



# Serial Communication Protocol v3.33

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Glossary . . . . .	5
<b>2</b>	<b>General Packet Structures</b>	<b>6</b>
2.1	Command / Set Register . . . . .	6
2.2	Get Register . . . . .	6
2.3	Cyclic Reply / Acyclic Reply / Status Reply . . . . .	6
<b>3</b>	<b>Command/Register Overview</b>	<b>7</b>
<b>4</b>	<b>Status Reply</b>	<b>10</b>
4.1	Status Reply . . . . .	10
<b>5</b>	<b>Commands</b>	<b>11</b>
5.1	Reboot System . . . . .	11
5.2	Load Default Values . . . . .	11
5.3	Get Register Value . . . . .	11
5.4	Degauss . . . . .	11
5.5	Calibrate Encoder . . . . .	12
5.6	Set RGB LEDs . . . . .	12
5.7	Disable All Haptic Functions . . . . .	13
5.8	Start/Stop Power Measurement . . . . .	13
5.9	Calibrate Push Pull . . . . .	13
5.10	Hapticore Power Supply . . . . .	14
<b>6</b>	<b>Register (readonly)</b>	<b>15</b>
6.1	Controller ID . . . . .	15
6.2	Firmware Version . . . . .	15
6.3	Communication Protocol Version . . . . .	15
6.4	Hapticore Library Version . . . . .	16
6.5	Controller Hardware Revision . . . . .	16
6.6	Hapticore Serial Number . . . . .	16
6.7	Hapticore Product Name . . . . .	16
6.8	Hapticore Item Number . . . . .	17
6.9	Operating Time 1 . . . . .	17
6.10	Operating Time 2 . . . . .	17
6.11	Encoder Temperature . . . . .	18
6.12	Power Measurement Peak . . . . .	18
6.13	Power Measurement RMS . . . . .	18
<b>7</b>	<b>Register (read-/writeable)</b>	<b>19</b>
7.1	Hapticore System Configuration . . . . .	19
7.2	Report Type . . . . .	19
7.3	Report Flags . . . . .	19
7.4	Report Frequency . . . . .	20
7.5	Clutch Activation After Idle . . . . .	20
7.6	Preset RGB LEDs Red . . . . .	21
7.7	Preset RGB LEDs Green . . . . .	21
7.8	Preset RGB LEDs Blue . . . . .	21
7.9	Clutch Deactivation Blanking Duration . . . . .	21
7.10	Clutch Base Current . . . . .	22
7.11	Clutch Activation Current . . . . .	22
7.12	Clutch Activation Duration . . . . .	23
7.13	Idle Detection Velocity Threshold . . . . .	23
7.14	Idle Detection Timeout . . . . .	23
7.15	Idle Detection Current Ramp Down Slope . . . . .	24
7.16	Current Controller Kp . . . . .	24
7.17	Current Controller Idle Current . . . . .	24
7.18	Current Controller Mode . . . . .	25
7.19	Current Controller Ki . . . . .	25
7.20	Current Controller Coil Resistance . . . . .	25

7.21	Current Controller Coil Inductance . . . . .	26
7.22	Encoder Angle Filter R . . . . .	26
7.23	Encoder Velocity Filter R . . . . .	27
7.24	Current Controller Supply Voltage Override . . . . .	27
7.25	Current Controller Supply Voltage . . . . .	27
7.26	Coil Driver PWM Polarity . . . . .	28
7.27	Coil Driver Stop Mode . . . . .	28
7.28	Current Sense Polarity . . . . .	28
7.29	Encoder Mode . . . . .	29
7.30	Encoder Angle . . . . .	29
7.31	Encoder Time Triggered Angle Increment . . . . .	30
7.32	Encoder Velocity Filter Q . . . . .	30
7.33	Encoder Velocity Noise Threshold . . . . .	30
7.34	Encoder Direction . . . . .	31
7.35	Encoder Angle Filter Q . . . . .	31
7.36	Encoder Angle Resolution . . . . .	31
7.37	Encoder Offset . . . . .	32
7.38	Encoder Scaling . . . . .	32
7.39	Degauss Mode . . . . .	32
7.40	Degauss Frequency . . . . .	33
7.41	Degauss Current . . . . .	33
7.42	Degauss Duration . . . . .	33
7.43	Tick Index Count Mode . . . . .	34
7.44	Tick Degauss Current . . . . .	34
7.45	Tick Degauss Duration . . . . .	34
7.46	Tick Exponential Ramp Down Velocity Threshold . . . . .	35
7.47	Tick Exponential Ramp Down Factor . . . . .	35
7.48	Tick Exponential Ramp Down Current Min Percent . . . . .	35
7.49	Tick Mode . . . . .	36
7.50	Tick Enable . . . . .	36
7.51	Tick Current . . . . .	36
7.52	Tick Angle CW . . . . .	37
7.53	Tick Duration Min . . . . .	37
7.54	Tick Duration Max . . . . .	37
7.55	Tick Velocity Factor . . . . .	38
7.56	Tick Index Window . . . . .	38
7.57	Tick Stickiness Prevention Factor . . . . .	38
7.58	Tick Angle CCW . . . . .	39
7.59	Tick Active Direction . . . . .	39
7.60	Tick Freewheeling Velocity Threshold . . . . .	39
7.61	Tick Freewheeling Extension Time . . . . .	40
7.62	Tick Window . . . . .	40
7.63	Tick Start Angle . . . . .	40
7.64	Tick Stop Angle . . . . .	41
7.65	Barrier Enable . . . . .	41
7.66	Barrier Current . . . . .	41
7.67	Barrier Start Angle . . . . .	42
7.68	Barrier Stop Angle . . . . .	42
7.69	Barrier Polarity . . . . .	42
7.70	Barrier Limit Angle . . . . .	43
7.71	Current Enable . . . . .	43
7.72	Current Start Angle . . . . .	43
7.73	Current Stop Angle . . . . .	44
7.74	Current Start Current CW . . . . .	44
7.75	Current Stop Current CW . . . . .	44
7.76	Current Start Current CCW . . . . .	45
7.77	Current Stop Current CCW . . . . .	45
7.78	Current Freewheeling Extension Time . . . . .	45
7.79	Current Active Direction . . . . .	46
7.80	Torque Enable . . . . .	46
7.81	Torque Start Angle . . . . .	46
7.82	Torque Stop Angle . . . . .	47

7.83	Torque Start Factor CW	47
7.84	Torque Stop Factor CW	47
7.85	Torque Start Factor CCW	48
7.86	Torque Stop Factor CCW	48
7.87	Torque Freewheeling Velocity Threshold	48
7.88	Torque Freewheeling Extension Time	49
7.89	Torque Active Direction	49
7.90	Lock Enable	49
7.91	Lock Direction	50
7.92	Lock Current	50
7.93	Freewheeling Enable	51
7.94	Freewheeling Start Velocity Threshold	51
7.95	Freewheeling Friction	51
7.96	Freewheeling Damping	52
7.97	Freewheeling Inertia	52
7.98	Single Tick Enable	52
7.99	Single Tick Active Direction	52
7.100	Single Tick Angle	53
7.101	Single Tick Current	53
7.102	Single Tick Mode	53
7.103	Single Tick Duration Min	54
7.104	Single Tick Duration Max	54
7.105	Single Tick Window	54
7.106	Single Tick Velocity Factor	55
7.107	Haptics Generator Enable	55
7.108	Loopback	55
<b>8</b>	<b>Report Register (readonly)</b>	<b>57</b>
8.1	Encoder Angle	57
8.2	Encoder Velocity	57
8.3	Push Pull Value	57
8.4	Tick Index	57
8.5	Push Pull State	58
8.6	Encoder Multi Turn Count	58
8.7	Encoder Temperature	58
8.8	Coil Current	59
8.9	Total Turn Counter1	59
8.10	Total Turn Counter2	59
8.11	Device Error Status	60
8.12	Encoder Calibration Status	60
8.13	Coil Resistance	60
8.14	Push Calibration Status	61
8.15	Pull Calibration Status	61
8.16	Device Connection State	61
8.17	Encoder Angle Min	62
8.18	Encoder Angle Max	62
8.19	Supply Voltage	62
8.20	Degauss Status	63
<b>9</b>	<b>Haptics Generator General Packet Structure</b>	<b>64</b>
<b>10</b>	<b>HapticsGenerator Commands</b>	<b>65</b>
10.1	Write LUT Angle Current CW	65
10.2	Write LUT Angle Current CCW	65
10.3	Write LUT Angle Current Size CW	65
10.4	Write LUT Angle Current Size CCW	66
10.5	Write Alternate LUT Current	66
10.6	Write LUT Velocity Compensation Type	67
10.7	Write Tick Index Count Angle CW	67
10.8	Write Tick Index Count Angle CCW	68
10.9	Write LUT Velocity Factor	68
10.10	Write LUT Velocity Factor Size	68

10.11	Write LUT Freewheeling Velocity Threshold . . . . .	69
10.12	Write LUT Force Freewheeling . . . . .	69
10.13	Read LUT Angle Current CW . . . . .	69
10.14	Read LUT Angle Current CCW . . . . .	70
10.15	Read LUT Angle Current Size CW . . . . .	70
10.16	Read LUT Angle Current Size CCW . . . . .	70
10.17	Read Alternate LUT Current . . . . .	71
10.18	Read LUT Velocity Compensation Type . . . . .	71
10.19	Read Tick Index Count Angle CW . . . . .	71
10.20	Read Tick Index Count Angle CCW . . . . .	72
10.21	Read LUT Velocity Factor . . . . .	72
10.22	Read LUT Velocity Factor Size . . . . .	72
10.23	Read Freewheeling Velocity Threshold . . . . .	72
10.24	Read Force Freewheeling . . . . .	73

# 1 Introduction

## 1.1 Glossary

HAPTICORE device	Device with a haptic feedback function based on magnetorheological materials (MRM).
HAPTICORE controller device	Device which controls a HAPTICORE device.
Host device	A PC, embedded system, PLC, mobile device, etc.
Command	Packet which is sent from a host device to a HAPTICORE controller device.
Reply	Packet which is sent from a HAPTICORE controller device to a host device in reply to a command.
Cyclic Reply	Packet which is sent cyclically from a HAPTICORE device to a host device.
Acyclic Reply	Packet which is sent to a host device only if the value has changed, but in sync with the cyclic interval.

## 2 General Packet Structures

COM Port Settings: 115.200 Baud, 8N1

Packet length = 6 Byte

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
START	TYPE	DATA_HIGH	DATA_LOW	LRC	STOP

### Packet Byte Description

Name	Description	Value
START	Start Byte	'&' (0x26)
TYPE	Type of command or reply	See below
DATA_HIGH	High Byte of 16bit Value	See below
DATA_LOW	Low Byte of 16bit Value	See below
LRC	Longitudinal Redundancy Check	Byte 1 ^ Byte 2 ^ Byte 3
STOP	Stop Byte	'\r' (0x0D)

### Note

Command and reply packets have the same length and structure!

