1 Byte	0x4113	2 Bytes Full-steps	1 Byte	Command approved e.g. Pos-max = 2404	0xAA21 4113 0964 8C

6.7 Get-Acceleration-Speed (0x4021)

Following command returns the serial-number of motorized capacitor.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4021	none	1 Byte		0xAA21 4021 2C
← **RX (ACK)**	0xAA	1 Byte	0x4121	2 Bytes: Byte 1: Acceleration [0…15] Byte 2: Drive-Speed [0…15]	1 Byte	Command approved e.g. Acceleration = 5 Speed = 7	0xAA21 4121 0507 39

6.8 Get-Status (0x4022)

Command returns the status of motorized capacitor.

Data or Error byte (8 bits which are individually set to 1) to indicate various error states. Bits are 0 in the absence of the respective error state.

Protected Command	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4022	none	1 Byte		
← **RX (ACK)**	0xAA	1 Byte	0x4122	2 Byte: Byte 1: Error bits Byte 2: n/a	1 Byte	Command approved	Command approved

6.8.1 Description of the Error Byte

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

Byte1 Bit	Error Bytes	Name	Function	Remark
	0x0000		No error	All error bits 0 = no error
0	0x0100	OCA	overcurrent bridge A low side (ok: 0, error: 1)	Driver: 3PWM cycles with overcurrent within 64PWM cycles
1	0x0200	OCB	overcurrent bridge B low side (ok: 0, error: 1)	Driver: 3PWM cycles with overcurrent within 64PWM cycles
2	0x0400	OCHS	overcurrent high side (ok: 0, error: 1)	Driver: 3PWM cycles with overcurrent within 64PWM cycles
3	0x0800	UV	driver under voltage (ok: 0, error: 1)	Driver under voltage on VS. (Driver voltage < 5.9V typical [min.: 5.5V, max.: 6.2V])
4	0x1600	OT	Over temperature (ok: 0, overtemp: 1)	Temperature: tbd
5	0x2000	RESET	Reset indicator (ok: 0, reset actuated: 1)	Reset indicator flag will be reset after reading
6		-	Per default, bit 6 is set to 0 always	Reserved
7		-	Per default, bit 7 is set to 0 always	Reserved

6.9 Get-C-curve-Indexed-Full-Step-Capacitance (0x4030)

Command returns the indexed tuple (full-step-position and capacitance) of C-curve table from EPROM.
C-curve can have a maximum of 500 entries.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4030	2 Bytes: Index in Table [0…x]	1 Byte	e.g. get entry 1	0xAA21 4030 0003 3E
← **RX (ACK)**	0xAA	1 Byte	0x4130	6 Byte: Byte 1&2: Index in Table Byte 2&3: Full-Steps Byte 4&5: C [0.1 pF]	1 Byte	Command approved e.g. Full-Step = 150, C = 32.5 pF	0xAA21 4130 0003 0096 0145 1B
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved: Index out of range	0xAA21 945F

6.10 Get-Controller-Temperature (0x4032)

Command returns the temperature of the controller-internal temperature sensor.
The one data byte (hex) is converted to an U8 (unsigned 8-bit integer) – temperature gets calculated from this value (*U16*) as per:

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4032	none	1 Byte		0xAA21 4032 3D
← **RX (ACK)**	0xAA	1 Byte	0x4132	1 Byte: Temperature [°C]	1 Byte	Command approved e.g. T = 33 °C	0xAA21 4132 215F

6.11 Get-Micro-Step-Position (0x4036)

Command returns the current micro-step-position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4036	none	1 Byte		0xAA21 4036 41
← **RX (ACK)**	0xAA	1 Byte	0x4136	4 Bytes: MicroStep-Position	1 Byte	Command approved e.g. micro-steps = 8000	0xAA21 4136 0000 1F40 A1

6.12 Get-Firmware Version (0x4061)

Following command returns the serial-number of motorized capacitor.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4061	none	1 Byte		0xAA21 4061 6C
← **RX (ACK)**	0xAA	1 Byte	0x4161	11 Bytes [ASCII character]	1 Byte	Command approved e.g. FW = 20100433.00	0xAA21 4161 3230 3130 3034 3333 2E30 30 88

6.13 Get-Serial-Number of Capacitor (0x4062)

Following command returns the serial-number of motorized capacitor.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4062	none	1 Byte		0xAA21 4062 6D
← **RX (ACK)**	0xAA	1 Byte	0x4162	8 Bytes [ASCII character]	1 Byte	Command approved e.g. s/n = ##654321	0xAA21 4162 2323 3635 3433 3231 E9

6.14 Get-Cmin-nom (0x4070)

Command returns the Cmin-nom value.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4070	none	1 Byte		0xAA21 4070 7B
← **RX (ACK)**	0xAA	1 Byte	0x4170	2 Bytes: Cmin-nom [0.1 pF]	1 Byte	Command approved e.g. Cmin-nom = 25 pF	0xAA21 4170 00FA76

6.15 Get-Cmax-nom (0x4071)

Command returns the Cmax-nom value.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4071	none	1 Byte		0xAA21 4071 7C
← **RX (ACK)**	0xAA	1 Byte	0x4171	2 Bytes: Cmin-nom [0.1 pF]	1 Byte	Command approved e.g. Cmax-nom = 250 pF	AA21 4171 09C4 4A