### 2.1 Command / Set Register

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
START	TYPE	DATA_HIGH	DATA_LOW	LRC	STOP

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
START	TYPE	DATA_HIGH	DATA_LOW	LRC	STOP

### 2.2 Get Register

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
START	CMD_GET_REGISTER_VALUE	[INDEX]	TYPE	LRC	STOP

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
START	TYPE	DATA_HIGH	DATA_LOW	LRC	STOP

### 2.3 Cyclic Reply / Acyclic Reply / Status Reply

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
START	TYPE	DATA_HIGH	DATA_LOW	LRC	STOP

### 3 Command/Register Overview

Type	Name	Category	Conversion Factor	Signedness
0x00	Status Reply	Reply	-	-
0x01	Reboot System	Command	-	-
0x02	Load Default Values	Command	-	-
0x03	Get Register Value	Command	-	-
0x04	Degauss	Command	-	-
0x05	Calibrate Encoder	Command	-	-
0x06	Set RGB LEDs	Command	-	-
0x08	Disable All Haptic Functions	Command	-	-
0x09	Start/Stop Power Measurement	Command	-	-
0x0E	Calibrate Push Pull	Command	-	-
0x0F	Hapticore Power Supply	Command / Register (r)	-	-
0x10	Controller ID	Register (r)	-	-
0x12	Firmware Version	Register (r)	-	-
0x13	Communication Protocol Version	Register (r)	-	-
0x14	Hapticore Library Version	Register (r)	-	-
0x15	Hapticore Controller Hardware Revision	Register (r)	-	-
0x16	Hapticore Serial Number	Register (r)	-	-
0x17	Hapticore Type	Register (r)	-	-
0x1B	Hapticore Product Family	Register (r)	1	unsigned
0x1C	Hapticore Product Model	Register (r)	1	unsigned
0x1D	Hapticore Product Variant	Register (r)	1	unsigned
0x1E	Hapticore Item Number	Register (r)	-	unsigned
0x20	Operating Time1	Register (r)	-	-
0x21	Operating Time2	Register (r)	-	-
0x22	Encoder Temperature	Register (r)	100	signed
0x23	Power Measurement Peak	Register (r)	100	unsigned
0x24	Power Measurement RMS	Register (r)	100	unsigned
0x29	Hapticore System Configuration	Register (rw)	1	unsigned
0x30	Report Type	Register (rw)	-	-
0x31	Report Flags	Register (rw)	-	-
0x32	Report Frequency	Register (rw)	1	unsigned
0x34	Clutch Activation After Idle Mode	Register (rw)	1	unsigned
0x36	Preset RGB LED Red	Register (rw)	-	-
0x37	Preset RGB LED Green	Register (rw)	-	-
0x38	Preset RGB LED Blue	Register (rw)	-	-
0x39	Clutch Deactivation Blanking Duration	Register (rw)	1	unsigned
0x3A	Clutch Base Current	Register (rw)	-	-
0x3B	Clutch Activation Current	Register (rw)	-	-
0x3C	Clutch Activation Duration	Register (rw)	-	-
0x3D	Idle Detection Velocity Threshold	Register (rw)	10	unsigned
0x3E	Idle Detection Timeout	Register (rw)	1000	unsigned
0x3F	Idle Detection Current Ramp Down Slope	Register (rw)	10	signed
0x40	Current Controller Update Frequency	Register (rw)	1	unsigned
0x41	Current Controller PWM Frequency	Register (rw)	1	unsigned
0x42	Current Controller Kp	Register (rw)	100	unsigned
0x43	Current Controller Idle Current	Register (rw)	1000	signed
0x44	Current Controller Mode	Register (rw)	-	-
0x45	Current Controller Ki	Register (rw)	1	unsigned
0x46	Current Controller Coil Resistance	Register (rw)	100	unsigned
0x47	Current Controller Coil Inductance	Register (rw)	10000	unsigned
0x48	Encoder Angle Filter R	Register (rw)	10	unsigned
0x49	Encoder Velocity Filter R	Register (rw)	10	unsigned
0x4A	Current Controller Supply Voltage Override	Register (rw)	-	-
0x4B	Current Controller Supply Voltage	Register (rw)	1000	unsigned
0x50	Encoder Mode	Register (rw)	-	-
0x51	Encoder Angle	Register (rw)	100(r)/10(w)	unsigned(r)/signed
0x52	Encoder Time Triggered Angle Increment	Register (rw)	10	signed
0x53	Encoder Velocity Filter Q	Register (rw)	10000	unsigned
0x54	Encoder Velocity Noise Threshold	Register (rw)	10	unsigned
0x55	Encoder Direction	Register (rw)	-	-
0x56	Encoder Transmission Ratio	Register (rw)	10000	unsigned
0x58	Encoder Angle Filter Q	Register (rw)	10000	unsigned
0x59	Encoder Angle Resolution	Register (rw)	100	unsigned



0x5E	Encoder Offset	Register (rw)	10	signed
0x5F	Encoder Scaling	Register (rw)	100	signed
0x60	Degauss Mode	Register (rw)	-	-
0x61	Degauss Frequency	Register (rw)	1	unsigned
0x62	Degauss Current	Register (rw)	1000	unsigned
0x63	Degauss Duration	Register (rw)	1000	unsigned
0x6A	Tick Index Count Mode	Register (rw)	1000	signed
0x6B	Tick Degauss Current	Register (rw)	1000	signed
0x6C	Tick Degauss Duration	Register (rw)	1000	unsigned
0x6D	Tick Exp. Ramp Down Velocity Threshold	Register (rw)	10	unsigned
0x6E	Tick Exp. Ramp Down Velocity Factor	Register (rw)	100	unsigned
0x6F	Tick Exp. Ramp Down Velo. Current Perc. Min	Register (rw)	100	unsigned
0x70	Tick Mode	Register (rw)	-	-
0x71	Tick Enable	Register (rw)	-	-
0x72	Tick Current	Register (rw)	1000	signed
0x73	Tick Angle CW	Register (rw)	10	unsigned
0x74	Tick Duration Min	Register (rw)	1000	unsigned
0x75	Tick Duration Max	Register (rw)	1000	unsigned
0x76	Tick Velocity Factor	Register (rw)	1000	signed
0x77	Tick Index Window	Register (rw)	10	unsigned
0x78	Tick Stickiness Prevention Factor	Register (rw)	100	unsigned
0x79	Tick Angle CCW	Register (rw)	10	unsigned
0x7A	Tick Active Direction	Register (rw)	-	-
0x7B	Tick Freewheeling Velocity Threshold	Register (rw)	10	unsigned
0x7C	Tick Freewheeling Extension Time	Register (rw)	1000	unsigned
0x7D	Tick Window	Register (rw)	10	unsigned
0x7E	Tick Start Angle	Register (rw)	10	signed
0x7F	Tick Stop Angle	Register (rw)	10	signed
0x81	Barrier Enable	Register (rw)	-	-
0x82	Barrier Current	Register (rw)	1000	signed
0x83	Barrier Start Angle	Register (rw)	10	signed
0x84	Barrier Stop Angle	Register (rw)	10	signed
0x8C	Barrier Polarity	Register (rw)	-	-
0x8D	Barrier Limit Angle	Register (rw)	-	-
0x90	Current Enable	Register (rw)	-	-
0x91	Current Start Angle	Register (rw)	10	signed
0x92	Current Stop Angle	Register (rw)	10	signed
0x93	Current Start Current CW	Register (rw)	1000	signed
0x94	Current Stop Current CW	Register (rw)	1000	signed
0x95	Current Start Current CCW	Register (rw)	1000	signed
0x96	Current Stop Current CCW	Register (rw)	1000	signed
0x97	Current Freewheeling Velocity Threshold	Register (rw)	10	unsigned
0x98	Current Freewheeling Extension Time	Register (rw)	1000	unsigned
0x99	Current Active Direction	Register (rw)	-	-
0xA0	Torque Enable	Register (rw)	-	-
0xA1	Torque Start Angle	Register (rw)	10	signed
0xA2	Torque Stop Angle	Register (rw)	10	signed
0xA3	Torque Start Factor CW	Register (rw)	10000	signed
0xA4	Torque Stop Factor CW	Register (rw)	10000	signed
0xA5	Torque Start Factor CCW	Register (rw)	10000	signed
0xA6	Torque Stop Factor CCW	Register (rw)	10000	signed
0xA7	Torque Freewheeling Velocity Threshold	Register (rw)	10	unsigned
0xA8	Torque Freewheeling Extension Time	Register (rw)	1000	unsigned
0xA9	Torque Active Direction	Register (rw)	-	-
0xB0	Lock Enable	Register (rw)	-	-
0xB1	Lock Direction	Register (rw)	-	-
0xB2	Lock Current	Register (rw)	1000	signed
0xB8	Freewheeling Enable	Register (rw)	-	-
0xB9	Freewheeling Start Velocity Threshold	Register (rw)	10	unsigned
0xBA	Freewheeling Friction	Register (rw)	1000	unsigned
0xBB	Freewheeling Damping	Register (rw)	1000	unsigned
0xBC	Freewheeling Inertia	Register (rw)	1000	unsigned
0xC0	Single Tick Enable	Register (rw)	-	-
0xC1	Single Tick Active Direction	Register (rw)	-	-
0xC2	Single Tick Angle	Register (rw)	10	signed
0xC3	Single Tick Current	Register (rw)	1000	signed
0xC4	Single Tick Mode	Register (rw)	-	-

0xC5	Single Tick Duration Min	Register (rw)	1000	unsigned
0xC6	Single Tick Duration Max	Register (rw)	10000	unsigned
0xC7	Single Tick Window	Register (rw)	10	unsigned
0xC8	Single Tick Velocity Factor	Register (rw)	1000	signed
0xCA	Haptics Generator Enable	Register (rw)	-	-
0xE0	Report Encoder Angle	Report Reg. (r)	100	unsigned
0xE1	Report Encoder Velocity	Report Reg. (r)	1	signed
0xE2	Report Push Pull Value	Report Reg. (r)	10000	unsigned
0xE3	Report Tick Index	Report Reg. (r)	1	signed
0xE4	Report Push Pull State	Report Reg. (r)	-	-
0xE5	Report Encoder Multi Turn Counter	Report Reg. (r)	1	signed
0xE6	Report Encoder Temperature	Report Reg. (r)	100	signed
0xE7	Report Coil Current	Report Reg. (r)	1000	signed
0xEA	Report Device Error Status	Report Reg. (r)	-	-
0xEB	Report Encoder Calibration Status	Report Reg. (r)	-	-
0xEC	Report Coil Resistance	Report Reg. (r)	100	unsigned
0xED	Report Push Calibration Status	Report Reg. (r)	-	-
0xEF	Report Device Connection State	Report Reg. (r)	-	-
0xF4	Report Supply Voltage	Report Reg. (r)	1000	unsigned
0xF5	Report Degauss Status	Report Reg. (r)	-	-

## 4 Status Reply

### 4.1 Status Reply

A status reply of a respective Command / SetRegister / GetRegister is sent in case of an error.

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	Type	Reply Status	LRC	Stop

#### Type

Command / Register Type

#### Reply Status

REPLY_STATUS_OK	0x00
REPLY_STATUS_ERROR	0x01
REPLY_STATUS_NOT_SUPPORTED	0x02

## 5 Commands

### 5.1 Reboot System

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x01	-	-	LRC	Stop

#### Note

After a successful reboot the operating time is reset.

### 5.2 Load Default Values

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x02	-	-	LRC	Stop

#### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x02	0x00	LRC	Stop

#### Note

Loads default settings of the HAPTICORE control device as well as HAPTICORE device specific settings, which are stored on the HAPTICORE device.

### 5.3 Get Register Value

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	[Index]	Type	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	Type	[Index] / Register Value High	Register Value Low	LRC	Stop

#### Note

If a register type is unknown a status reply with DATA\_HIGH = type of the requested register and DATA\_LOW = STATUS\_NOT\_SUPPORTED will be returned.

#### Example

Type = 0x51 → Reply contains value of encoder angle register in DATA\_LOW and DATA\_HIGH.

### 5.4 Degauss

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x04	-	-	LRC	Stop

#### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x02	Status	LRC	Stop

#### Status

REPLY\_STATUS\_OK            0x00  
 REPLY\_STATUS\_ERROR        0x01

#### Note

Triggers coil degaussing based on degaussing settings.

## 5.5 Calibrate Encoder

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x05	-	-	LRC	Stop

#### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x05	Status	LRC	Stop

#### Status

REPLY\_STATUS\_OK                    0x00  
 REPLY\_STATUS\_ERROR                0x01  
 REPLY\_STATUS\_NOT\_SUPPORTED       0x02

#### Note

Starts the encoder calibration routine. The HAPTICORE device must be turned 3 full turns until the calibration routine is complete.

## 5.6 Set RGB LEDs

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x06	-	-	LRC	Stop

#### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x06	0x00	LRC	Stop

#### Note

Applies RGB LED presets which were previously set with RGB LED registers REG\_LED\_RGB\_RED, REG\_LED\_RGB\_GREEN and REG\_LED\_RGB\_BLUE.

## 5.7 Disable All Haptic Functions

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x08	-	-	LRC	Stop

### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x08	0x00	LRC	Stop

### Note

Sets the enable registers of all haptic functions to false.

## 5.8 Start/Stop Power Measurement

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x09	-	-	LRC	Stop

### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x09	0x00	LRC	Stop

### Note

Starts/Stops power measurement. Results can be read out via registers REG\_POWER\_MEASUREMENT\_PEAK and REG\_POWER\_MEASUREMENT\_RMS.

## 5.9 Calibrate Push Pull

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x0E	-	Calibration Type	LRC	Stop

### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x0E	Status	LRC	Stop

### Calibration Type

CALIBRATION\_TYPE\_PUSH    0x00  
CALIBRATION\_TYPE\_PULL    0x01

### Status

REPLY\_STATUS\_OK            0x00  
REPLY\_STATUS\_ERROR        0x01  
REPLY\_STATUS\_NOT\_SUPPORTED 0x02

### Note

Starts the push/pull calibration routine. The HAPTICORE device must be pushed/pulled 3 times until the calibration routine is complete.

## 5.10 Hapticore Power Supply

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x0F	-	Enable	LRC	Stop

### Status Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x00	0x0F	Status	LRC	Stop

### Get Register Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x0F	0x00	Enable	LRC	Stop

### Enable

HAPTICORE\_POWER\_DISABLED    0x00  
HAPTICORE\_POWER\_ENABLED     0x01

### Status

REPLY\_STATUS\_OK            0x00  
REPLY\_STATUS\_ERROR        0x01

### Note

This command is only supported by the HAPTICORE Control Unit Pro and the HAPTICORE Controller +24V PCB.

## 6 Register (readonly)

### 6.1 Controller ID

Returns the ID of the HAPTICORE control device.

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x10	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x10	-	Device ID	LRC	Stop

#### Controller ID

CONTROLLER_ID_UNKNOWN	0x00
CONTROLLER_ID_HAPTICORE_CONTROL_UNIT	0x05
CONTROLLER_ID_HAPTICORE_CONTROL_UNIT_PRO	0x06

### 6.2 Firmware Version

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x12	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x12	Firmware Version Major	Firmware Version Minor	LRC	Stop

#### Example

Firmware v3.1: Version Major = 0x03, Version Minor = 0x01

### 6.3 Communication Protocol Version

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x13	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x13	Communication Version Major	Communication Version Minor	LRC	Stop

#### Example

Communication Protocol v3.4: Version Major = 0x03, Version Minor = 0x04



## 6.4 Hapticore Library Version

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x14	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x14	Library Version Major	Library Version Minor	LRC	Stop

### Example

Hapticore Library v3.5: Version Major = 0x03, Version Minor = 0x05

## 6.5 Controller Hardware Revision

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x15	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x15	Controller Hardware Revision High	Controller Hardware Revision Low	LRC	Stop

### Example

Returns Controller Hardware Revision as integer.

## 6.6 Hapticore Serial Number

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	Index	0x16	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x16	Index	Data[Index]	LRC	Stop

### Note

The 11 Bytes will be returned byte-wise and are ASCII encoded with null termination. The index is used to get the individual bytes. e.g. "1364AAAAPC\0"

## 6.7 Hapticore Product Name

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	Index	0x1A	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x1A	Index	Data[Index]	LRC	Stop

### Note

The 24 Bytes will be returned byte-wise and are ASCII encoded with null termination. The index is used to get the individual bytes. e.g. "HAPTICORE 34-P001X2 A1\0"

## 6.8 Hapticore Item Number

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	Index	0x1E	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x1E	Index	Data[Index]	LRC	Stop

### Note

The 14 Bytes will be returned byte-wise and are ASCII encoded with null termination. The index is used to get the individual bytes. e.g. "2000001.AB\0"

## 6.9 Operating Time 1

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x20	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x20	Operating Time 1 High	Operating Time 1 Low	LRC	Stop

### Note

Operating time in seconds. Maximum Operating time = 136.2 years.

### Example

$\text{OperatingTime} = (\text{OperatingTime2High} \ll 24) | (\text{OperatingTime2Low} \ll 16) | (\text{OperatingTime1High} \ll 8) | \text{OperatingTime1Low}$

## 6.10 Operating Time 2

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x21	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x21	Operating Time 2 High	Operating Time 2 Low	LRC	Stop

#### Note

Operating time in seconds. Maximum Operating time = 136.2 years.

#### Example

$\text{OperatingTime} = (\text{OperatingTime2High} \ll 24) | (\text{OperatingTime2Low} \ll 16) | (\text{OperatingTime1High} \ll 8) | \text{OperatingTime1Low}$

## 6.11 Encoder Temperature

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x22	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x22	Encoder Temperature High	Encoder Temperature Low	LRC	Stop

#### Note

Encoder Temperature in °C multiplied by 10, converted as 16 bit signed integer and split into high and low byte.

#### Example

Encoder Temperature = 23.5 °C → 235 → EncoderTemperatureHigh = 0x00/0d, EncoderTemperatureLow = 0xEB/235d

## 6.12 Power Measurement Peak

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x23	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x23	Power Measurement Peak High	Power Measurement Peak Low	LRC	Stop

#### Note

Power Measurement Peak in Watt multiplied by 100, converted to 16 bit unsigned integer and split into high and low byte.

## 6.13 Power Measurement RMS

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0x24	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x24	Power Measurement RMS High	Power Measurement RMS Low	LRC	Stop

#### Note

Power Measurement RMS in Watt multiplied by 100, converted to 16 bit unsigned integer and split into high and low byte.

## 7 Register (read-/writeable)

### 7.1 Hapticore System Configuration

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x29	-	System Configuration	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x29	-	System Configuration	LRC	Stop

#### Note

Returns system configuration as 8 bit integer. This value is used to distinguish HAPTICORES in software (e.g. different caps, mounting etc.)

### 7.2 Report Type

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x30	-	Report Type	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x30	-	Report Type	LRC	Stop

#### Report Type

REPORT\_TYPE\_CYCLIC    0x00  
REPORT\_TYPE\_ACYCLIC   0x01

### 7.3 Report Flags

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x31	Report Flags High	Report Flags Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x31	Report Flags High	Report Flags Low	LRC	Stop

#### Report Flags

REPORT_NOTHING	0x0000
REPORT_ENCODER_ANGLE	0x0001
REPORT_ENCODER_VELOCITY	0x0002
REPORT_DEVICE_ERROR_STATUS	0x0004
REPORT_TICK_INDEX	0x0008
REPORT_PUSH_PULL_STATE	0x0010
REPORT_ENCODER_MULTI_TURN_COUNT	0x0020
REPORT_COIL_CURRENT	0x0040
REPORT_ENCODER_TEMPERATURE	0x0080
REPORT_CALIBRATION_STATUS	0x0200
REPORT_COIL_RESISTANCE	0x0400
REPORT_DEVICE_CONNECTION_STATE	0x0800
REPORT_SUPPLY_VOLTAGE	0x1000
REPORT_DEGAUSS_STATUS	0x2000
REPORT_PUSH_PULL_VALUE	0x4000

#### Note

If REPORT\_CALIBRATION\_STATUS is enabled the reports Encoder Calibration Status, Push Calibration Status and Pull Calibration Status are sent.

## 7.4 Report Frequency

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x32	Report Frequency High	Report Frequency Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x32	Report Frequency High	Report Frequency Low	LRC	Stop

#### Note

Report frequency in Hz from 1 to 65535 converted to 16 bit unsigned integer and split into high and low byte. Cannot be set to 0.

#### Example

Report Frequency = 100 Hz → ReportFrequencyHigh = 0x00/0d, ReportFrequencyLow = 0x64/100d

## 7.5 Clutch Activation After Idle

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x71	-	Clutch Activation	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x71	-	Clutch Activation	LRC	Stop

#### Clutch Activation

CLUTCH_ACTIVATION_DISABLED	0x00
CLUTCH_ACTIVATION_ENABLED	0x01

## 7.6 Preset RGB LEDs Red

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x36	LED Index	LED Indensity Red	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x36	LED Index	LED Indensity Red	LRC	Stop

### Note

All RGB LEDs are set simultaneously with LED Index = 0xFF. LED Indensity range is 0-255. The presets are applied after the Set RGB LEDs command is sent.

## 7.7 Preset RGB LEDs Green

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x37	LED Index	LED Indensity Green	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x37	LED Index	LED Indensity Green	LRC	Stop

### Note

All RGB LEDs are set simultaneously with LED Index = 0xFF. LED Indensity range is 0-255. The presets are applied after the Set RGB LEDs command is sent.

## 7.8 Preset RGB LEDs Blue

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x38	LED Index	LED Indensity Blue	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x38	LED Index	LED Indensity Blue	LRC	Stop

### Note

All RGB LEDs are set simultaneously with LED Index = 0xFF. LED Indensity range is 0-255. The presets are applied after the Set RGB LEDs command is sent.

## 7.9 Clutch Deactivation Blanking Duration

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x39	Blanking Duration High	Blanking Duration Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x39	Blanking Duration High	Blanking Duration Low	LRC	Stop

### Note

Clutch Deactivation Blanking Duration in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Blanking Duration = 0.995 → 995 → Blanking Duration High = 0x03/03d, Blanking Duration Low = 0xE3/227d

## 7.10 Clutch Base Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3A	Clutch Base Current High	Clutch Base Current Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3A	Clutch Base Current High	Clutch Base Current Low	LRC	Stop

### Note

Clutch Base Current in A multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Clutch Base Current = 0.1 A → Clutch Base Current High = 0x00/0d, Clutch Base Current Low = 0xE8/100d  
Clutch Base Current = (Clutch Base Current High « 8) | Clutch Base Current Low

## 7.11 Clutch Activation Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3B	Activation Current High	Activation Current Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3B	Activation Current High	Activation Current Low	LRC	Stop

### Note

Activation Current in A multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Activation Current = 1 A → Activation Current High = 0x03/3d, Activation Current Low = 0x64/232d  
Activation Current = (Activation Current High « 8) | Activation Current Low

## 7.12 Clutch Activation Duration

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3C	Activation Duration High	Activation Duration Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3C	Activation Duration High	Activation Duration Low	LRC	Stop

### Note

Activation Duration in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Activation Duration = 0.995 → 995 → Activation Duration High = 0x03/03d, Activation Duration Low = 0xE3/227d

## 7.13 Idle Detection Velocity Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3D	Idle Detection Velocity Threshold High	Idle Detection Velocity Threshold Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3D	Idle Detection Velocity Threshold High	Idle Detection Velocity Threshold Low	LRC	Stop

### Note

If the encoder velocity for the set idle detection timeout is below the idle detection velocity threshold, the idle state is detected and the haptic is disabled. As soon as the velocity threshold is exceeded, the set haptic mode is continued.