6.16 Get-Lower-Customer-Limit (0x4072 01)

Command returns the capacitance value at “Lower Customer Limit”.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4072	0x01	1 Byte		0xAA21 4072 017E
← **RX (ACK)**	0xAA	1 Byte	0x4172	3 Bytes: Byte 1: 0x01 Byte 2&3: C [0.1 pF]	1 Byte	Command approved e.g. LowCLim = 25 pF	0xAA21 4172 0100 FA79
← **RX (NAK)**	0xAA	1 Byte	0x94		1 Byte	Command not approved: Index out of range	0xAA21 945F

6.17 Get-Upper- Customer -Limit (0x4072 02)

Command returns the capacitance value at “Upper Customer Limit”.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4072	0x02	1 Byte		0xAA21 4072 027F
← **RX (ACK)**	0xAA	1 Byte	0x4172	3 Bytes: Byte 1: 0x02 Byte 2&3: C [0.1 pF]	1 Byte	Command approved e.g. UppCLim = 250 pF	0xAA21 4172 0209 C44D
← **RX (NAK)**	0xAA	1 Byte	0x94		1 Byte	Command not approved: Index out of range	0xAA21 945F

6.18 Get-Lower-Factory-Limit (0x4072 03)

Command returns the capacitance value at “Lower Factory Limit”.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4072	0x03	1 Byte		0xAA21 4072 0380
← **RX (ACK)**	0xAA	1 Byte	0x4172	3 Bytes: Byte 1: 0x03 Byte 2&3: C [0.1 pF]	1 Byte	Command approved e.g. LowFLim = 25 pF	0xAA21 4172 0300 FA7B
← **RX (NAK)**	0xAA	1 Byte	0x94		1 Byte	Command not approved: Index out of range	0xAA21 945F

6.19 Get-Upper-Factory-Limit (0x4072 04)

Command returns the capacitance value at “Upper Factory Limit”.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4072	0x04	1 Byte		0xAA21 4072 0481
← **RX (ACK)**	0xAA	1 Byte	0x4172	3 Bytes: Byte 1: 0x04 Byte 2&3: C [0.1 pF]	1 Byte	Command approved e.g. UppFLim = 250 pF	0xAA21 4172 0409 C44F
← **RX (NAK)**	0xAA	1 Byte	0x94		1 Byte	Command not approved: Index out of range	0xAA21 945F

6.20 Get-Preset-Capacitance (0x4073)

Command returns the equivalent capacitance at preset position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4073	0x00	1 Byte		0xAA21 4073 007E
← **RX (ACK)**	0xAA	1 Byte	0x4173	3 Bytes: Byte1: 0x00 Bytes 2&3: Preset-C [0.1 pF]	1 Byte	Command approved e.g. Cpreset = 81.1 pF	0xAA21 4173 0003 2BAD
← **RX (NAK)**	0xAA	1 Byte	0x94		1 Byte	Command not approved: Index out of range	0xAA21 945F

6.21 Get-Preset-Full-Step-Position (0x4073)

Command returns the equivalent full-step position at preset position.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4073	0x01	1 Byte		0xAA21 4073 017F
← **RX (ACK)**	0xAA	1 Byte	0x4173	3 Bytes: Byte 1: 0x01 Bytes 2&3: Full-steps	1 Byte	Command approved e.g. Pos-preset = 638	0xAA21 4173 0102 7E00
← **RX (NAK)**	0xAA	1 Byte	0x94		1 Byte	Command not approved: Index out of range	0xAA21 945F

6.22 Get-Stored-Full-Step-Position (0x4075)

Command returns the stored indexed full-step position for requested index.

	Start	Address	Cmd	Data	Check Sum	Comment	Example
→ **TX**	0xAA	1 Byte	0x4075	1 Byte: Index [0 .. 9]	1 Byte	e.g. for index = 2	0xAA21 4075 0282
← **RX (ACK)**	0xAA	1 Byte	0x4175	3 Bytes: Byte1: Index [0 .. 9] Bytes 2&3: Full-steps	1 Byte	Command approved e.g. Stored Pos = 532	0xAA21 4175 0202 1499
← **RX (NAK)**	0xAA	1 Byte	0x94	none	1 Byte	Command not approved: Index out of range	0xAA21 945F

7 Detailed syntax of error (NAK) response codes

7.1 Error codes (NAK) for all commands

For all commands listed in chapters 4.1 to 6.22, the following possible error responses shall apply (beside the ACK response in each corresponding chapter).

Answer	RX-Cmd Bytes	Description
Not acknowledged / Unknown command	0x90	The command code (byte 3&4) is not defined
Not acknowledged / Checksum error	0x92	The checksum is not correct

Table 6: Answer codes from the Motorized Capacitor to the Control Unit

7.2 Error codes (NAK) for commands triggering a movement

Error code 0x93 only applies for commands triggering a movement. If requested target position is outside of given customer limit, error code 0x93 will be returned.

Answer	RX-Cmd Bytes	Description
Not acknowledged / Limit reached	0x93	Target position outside of customer limits

8 Abbreviations

Abbreviation	Description of change
C	Capacitance of motorized capacitor
Cmd	Command or command bytes
RX	Receiving
TX	Transmitting