## 7.14 Idle Detection Timeout

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3E	Idle Detection Timeout High	Idle Detection Timeout Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3E	Idle Detection Timeout High	Idle Detection Timeout Low	LRC	Stop

### Note

If the encoder velocity for the set idle detection timeout is below the idle detection velocity threshold, the idle state is detected and the haptic is disabled. As soon as the velocity threshold is exceeded, the set haptic mode is continued.



## 7.15 Idle Detection Current Ramp Down Slope

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3F	Idle Detection Current Ramp Down Slope High	Idle Detection Current Ramp Down Slope Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x3F	Idle Detection Current Ramp Down Slope High	Idle Detection Current Ramp Down Slope Low	LRC	Stop

### Note

Idle Detection Ramp Down Slope in Ampere per second multiplied by 10, converted to 16 bit signed integer and split into high and low byte. Note that the signedness of this parameter is handled in firmware, so that the current always ramps down.

### Example

Idle Detection Ramp Down Slope = 1 A/s  $\rightarrow$  1000  $\rightarrow$  Idle Detection Down Slope High = 0x03/3d, Idle Detection Ramp Down Slope Low = 0xE8/232d

## 7.16 Current Controller Kp

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x42	Current Controller Kp High	Current Controller Kp Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x42	Current Controller Kp High	Current Controller Kp Low	LRC	Stop

### Note

Current Controller Kp multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Current Controller Kp = 2.1  $\rightarrow$  Current Controller Kp High = 0x00/0d, Current Controller Kp Low = 0x15/21d

## 7.17 Current Controller Idle Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x43	Idle Current High	Idle Current Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x43	Idle Current High	Idle Current Low	LRC	Stop

### Note

Idle current of current controller in Ampere multiplied by 1000, converted to 16 bit signed integer and split into high and low byte.

### Example

Current Controller Idle Current = 0.008 A → Current Controller Idle Current High = 0x00/0d, Current Controller Idle Current Low = 0x08/8d

## 7.18 Current Controller Mode

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x44	-	Current Controller Mode	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x44	-	Current Controller Mode	LRC	Stop

### Current Controller Mode

CURRENT_CONTROLLER_MODE_DUTY_CYCLE	0x00
CURRENT_CONTROLLER_MODE_P	0x01
CURRENT_CONTROLLER_MODE_PI	0x02
CURRENT_CONTROLLER_MODE_SETTER	0x03
CURRENT_CONTROLLER_MODE_P_CURRENT_OBSERVER	0x04
CURRENT_CONTROLLER_MODE_PI_CURRENT_OBSERVER	0x05

### Note

If current controller mode is set to DUTY\_CYCLE, any current setting is interpreted as duty cycle (e.g. 0.1 A → 10 % duty cycle) and limited to a range of +/- 1.0.

If current controller mode is set to SETTER, the current is set according to the set coil resistance and the set coil driver supply voltage. The P\_CURRENT\_OBSERVER and PI\_CURRENT\_OBSERVER modes are using the coil resistance and coil inductance settings to estimate the resulting current. This mode are used when no current sensing is available.

## 7.19 Current Controller Ki

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x44	Current Controller Ki High	Current Controller Ki Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x44	Current Controller Ki High	Current Controller Ki Low	LRC	Stop

### Note

Current Controller Ki, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Current Controller Ki = 250 → Current Controller Ki High = 0x00/0d, Current Controller Ki Low = 0xFA/250d

## 7.20 Current Controller Coil Resistance

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x46	Current Controller Coil Resistance High	Current Controller Coil Resistance Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x46	Current Controller Coil Resistance High	Current Controller Coil Resistance Low	LRC	Stop

#### Note

Sets the coil resistance for current controller mode: Setter, P, P Current Observer, PI Current Observer  
Current Controller Coil Resistance multiplied by 100, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Current Controller Coil Resistance = 4.75 Ohm → Current Controller Coil Resistance High = 0x01/01d, Current Controller Coil Resistance Low = 0xDB/219d

## 7.21 Current Controller Coil Inductance

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x47	Current Controller Coil Inductance High	Current Controller Coil Inductance Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x47	Current Controller Coil Inductance High	Current Controller Coil Inductance Low	LRC	Stop

#### Note

Sets the coil inductance for current controller mode: P Current Observer, PI Current Observer  
Current Controller Coil Inductance multiplied by 10000, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Current Controller Coil Inductance = 0.0012 H → Current Controller Coil Inductance High = 0x00/0d, Current Controller Coil Inductance Low = 0x0C/12d

## 7.22 Encoder Angle Filter R

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x48	Encoder Angle Filter R High	Encoder Angle Filter R Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x48	Encoder Angle Filter R High	Encoder Angle Filter R Low	LRC	Stop

#### Note

Encoder Angle Filter R multiplied by 10, converted to 16bit unsigned integer and split into high and low byte.

#### Example

Encoder Angle Filter R = 10 → Encoder Angle Filter R High = 0x00/0d, Encoder Angle Filter R Low = 0x64/100d

## 7.23 Encoder Velocity Filter R

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x49	Encoder Velocity Filter R High	Encoder Velocity Filter R Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x49	Encoder Velocity Filter R High	Encoder Velocity Filter R Low	LRC	Stop

#### Note

Encoder Velocity Filter R multiplied by 10, converted to 16bit unsigned integer and split into high and low byte.

#### Example

Encoder Filter Velocity R = 10 → Encoder Velocity Filter R High = 0x00/0d, Encoder Filter Velocity R Low = 0x64/100d

## 7.24 Current Controller Supply Voltage Override

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x49	-	Current Controller Supply Voltage Override	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x49	-	Current Controller Supply Voltage Override	LRC	Stop

### Current Controller Supply Voltage Override

CURRENT\_CONTROLLER\_SUPPLY\_VOLTAGE\_OVERRIDE\_DISABLED 0x00  
CURRENT\_CONTROLLER\_SUPPLY\_VOLTAGE\_OVERRIDE\_ENABLED 0x01

#### Note

If Current Controller Supply Voltage Override is enabled the value set in the register Current Controller Supply Voltage is used for the current controller instead of the measured supply voltage of the coil driver.

## 7.25 Current Controller Supply Voltage

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x49	Current Controller Supply Voltage High	Current Controller Supply Voltage Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x49	Current Controller Supply Voltage High	Current Controller Supply Voltage Low	LRC	Stop

#### Note

Current Controller Supply Voltage multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Current Controller Supply Voltage = 12 → 12000 → Current Controller Supply Voltage High = 0x2E/46d, Current Controller Supply VoltageR Low = 0xE0/224d

## 7.26 Coil Driver PWM Polarity

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x4C	-	Coil Driver PWM Polarity	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x4C	-	Coil Driver PWM Polarity	LRC	Stop

#### PWM Polarity

COIL\_DRIVER\_PWM\_POLARITY\_DEFAULT 0x00

COIL\_DRIVER\_PWM\_POLARITY\_REVERSE 0x01

## 7.27 Coil Driver Stop Mode

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x4D	-	Coil Driver Stop Mode	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x4D	-	Coil Driver Stop Mode	LRC	Stop

#### Coil Driver Stop Mode

COIL\_DRIVER\_STOP\_MODE\_SLOW\_DECAY 0x00

COIL\_DRIVER\_STOP\_MODE\_FAST\_DECAY 0x01

## 7.28 Current Sense Polarity

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x4E	-	Current Sense Polarity	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x4E	-	Current Sense Polarity	LRC	Stop

#### Current Sense Polarity

CURRENT\_SENSE\_POLARITY\_DEFAULT 0x00

CURRENT\_SENSE\_POLARITY\_REVERSE 0x01

## 7.29 Encoder Mode

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x50	-	Encoder Mode	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x50	-	Encoder Mode	LRC	Stop

#### Encoder Mode

ENCODER\_MODE\_DEFAULT 0x01

ENCODER\_MODE\_EMULATION 0x02

#### Note

If encoder mode is set to emulation mode the setting Time Triggered Angle Increment will be used to automatically increment the encoder angle.

## 7.30 Encoder Angle

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x51	Encoder Angle High	Encoder Angle Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x51	Encoder Angle High	Encoder Angle Low	LRC	Stop

#### Note

Write: Encoder Angle in degrees multiplied by 10, converted to 16bit signed integer and split into high and low byte. Read: Encoder Angle interpreted as 16bit unsigned integer, divided by 100.

#### Example

Write: Encoder Angle =  $123.4^{\circ} \rightarrow 12340 \rightarrow$  Encoder Angle High = 0x04/48d, Encoder Angle Low = 0xd2/210d Read: Encoder Angle High = 0x30/48d, Encoder Angle Low = 0x35/53d  $\rightarrow 12341 \rightarrow$  Encoder Angle =  $123.41^{\circ}$

## 7.31 Encoder Time Triggered Angle Increment

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x52	Encoder Time Triggered Angle Increment High	Encoder Time Triggered Angle Increment Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x52	Encoder Time Triggered Angle Increment High	Encoder Time Triggered Angle Increment Low	LRC	Stop

### Note

Encoder Time Triggered Angle in degree per second multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

### Example

Encoder Time Triggered Angle Increment =  $0.1^{\circ}/s \rightarrow 50 \rightarrow$  Encoder Time Triggered Angle Increment High = 0x00/00d, Encoder Time Triggered Angle Increment Low = 0x32/50d

## 7.32 Encoder Velocity Filter Q

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x53	Encoder Velocity Filter Q High	Encoder Velocity Filter Q Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x53	Encoder Velocity Filter Q High	Encoder Velocity Filter Q Low	LRC	Stop

### Note

Encoder Velocity Filter Q multiplied by 10000, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Encoder Velocity Filter Q =  $0.0005 \rightarrow 5 \rightarrow$  Encoder Velocity Filter Q High = 0x00/0d, Encoder Velocity Filter Q Low = 0x05/5d

## 7.33 Encoder Velocity Noise Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x54	Encoder Velocity Noise Threshold High	Encoder Velocity Noise Threshold Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x54	Encoder Velocity Noise Threshold High	Encoder Velocity Noise Threshold Low	LRC	Stop

### Note

Encoder Velocity Noise Threshold in degree per second multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Encoder Velocity Noise Threshold = 3 → 30 → Encoder Velocity Noise Threshold High = 0x00/0d, Encoder Velocity Noise Threshold Low = 0x1E/30d

## 7.34 Encoder Direction

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x55	-	Encoder Direction	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x55	-	Encoder Direction	LRC	Stop

#### Encoder Direction

ENCODER\_DIRECTION\_DEFAULT 0x00  
ENCODER\_DIRECTION\_REVERSE 0x01

#### Note

If encoder mode is set to emulation mode the setting Time Triggered Angle Increment will be used to automatically increment the encoder angle.

## 7.35 Encoder Angle Filter Q

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x58	Encoder Angle Filter Q High	Encoder Angle Filter Q Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x58	Encoder Angle Filter Q High	Encoder Angle Filter Q Low	LRC	Stop

#### Note

Encoder Angle Filter Q multiplied by 10000, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Encoder Angle Filter Q = 0.001 → 10 → Encoder Angle Filter Q High = 0x00/0d, Encoder Angle Filter Q Low = 0x64/100d

## 7.36 Encoder Angle Resolution

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x59	Encoder Angle Resolution High	Encoder Angle Resolution Low	LRC	Stop

#### Reply



Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x59	Encoder Angle Resolution High	Encoder Angle Resolution Low	LRC	Stop

#### Note

Encoder Angle Resolution in degree multiplied by 100, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Encoder Angle Resolution = 0.1 → Encoder Angle Resolution High = 0x00/0d, Encoder Angle Resolution Low = 0x0A/10d

## 7.37 Encoder Offset

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x5E	Encoder Offset High	Encoder Offset Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x5F	Encoder Offset High	Encoder Offset Low	LRC	Stop

#### Note

Encoder Offset in degree multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Encoder Offset = 1 → Encoder Offset High = 0x00/0d, Encoder Offset Low = 0x0A/10d

## 7.38 Encoder Scaling

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x5F	Encoder Scaling High	Encoder Scaling Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x5F	Encoder Scaling High	Encoder Scaling Low	LRC	Stop

**Note** Encoder Scaling multiplied by 100, converted to 16 bit signed integer and split into high and low byte. **Example**

Encoder Scaling = 0.1 → Encoder Scaling High = 0x00/0d, Encoder Scaling Low = 0x0A/10d

## 7.39 Degauss Mode

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x60	-	Degauss Mode	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x60	-	Degauss Mode	LRC	Stop

## Degauss Mode

DEGAUSS_MODE_SINE_EXPONENTIAL	0x00
DEGAUSS_MODE_SINE_LINEAR	0x01
DEGAUSS_MODE_SQUARE_EXPONENTIAL	0x02
DEGAUSS_MODE_SQUARE_LINEAR	0x03

## 7.40 Degauss Frequency

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x61	Degauss Frequency High	Degauss Frequency Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x61	Degauss Frequency High	Degauss Frequency Low	LRC	Stop

### Note

Degauss Frequency in Hz, converted to 16 bit unsigned integer and split into high and low byte

### Example

Degauss Frequency = 100 Hz → Degauss Frequency High = 0x00/0d, Degauss Frequency Low = 0x64/100d

## 7.41 Degauss Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x62	Degauss Current High	Degauss Current Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x62	Degauss Current High	Degauss Current Low	LRC	Stop

### Note

Degauss Current in A multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Degauss Current = 1 A → Degauss Current High = 0x03/3d, Degauss Current Low = 0xE8/232d  
Degauss Current = (Degauss Current High « 8) | Degauss Current Low

## 7.42 Degauss Duration

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x63	Degauss Duration High	Degauss Duration Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x63	Degauss Duration High	Degauss Duration Low	LRC	Stop

#### Note

Degauss Duration in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Degauss Duration = 0.995 → 995 → Degauss Duration High = 0x03/03d, Degauss Duration Low = 0xE3/227d

### 7.43 Tick Index Count Mode

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6A	Tick Index Count Mode High	Tick Index Count Mode Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6A	Tick Index Count Mode High	Tick Index Count Mode Low	LRC	Stop

#### Tick Index Count Mode

TICK\_INDEX\_COUNT\_MODE\_1 0x00

TICK\_INDEX\_COUNT\_MODE\_2 0x01

### 7.44 Tick Degauss Current

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6B	Tick Degauss Current High	Tick Degauss Current Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6B	Tick Degauss Current High	Tick Degauss Current Low	LRC	Stop

#### Note

Tick Degauss Current in Ampere multiplied by 1000, converted as 16 bit signed integer and split into high and low byte.

#### Example

Tick Degauss Current = 0.1 A → 100 → Tick Degauss Current High = 0x00/0d, Tick Degauss Current Low = 0x64/100d

### 7.45 Tick Degauss Duration

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6C	Tick Degauss Duration High	Tick Degauss Duration Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6C	Tick Degauss Duration High	Tick Degauss Duration Low	LRC	Stop

#### Note

Tick Degauss Duration in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.46 Tick Exponential Ramp Down Velocity Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6D	Tick Exponential Ramp Down Velocity Threshold High	Tick Exponential Ramp Down Velocity Threshold Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6D	Tick Exponential Ramp Down Velocity Threshold High	Tick Exponential Ramp Down Velocity Threshold Low	LRC	Stop

### Note

Tick Exponential Ramp Down Velocity Threshold multiplied by 10, converted to 16bit unsigned integer and split into high and low byte.

### Example

Tick Exponential Ramp Down Velocity Threshold = 100 → 1000 → Threshold High = 0x03/03d, Threshold Low = 0xE8/232d

## 7.47 Tick Exponential Ramp Down Factor

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6E	Tick Exponential Ramp Down Factor High	Tick Exponential Ramp Down Factor Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6E	Tick Exponential Ramp Down Factor High	Tick Exponential Ramp Down Factor Low	LRC	Stop

### Note

Tick Exponential Ramp Down Factor multiplied by 100, converted to 16bit unsigned integer and split into high and low byte.

### Example

Tick Exponential Ramp Down Factor = 0.1 → 10 → Factor High = 0x00/0d, Factor Low = 0x0A/10d

## 7.48 Tick Exponential Ramp Down Current Min Percent

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6F	Tick Exponential Ramp Down Current Min Percent High	Tick Exponential Ramp Down Current Min Percent Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x6F	Tick Exponential Ramp Down Current Min Percent High	Tick Exponential Ramp Down Current Min Percent Low	LRC	Stop

### Note

Tick Exponential Ramp Down Current Min Percent multiplied by 100, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Tick Exponential Ramp Down Current Min Percent = 10  $\rightarrow$  1000  $\rightarrow$  Factor High = 0x03/3d, Factor Low = 0xE8/232d

## 7.49 Tick Mode

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x70	-	Tick Mode	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x70	-	Tick Mode	LRC	Stop

### Tick Mode

TICK\_MODE\_1 0x00  
TICK\_MODE\_2 0x01  
TICK\_MODE\_3 0x02  
TICK\_MODE\_4 0x03

### Note

Only one mode can be set at a time.

## 7.50 Tick Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x71	-	Tick Enable	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x71	-	Tick Enable	LRC	Stop

### Tick Enable

TICK\_DISABLED 0x00  
TICK\_ENABLED 0x01

## 7.51 Tick Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x72	Tick Current High	Tick Current Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x72	Tick Current High	Tick Current Low	LRC	Stop

#### Note

Tick Current in Ampere multiplied by 1000, converted as 16 bit signed integer and split into high and low byte.

#### Example

Tick Current = 0.1 A  $\rightarrow$  100  $\rightarrow$  Tick Current High = 0x00/0d, Tick Current Low = 0x64/100d

## 7.52 Tick Angle CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x73	Tick Angle CW High	Tick Angle CW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x73	Tick Angle CW High	Tick Angle CW Low	LRC	Stop

#### Note

Tick Angle CW in degrees multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Tick Angle CW = 123.4°  $\rightarrow$  1234  $\rightarrow$  Tick Angle CW High = 0x04/04d, Tick Angle CW Low = 0xD2/210d

## 7.53 Tick Duration Min

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x74	Tick Duration Min High	Tick Duration Min Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x74	Tick Duration Min High	Tick Duration Min Low	LRC	Stop

#### Note

Tick Duration Minimum in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.54 Tick Duration Max

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x75	Tick Duration Max High	Tick Duration Max Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x75	Tick Duration Max High	Tick Duration Max Low	LRC	Stop

### Note

Tick Duration Maximum in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte. If Tick Duration Max < Tick Duration Min then Tick Duration Max will be set to Tick Duration Min in firmware.

## 7.55 Tick Velocity Factor

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x76	Tick Velocity Factor High	Tick Velocity Factor Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x76	Tick Velocity Factor High	Tick Velocity Factor Low	LRC	Stop

### Note

Tick Velocity Factor multiplied by 100, converted to 16 bit signed integer and split into high and low byte. Set Tick Velocity Factor to 0 to disable velocity compensation.

### Example

Tick Velocity Factor = 0.5 → 50 → Tick Velocity Factor High = 0x00/0d, Tick Velocity Factor Low = 0x32/50d

## 7.56 Tick Index Window

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x77	Tick Index Window High	Tick Index Window Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x77	Tick Index Window High	Tick Index Window Low	LRC	Stop

### Note

Tick Index Window in degrees multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Tick Index Window = 5° → 50 → Tick Index Window High = 0x00/0d, Tick Index Window Low = 0x32/50d

## 7.57 Tick Stickiness Prevention Factor

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x78	Tick Stickiness Prevention Factor High	Tick Stickiness Prevention Factor Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x78	Tick Stickiness Prevention Factor High	Tick Stickiness Prevention Factor Low	LRC	Stop

### Note

Tick Stickiness Prevention Factor multiplied by 100, converted to 16 bit unsigned integer and split into high and low byte.

## 7.58 Tick Angle CCW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x79	Tick Angle CCW High	Tick Angle CCW Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x79	Tick Angle CCW High	Tick Angle CCW Low	LRC	Stop

### Note

Tick Angle CCW in degrees multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Tick Angle CCW =  $123.4^{\circ} \rightarrow 1234 \rightarrow$  Tick Angle CCW High = 0x04/04d, Tick Angle CCW Low = 0xD2/210d

## 7.59 Tick Active Direction

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7A	-	Tick Active Direction	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7A	-	Tick Active Direction	LRC	Stop

### Tick Active Direction

TICK\_ACTIVE\_DIRECTION\_BOTH 0x00  
TICK\_ACTIVE\_DIRECTION\_CW 0x01  
TICK\_ACTIVE\_DIRECTION\_CCW 0x02

## 7.60 Tick Freewheeling Velocity Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7B	Tick Freewheeling Velocity Threshold High	Tick Freewheeling Velocity Threshold Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7B	Tick Freewheeling Velocity Threshold High	Tick Freewheeling Velocity Threshold Low	LRC	Stop

### Note



Tick Freewheeling Velocity Threshold in degree per second multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Tick Freewheeling Velocity Threshold = 100 → 1000 → Tick Freewheeling Velocity Threshold High = 0x03/3d, Tick Freewheeling Velocity Threshold Low = 0xE8/232d

## 7.61 Tick Freewheeling Extension Time

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7C	Tick Freewheeling Extension Time High	Tick Freewheeling Extension Time Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7C	Tick Freewheeling Extension Time High	Tick Freewheeling Extension Time Low	LRC	Stop

### Note

Tick Freewheeling Extension Time in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.62 Tick Window

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7D	Tick Window High	Tick Window Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7D	Tick Window High	Tick Window Low	LRC	Stop

### Note

Tick Window in degrees multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

#### Example

Tick Window = 5° → 50 → Tick Window High = 0x00/0d, Tick Window Low = 0x32/50d

## 7.63 Tick Start Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7E	Tick Start Angle High	Tick Start Angle Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7E	Tick Start Angle High	Tick Start Angle Low	LRC	Stop

### Note

Tick Start Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

### Example

Tick Start Angle =  $5^{\circ} \rightarrow 50 \rightarrow$  Tick Start Angle High = 0x00/0d, Tick Start Angle Low = 0x32/50d

## 7.64 Tick Stop Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7E	Tick Stop Angle High	Tick Stop Angle Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x7E	Tick Stop Angle High	Tick Stop Angle Low	LRC	Stop

### Note

Tick Stop Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

### Example

Tick Stop Angle =  $5^{\circ} \rightarrow 50 \rightarrow$  Tick Stop Angle High = 0x00/0d, Tick Stop Angle Low = 0x32/50d

## 7.65 Barrier Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x81	-	Barrier Enable	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x81	-	Barrier Enable	LRC	Stop

### Barrier Enable

BARRIER\_DISABLED 0x00

BARRIER\_ENABLED 0x01

## 7.66 Barrier Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x82	Barrier Current High	Barrier Current Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x82	Barrier Current High	Barrier Current Low	LRC	Stop

### Note

Barrier Current in Ampere multiplied by 1000, converted as 16 bit signed integer and split into high and low byte.

#### Example

Barrier Current = 1 A  $\rightarrow$  1000  $\rightarrow$  Barrier Current High = 0x03/3d, Barrier Current Low = 0xE8/232d

## 7.67 Barrier Start Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x83	Barrier Start Angle High	Barrier Start Angle Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x83	Barrier Start Angle High	Barrier Start Angle Low	LRC	Stop

### Note

Barrier Start Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Barrier Start Angle = 90.0°  $\rightarrow$  900  $\rightarrow$  Barrier Start Angle High = 0x03/3d, Barrier Start Angle Low = 0x84/132d

## 7.68 Barrier Stop Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x84	Barrier Stop Angle High	Barrier Stop Angle Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x84	Barrier Stop Angle High	Barrier Stop Angle Low	LRC	Stop

### Note

Barrier Stop Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Barrier Stop Angle = 90.0°  $\rightarrow$  900  $\rightarrow$  Barrier Stop Angle High = 0x03/3d, Barrier Stop Angle Low = 0x84/132d

## 7.69 Barrier Polarity

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x8C	-	Barrier Polarity	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x8C	-	Barrier Polarity	LRC	Stop

### Barrier Enable

BARRIER_POLARITY_START_POSITIVE_STOP_POSITIVE	0x00
BARRIER_POLARITY_START_POSITIVE_STOP_NEGATIVE	0x01
BARRIER_POLARITY_START_NEGATIVE_STOP_POSITIVE	0x02
BARRIER_POLARITY_START_NEGATIVE_STOP_NEGATIVE	0x03

#### Note

If Barrier is used in combination with Current Controller Idle Current use only either BARRIER\_POLARITY\_START\_POSITIVE\_STOP\_POSITIVE or BARRIER\_POLARITY\_START\_NEGATIVE\_STOP\_NEGATIVE as setting to avoid direction dependent haptics.

## 7.70 Barrier Limit Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x8D	-	Barrier Limit Angle	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x8D	-	Barrier Limit Angle	LRC	Stop

### Barrier Limit Angle

BARRIER_LIMIT_ANGLE_DISABLED	0x00
BARRIER_LIMIT_ANGLE_ENABLED	0x01

## 7.71 Current Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x90	-	Current Enable	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x90	-	Current Enable	LRC	Stop

### Current Enable

CURRENT_DISABLED	0x00
CURRENT_ENABLED	0x01

## 7.72 Current Start Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x91	Current Start Angle High	Current Start Angle Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x91	Current Start Angle High	Current Start Angle Low	LRC	Stop

#### Note

Current Start Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Current Start Angle =  $90.0^{\circ} \rightarrow 900 \rightarrow$  Current Start Angle High = 0x03/3d, Current Start Angle Low = 0x84/132d

## 7.73 Current Stop Angle

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x92	Current Stop Angle High	Current Stop Angle Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x92	Current Stop Angle High	Current Stop Angle Low	LRC	Stop

#### Note

Current Stop Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Current Stop Angle =  $90.0^{\circ} \rightarrow 900 \rightarrow$  Current Stop Angle High = 0x03/3d, Current Stop Angle Low = 0x84/132d

## 7.74 Current Start Current CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x93	Current Start Current CW High	Current Start Current CW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x93	Current Start Current CW High	Current Start Current CW Low	LRC	Stop

#### Note

Current Start Current CW in Ampere, converted to 16 bit signed integer and split into high and low byte.

#### Example

Current Start Current CW = 1 A  $\rightarrow 1000 \rightarrow$  Current Start Current CW High = 0x03/3d, Current Start Current CW Low = 0xE8/232d

## 7.75 Current Stop Current CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x94	Current Stop Current CW High	Current Stop Current CW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x94	Current Stop Current CW High	Current Stop Current CW Low	LRC	Stop

#### Note

Current Stop Current CW in Ampere, converted to 16 bit signed integer and split into high and low byte.

#### Example

Current Stop Current CW = 1 A  $\rightarrow$  1000  $\rightarrow$  Current Stop Current CW High = 0x03/3d, Current Stop Current CW Low = 0xE8/232d

## 7.76 Current Start Current CCW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x95	Current Start Current CCW High	Current Start Current CCW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x95	Current Start Current CCW High	Current Start Current CCW Low	LRC	Stop

#### Note

Current Start Current CCW in Ampere, converted to 16 bit signed integer and split into high and low byte.

#### Example

Current Start Current CCW = 1 A  $\rightarrow$  1000  $\rightarrow$  Current Start Current CCW High = 0x03/3d, Current Start Current CCW Low = 0xE8/232d

## 7.77 Current Stop Current CCW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x96	Current Stop Current CCW High	Current Stop Current CCW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x96	Current Stop Current CCW High	Current Stop Current CCW Low	LRC	Stop

#### Note

Current Stop Current CCW in Ampere, converted to 16 bit signed integer and split into high and low byte.

#### Example

Current Stop Current CCW = 1 A  $\rightarrow$  1000  $\rightarrow$  Current Stop Current CCW High = 0x03/3d, Current Stop Current CCW Low = 0xE8/232d

## 7.78 Current Freewheeling Extension Time

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x98	Current Freewheeling Extension Time High	Current Freewheeling Extension Time Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x98	Current Freewheeling Extension Time High	Current Freewheeling Extension Time Low	LRC	Stop

## Note

Current Freewheeling Extension Time in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.79 Current Active Direction

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x99	-	Current Active Direction	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x99	-	Current Active Direction	LRC	Stop

### Current Active Direction

CURRENT\_ACTIVE\_DIRECTION\_BOTH 0x00  
CURRENT\_ACTIVE\_DIRECTION\_CW 0x01  
CURRENT\_ACTIVE\_DIRECTION\_CCW 0x02

## 7.80 Torque Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA0	-	Torque Enable	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA0	-	Torque Enable	LRC	Stop

### Torque Enable

TORQUE\_DISABLED 0x00  
TORQUE\_ENABLED 0x01

## 7.81 Torque Start Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA1	Torque Start Angle High	Torque Start Angle Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA1	Torque Start Angle High	Torque Start Angle Low	LRC	Stop

#### Note

Torque Start Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Torque Start Angle =  $90.0^{\circ} \rightarrow 900 \rightarrow$  Torque Start Angle High = 0x03/3d, Torque Start Angle Low = 0x84/132d

## 7.82 Torque Stop Angle

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x92	Torque Stop Angle High	Torque Stop Angle Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x92	Torque Stop Angle High	Torque Stop Angle Low	LRC	Stop

#### Note

Torque Stop Angle in degrees multiplied by 10, converted to 16 bit signed integer and split into high and low byte.

#### Example

Torque Stop Angle =  $90.0^{\circ} \rightarrow 900 \rightarrow$  Torque Stop Angle High = 0x03/3d, Torque Stop Angle Low = 0x84/132d

## 7.83 Torque Start Factor CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA3	Torque Start Factor CW High	Torque Start Factor CW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA3	Torque Start Factor CW High	Torque Start Factor CW Low	LRC	Stop

#### Note

Torque Start Factor CW multiplied by 10000, converted to 16 bit signed integer and split into high and low byte.

#### Example

Torque Start Factor CW =  $0.0005 \rightarrow 5 \rightarrow$  Torque Start Factor CW High = 0d/0x00, Torque Start Factor CW Low = 05d/0x05

## 7.84 Torque Stop Factor CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA4	Torque Stop Factor CW High	Torque Stop Factor CW Low	LRC	Stop

#### Reply



Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA4	Torque Stop Factor CW High	Torque Stop Factor CW Low	LRC	Stop

#### Note

Torque Stop Factor CW multiplied by 10000, converted to 16 bit signed integer and split into high and low byte.

#### Example

Torque Stop Factor CW = 0.0005  $\rightarrow$  5  $\rightarrow$  Torque Stop Factor CW High = 0d/0x00, Torque Stop Factor CW Low = 05d/0x05

## 7.85 Torque Start Factor CCW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA5	Torque Start Factor CCW High	Torque Start Factor CCW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA5	Torque Start Factor CCW High	Torque Start Factor CCW Low	LRC	Stop

#### Note

Torque Start Factor CCW multiplied by 10000, converted to 16 bit signed integer and split into high and low byte.

#### Example

Torque Start Factor CCW = 0.0005  $\rightarrow$  5  $\rightarrow$  Torque Start Factor CCW High = 0d/0x00, Torque Start Factor CCW Low = 05d/0x05

## 7.86 Torque Stop Factor CCW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA6	Torque Stop Factor CCW High	Torque Stop Factor CCW Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA6	Torque Stop Factor CCW High	Torque Stop Factor CCW Low	LRC	Stop

#### Note

Torque Stop Factor CCW multiplied by 10000, converted to 16 bit signed integer and split into high and low byte.

#### Example

Torque Stop Factor CCW = 0.0005  $\rightarrow$  5  $\rightarrow$  Torque Stop Factor CCW High = 0d/0x00, Torque Stop Factor CCW Low = 05d/0x05

## 7.87 Torque Freewheeling Velocity Threshold

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA7	Torque Freewheeling Velocity Threshold High	Torque Freewheeling Velocity Threshold Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA7	Torque Freewheeling Velocity Threshold High	Torque Freewheeling Velocity Threshold Low	LRC	Stop

## Note

Torque Freewheeling Velocity Threshold in degree per second multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

## Example

Torque Freewheeling Velocity Threshold = 100 → 1000 → Torque Freewheeling Velocity Threshold High = 0x03/3d, Torque Freewheeling Velocity Threshold Low = 0xE8/232d

## 7.88 Torque Freewheeling Extension Time

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA8	Torque Freewheeling Extension Time High	Torque Freewheeling Extension Time Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA8	Torque Freewheeling Extension Time High	Torque Freewheeling Extension Time Low	LRC	Stop

## Note

Torque Freewheeling Extension Time in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.89 Torque Active Direction

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA9	-	Torque Active Direction	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xA9	-	Torque Active Direction	LRC	Stop

### Torque Active Direction

TORQUE\_ACTIVE\_DIRECTION\_BOTH 0x00  
TORQUE\_ACTIVE\_DIRECTION\_CW 0x01  
TORQUE\_ACTIVE\_DIRECTION\_CCW 0x02

## 7.90 Lock Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB0	-	Lock Enable	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB0	-	Lock Enable	LRC	Stop

#### Lock Enable

LOCK\_DISABLED    0x00  
 LOCK\_ENABLED    0x01

## 7.91 Lock Direction

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB1	-	Lock Direction	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB1	-	Lock Direction	LRC	Stop

#### Lock Direction

LOCK\_DIRECTION\_BOTH    0x00  
 LOCK\_DIRECTION\_CW    0x01  
 LOCK\_DIRECTION\_CCW    0x02

## 7.92 Lock Current

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x82	Lock Current High	Lock Current Low	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x82	Lock Current High	Lock Current Low	LRC	Stop

#### Note

Lock Current in Ampere multiplied by 1000, converted as 16 bit signed integer and split into high and low byte.

#### Example

Lock Current = 1 A → 1000 → Lock Current High = 0x03/3d, Lock Current Low = 0xE8/232d

## 7.93 Freewheeling Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB8	-	Freewheeling Enable	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB8	-	Freewheeling Enable	LRC	Stop

### Freewheeling Enable

FREEWHEELING\_DISABLED 0x00

FREEWHEELING\_ENABLED 0x01

## 7.94 Freewheeling Start Velocity Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB9	Freewheeling Start Velocity Threshold High	Freewheeling Start Velocity Threshold Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xB9	Freewheeling Start Velocity Threshold High	Freewheeling Start Velocity Threshold Low	LRC	Stop

### Note

Freewheeling Start Velocity Threshold in degree per second multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

### Example

Freewheeling Start Freewheeling Velocity Threshold = 100 → 1000 → Freewheeling Start Velocity Threshold High = 0x03/3d, Freewheeling Start Velocity Threshold Low = 0xE8/232d

## 7.95 Freewheeling Friction

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xBA	Freewheeling Friction High	Freewheeling Friction Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xBA	Freewheeling Friction High	Freewheeling Friction Low	LRC	Stop

### Note

Freewheeling Friction multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.96 Freewheeling Damping

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xBC	Freewheeling Damping High	Freewheeling Damping Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xBC	Freewheeling Damping High	Freewheeling Damping Low	LRC	Stop

### Note

Freewheeling Damping multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.97 Freewheeling Inertia

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xBD	Freewheeling Inertia High	Freewheeling Inertia Low	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xBD	Freewheeling Inertia High	Freewheeling Inertia Low	LRC	Stop

### Note

Freewheeling Inertia multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.98 Single Tick Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC0	-	Single Tick Enable	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC0	-	Single Tick Enable	LRC	Stop

### Single Tick Enable

SINGLE\_TICK\_DISABLED    0x00  
SINGLE\_TICK\_ENABLED     0x01

## 7.99 Single Tick Active Direction

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC1	-	Single Tick Active Direction	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC1	-	Single Tick Active Direction	LRC	Stop

## Single Tick Active Direction

SINGLE\_TICK\_ACTIVE\_DIRECTION\_BOTH 0x00  
SINGLE\_TICK\_ACTIVE\_DIRECTION\_CW 0x01  
SINGLE\_TICK\_ACTIVE\_DIRECTION\_CCW 0x02

## 7.100 Single Tick Angle

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC2	Single Tick Angle High	Single Tick Angle Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC2	Single Tick Angle High	Single Tick Angle Low	LRC	Stop

## Note

Single Tick Angle in degrees multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

## Example

Single Tick Angle =  $123.4^{\circ} \rightarrow 6170 \rightarrow$  Single Tick Angle CW High = 0x18/24d, Single Tick Angle CW Low = 0x1A/26d

## 7.101 Single Tick Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC3	Single Tick Current High	Single Tick Current Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC3	Single Tick Current High	Single Tick Current Low	LRC	Stop

## Note

Single Tick Current in Ampere multiplied by 1000, converted as 16 bit signed integer and split into high and low byte.

## Example

Single Tick Current = 0.1 A  $\rightarrow 100 \rightarrow$  Single Tick Current High = 0x00/0d, Single Tick Current Low = 0x64/100d

## 7.102 Single Tick Mode

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC4	-	Single Tick Mode	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC4	-	Single Tick Mode	LRC	Stop

## Single Tick Mode

SINGLE\_TICK\_MODE\_1 0x00  
SINGLE\_TICK\_MODE\_2 0x01  
SINGLE\_TICK\_MODE\_3 0x02

## Note

Only one mode can be set at a time. Tick mode 4 is not available as Single Tick.

## 7.103 Single Tick Duration Min

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC5	Single Tick Duration Min High	Single Tick Duration Min Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC5	Single Tick Duration Min High	Single Tick Duration Min Low	LRC	Stop

## Note

Single Tick Duration Minimum in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte.

## 7.104 Single Tick Duration Max

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC6	Single Tick Duration Max High	Single Tick Duration Max Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC6	Single Tick Duration Max High	Single Tick Duration Max Low	LRC	Stop

## Note

Single Tick Duration Maximum in seconds multiplied by 1000, converted to 16 bit unsigned integer and split into high and low byte. If Single Tick Duration Max < Single Tick Duration Min then Single Tick Duration Max will be set to Single Tick Duration Min in firmware.

## 7.105 Single Tick Window

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC7	Single Tick Window High	Single Tick Window Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC7	Single Tick Window High	Single Tick Window Low	LRC	Stop

## Note

Single Tick Window in degrees multiplied by 10, converted to 16 bit unsigned integer and split into high and low byte.

## Example

Single Tick Window =  $5^{\circ} \rightarrow 50 \rightarrow$  Single Tick Window High = 0x00/0d, Single Tick Window Low = 0x32/50d

## 7.106 Single Tick Velocity Factor

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC8	Single Tick Velocity Factor High	Single Tick Velocity Factor Low	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xC8	Single Tick Velocity Factor High	Single Tick Velocity Factor Low	LRC	Stop

## Note

Single Tick Velocity Factor multiplied by 100, converted to 16 bit signed integer and split into high and low byte. Set Single Tick Velocity Factor to 0 to disable velocity compensation.

## Example

Single Tick Velocity Factor = 0.5  $\rightarrow 50 \rightarrow$  Single Tick Velocity Factor High = 0x00/0d, Single Tick Velocity Factor Low = 0x32/50d

## 7.107 Haptics Generator Enable

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xCA	-	Haptics Generator Enable	LRC	Stop

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xCA	-	Haptics Generator Enable	LRC	Stop

### Haptics Generator Enable

HAPTICS\_GENERATOR\_DISABLED 0x00  
HAPTICS\_GENERATOR\_ENABLED 0x01

## 7.108 Loopback

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xFF	-	-	LRC	Stop



**Reply**

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xFF	-	-	LRC	Stop

**Note**

The reply packet contains the same data as the received packet.

## 8 Report Register (readonly)

### 8.1 Encoder Angle

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE0	LRC	Stop

Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE0	Encoder Angle High	Encoder Angle Low	LRC	Stop

### 8.2 Encoder Velocity

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE1	LRC	Stop

Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE1	Encoder Velocity High	Encoder Velocity Low	LRC	Stop

Note

Encoder Velocity in degree per second multiplied by 100, converted to 16 bit signed integer and split into high and low byte.

Example

Encoder Velocity =  $0.5^{\circ}/s \rightarrow 50 \rightarrow ->$  Encoder Velocity High = 0x00/0d, Encoder Velocity Low = 0x32/50d

### 8.3 Push Pull Value

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE2	LRC	Stop

Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE2	Push Pull Value High	Push Pull Value Low	LRC	Stop

Note

Push Pull Value is normalized to 1 and multiplied by 10000, converted to 16 bit integer and split into high and low byte.

### 8.4 Tick Index

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE3	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE3	Tick Index High	Tick Index Low	LRC	Stop

### Note

Tick Index multiplied by 1, converted to 16 bit integer and split into high and low byte.

## 8.5 Push Pull State

### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE4	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE4	-	Push Pull State	LRC	Stop

### Push Pull State

PUSH_PULL_STATE_NONE	0x00
PUSH_PULL_STATE_PUSH	0x01
PUSH_PULL_STATE_PULL	0x02
PUSH_PULL_STATE_UNKNOWN	0x03

## 8.6 Encoder Multi Turn Count

### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE5	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE5	Encoder Multi Turn Count High	Encoder Multi Turn Count Low	LRC	Stop

### Note

Encoder Multi Turn Count multiplied by 1, converted to 16 bit integer and split into high and low byte.

## 8.7 Encoder Temperature

### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE6	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE6	Encoder Temperature High	Encoder Temperature Low	LRC	Stop

### Note

Encoder Temperature in Degree Celsius multiplied by 100, converted to 16 bit integer and split into high and low byte.

## 8.8 Coil Current

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE7	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE7	Coil Current High	Coil Current Low	LRC	Stop

### Note

Coil Current in Ampere multiplied by 1000, converted as 16 bit signed integer and split into high and low byte.

## 8.9 Total Turn Counter1

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE8	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE8	Total Turn Counter1 High	Total Turn Counter1 Low	LRC	Stop

### Note

Total Turn Counter1 multiplied by 1, converted as 16 bit unsigned integer and split into high and low byte.

## 8.10 Total Turn Counter2

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xE9	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xE9	Total Turn Counter2 High	Total Turn Counter2 Low	LRC	Stop

### Note

Total Turn Counter2 multiplied by 1, converted as 16 bit unsigned integer and split into high and low byte.

## 8.11 Device Error Status

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xEA	LRC	Stop

Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xEA	Device Status High	Device Status Low	LRC	Stop

Device Status

DEVICE_ERROR_STATUS_NO_ERROR	0x0000
DEVICE_ERROR_STATUS_COIL_RESISTANCE_TOO_LOW	0x0001
DEVICE_ERROR_STATUS_COIL_RESISTANCE_TOO_HIGH	0x0002
DEVICE_ERROR_STATUS_COIL_DRIVER_MAXIMUM_CURRENT_EXCEEDED	0x0004
DEVICE_ERROR_STATUS_ENCODER_COMMUNICATION_ERROR	0x0008
DEVICE_ERROR_STATUS_EEPROM_COMMUNICATION_ERROR	0x0010
DEVICE_ERROR_STATUS_COIL_DRIVER_ERROR	0x0020
DEVICE_ERROR_STATUS_COIL_TEMPERATURE_MAXIMUM_EXCEEDED	0x0040
DEVICE_ERROR_STATUS_EEPROM_CRC_INVALID	0x0080
DEVICE_ERROR_STATUS_EEPROM_INIT_ERROR	0x0100
DEVICE_ERROR_STATUS_ENCODER_DRIVER_INIT_ERROR	0x0200
DEVICE_ERROR_STATUS_ENCODER_DATA_INVALID_ERROR	0x0400
DEVICE_ERROR_STATUS_COIL_CURRENT_ERROR	0x0800

## 8.12 Encoder Calibration Status

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xEB	LRC	Stop

Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xEB	-	Encoder Calibration Status	LRC	Stop

Encoder Calibration Status

CALIBRATION_STATUS_NOT_SUPPORTED	0x00
CALIBRATION_STATUS_UNCALIBRATED	0x01
CALIBRATION_STATUS_CALIBRATING	0x02
CALIBRATION_STATUS_CALIBRATED	0x03
CALIBRATION_STATUS_FAILED	0x04

## 8.13 Coil Resistance

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xEC	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xEC	Coil Resistance High	Coil Resistance Low	LRC	Stop

### Note

Coil Resistance in Ohm multiplied by 100, converted as 16 bit unsigned integer and split into high and low byte.

## 8.14 Push Calibration Status

### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xED	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xED	-	Push Calibration Status	LRC	Stop

### Push Calibration Status

CALIBRATION_STATUS_NOT_SUPPORTED	0x00
CALIBRATION_STATUS_UNCALIBRATED	0x01
CALIBRATION_STATUS_CALIBRATING	0x02
CALIBRATION_STATUS_CALIBRATED	0x03
CALIBRATION_STATUS_FAILED	0x04

## 8.15 Pull Calibration Status

### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xEE	LRC	Stop

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xEE	-	Pull Calibration Status	LRC	Stop

### Pull Calibration Status

CALIBRATION_STATUS_NOT_SUPPORTED	0x00
CALIBRATION_STATUS_UNCALIBRATED	0x01
CALIBRATION_STATUS_CALIBRATING	0x02
CALIBRATION_STATUS_CALIBRATED	0x03
CALIBRATION_STATUS_FAILED	0x04

## 8.16 Device Connection State

### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xEF	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xEF	-	Device Connection State	LRC	Stop

#### Device Connection State

DEVICE\_CONNECTION\_STATE\_DISCONNECTED 0x00

DEVICE\_CONNECTION\_STATE\_CONNECTED 0x01

## 8.17 Encoder Angle Min

#### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xF2	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xF2	Encoder Angle Min High	Encoder Angle Min Low	LRC	Stop

#### Note

Encoder Angle Min in degree multiplied by 100, converted as 16 bit signed integer and split into high and low byte.

## 8.18 Encoder Angle Max

#### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xF3	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xF3	Encoder Angle Max High	Encoder Angle Max Low	LRC	Stop

#### Note

Encoder Angle Max in degree multiplied by 100, converted as 16 bit signed integer and split into high and low byte.

## 8.19 Supply Voltage

#### Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xF4	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xF4	Supply Voltage High	Supply Voltage Low	LRC	Stop

#### Note

Supply Voltage in Volt multiplied by 1000, converted as 16 bit unsigned integer and split into high and low byte.

## 8.20 Degauss Status

Command (optional)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0x03	-	0xF5	LRC	Stop

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Start	0xF5	-	Supply Voltage Low	LRC	Stop

#### Degauss Status

```
DEGAUSS_STATUS_NOT_DEGAUSSING  0x00
DEGAUSS_STATUS_DEGAUSSING       0x01
```



## 9 Haptics Generator General Packet Structure

COM Port Settings: 115.200 Baud, 8N1

Packet length = 5...1029 Byte

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte LEN+3	Byte LEN+4
TYPE_LOW	TYPE_HIGH	LEN_LOW	LEN_HIGH	DATA_LSB	DATA	DATA_MSB	LRC

### Packet Byte Description

Name	Description	Value
TYPE_HIGH	Type of command or reply	See below
TYPE_LOW	Type of command or reply	See below
LEN_HIGH	Length of Data in Byte	
LEN_LOW	Length of Data in Byte	
DATA_MSB	Most Significant Byte of Data	See below
DATA_LSB	Least Significant Byte of Data	See below
LRC	Longitudinal Redundancy Check	Byte 0 ^ Byte 1 ^ ... ^ Byte LEN+3

### Note

Command and reply packets use data in little endian format.

## 10 HapticsGenerator Commands

### 10.1 Write LUT Angle Current CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12..15	..			m
TYPE_LOW 0xA0	TYPE_HIGH 0x00	LEN_LOW 4 + (8 * n)	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 X_0	DATA8..11 Y_0	..	X_n	Y_n	LRC

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA0	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

#### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

#### Note

X\_0, Y\_0 to X\_n, Y\_n are encoded as 4 byte float. n\_max = 6 per packet.  
The offset is counted the same as n. e.g. If 4 data points (X0-Y0 to X3-Y3) are set, the next offset is 4.

### 10.2 Write LUT Angle Current CCW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12..15	..			m
TYPE_LOW 0xA1	TYPE_HIGH 0x00	LEN_LOW 4 + (8 * n)	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 X_0	DATA8..11 Y_0	..	X_n	Y_n	LRC

#### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA1	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

#### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

#### Note

X\_0, Y\_0 to X\_n, Y\_n are encoded as 4 byte float. n\_max = 6 per packet.  
The offset is counted the same as n. e.g. If 4 data points (X0-Y0 to X3-Y3) are set, the next offset is 4.

### 10.3 Write LUT Angle Current Size CW

#### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TYPE_LOW 0xA2	TYPE_HIGH 0x00	LEN_LOW 0x02	LEN_HIGH 0x00	DATA0 LUT_SIZE_LSB	DATA1 LUT_SIZE_MSB	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA2	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

### Note

LUT size is the number of data points used of the LUT Angle Current CW.

## 10.4 Write LUT Angle Current Size CCW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TYPE_LOW 0xA3	TYPE_HIGH 0x00	LEN_LOW 0x02	LEN_HIGH 0x00	DATA0 LUT_SIZE_LSB	DATA1 LUT_SIZE_MSB	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA3	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

### Note

LUT size is the number of data points used of the LUT Angle Current CCW.

## 10.5 Write Alternate LUT Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 6
TYPE_LOW 0xA5	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA0 ALTERNATE	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA5	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

### Alternate

ALTERNATE\_DISABLED    0x00  
ALTERNATE\_ENABLED    0x01

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

### Note

Returns always STATUS\_OK

## 10.6 Write LUT Velocity Compensation Type

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA6	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA VEL.COMP.TYPE	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA6	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA STATUS	LRC

### Velocity Compensation Type

VELOCITY\_COMPENSATION\_TYPE\_MULTIPLICATIVE    0x00  
VELOCITY\_COMPENSATION\_TYPE\_ADDITIVE         0x01

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

## 10.7 Write Tick Index Count Angle CW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8
TYPE_LOW 0xA8	TYPE_HIGH 0x00	LEN_LOW 0x04	LEN_HIGH 0x00	DATA0..3 Angle	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA8	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

## 10.8 Write Tick Index Count Angle CCW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8
TYPE_LOW 0xAA	TYPE_HIGH 0x00	LEN_LOW 0x04	LEN_HIGH 0x00	DATA0..3 Angle	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xAA	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

## 10.9 Write LUT Velocity Factor

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12..15	..			m
TYPE_LOW 0xB0	TYPE_HIGH 0x00	LEN_LOW 4 + (8 * n)	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 X_0	DATA8..11 Y_0	..	X_n	Y_n	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xB0	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

### Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

### Note

X\_0, Y\_0 to X\_n, Y\_n are encoded as 4 byte float. n\_max = 6 per packet.  
The offset is counted the same as n. e.g. If 4 data points (X0-Y0 to X3-Y3) are set, the next offset is 4.

## 10.10 Write LUT Velocity Factor Size

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TYPE_LOW 0xB1	TYPE_HIGH 0x00	LEN_LOW 0x02	LEN_HIGH 0x00	DATA0 LUT_SIZE_LSB	DATA1 LUT_SIZE_MSB	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xB1	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

## Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

## 10.11 Write LUT Freewheeling Velocity Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8
TYPE_LOW 0xC0	TYPE_HIGH 0x00	LEN_LOW 0x04	LEN_HIGH 0x00	DATA0..3 Angle	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xC0	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

## Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

## 10.12 Write LUT Force Freewheeling

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xC1	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA FORCE	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xC1	TYPE_HIGH 0x00	LEN_LOW 0x01	LEN_HIGH 0x00	DATA Status	LRC

## Force

FORCE\_FREEWHEELING\_DISABLED    0x00  
FORCE\_FREEWHEELING\_ENABLED    0x01

## Status

STATUS\_OK        0x00  
STATUS\_ERROR    0x01

## 10.13 Read LUT Angle Current CW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12
TYPE_LOW 0xA0	TYPE_HIGH 0x01	LEN_LOW 0x08	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 Read Length	LRC

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12..15	..			m
TYPE_LOW 0xA0	TYPE_HIGH 0x01	LEN_LOW 4 + (8 * n)	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 X_0	DATA8..11 Y_0	.. ..	X_n	Y_n	LRC

## Note

X\_0, Y\_0 to X\_n, Y\_n are encoded as 4 byte float. n\_max = 6 per packet.  
The offset is counted the same as n. e.g. If 4 data points (X0-Y0 to X3-Y3) are read, the next offset is 4.

## 10.14 Read LUT Angle Current CCW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12
TYPE_LOW 0xA1	TYPE_HIGH 0x01	LEN_LOW 0x08	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 Read Length	LRC

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12..15	..			m
TYPE_LOW 0xA1	TYPE_HIGH 0x01	LEN_LOW 4 + (8 * n)	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 X_0	DATA8..11 Y_0	.. ..	X_n	Y_n	LRC

## Note

X\_0, Y\_0 to X\_n, Y\_n are encoded as 4 byte float. n\_max = 6 per packet.  
The offset is counted the same as n. e.g. If 4 data points (X0-Y0 to X3-Y3) are read, the next offset is 4.

## 10.15 Read LUT Angle Current Size CW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xA2	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TYPE_LOW 0xA2	TYPE_HIGH 0x01	LEN_LOW 0x02	LEN_HIGH 0x00	DATA0 LUT_SIZE_LSB	DATA1 LUT_SIZE_MSB	LRC

## 10.16 Read LUT Angle Current Size CCW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xA3	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

## Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TYPE_LOW 0xA3	TYPE_HIGH 0x01	LEN_LOW 0x02	LEN_HIGH 0x00	DATA0 LUT_SIZE_LSB	DATA1 LUT_SIZE_MSB	LRC

## 10.17 Read Alternate LUT Current

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xA5	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA5	TYPE_HIGH 0x01	LEN_LOW 0x01	LEN_HIGH 0x00	DATA ALTERNATE	LRC

### Alternate

ALTERNATE\_DISABLED 0x00  
ALTERNATE\_ENABLED 0x01

## 10.18 Read LUT Velocity Compensation Type

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xA6	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xA6	TYPE_HIGH 0x01	LEN_LOW 0x01	LEN_HIGH 0x00	DATA VEL.COMP.TYPE	LRC

### Velocity Compensation Type

VELOCITY\_COMPENSATION\_TYPE\_MULTIPLICATIVE 0x00  
VELOCITY\_COMPENSATION\_TYPE\_ADDITIVE 0x01

## 10.19 Read Tick Index Count Angle CW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xA8	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8
TYPE_LOW 0xA8	TYPE_HIGH 0x01	LEN_LOW 0x04	LEN_HIGH 0x00	DATA0..3 Angle	LRC



## 10.20 Read Tick Index Count Angle CCW

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xAA	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8
TYPE_LOW 0xAA	TYPE_HIGH 0x01	LEN_LOW 0x04	LEN_HIGH 0x00	DATA0..3 Angle	LRC

## 10.21 Read LUT Velocity Factor

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12
TYPE_LOW 0xB0	TYPE_HIGH 0x01	LEN_LOW 0x08	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 Read Length	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8..11	Byte 12..15	..			m
TYPE_LOW 0xB0	TYPE_HIGH 0x01	LEN_LOW 4 + (8 * n)	LEN_HIGH 0x00	DATA0..3 Offset	DATA4..7 X_0	DATA8..11 Y_0	..	X_n	Y_n	LRC

### Note

X\_0, Y\_0 to X\_n, Y\_n are encoded as 4 byte float. n\_max = 6 per packet.

The offset is counted the same as n. e.g. If 4 data points (X0-Y0 to X3-Y3) are read, the next offset is 4.

## 10.22 Read LUT Velocity Factor Size

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xB1	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
TYPE_LOW 0xB1	TYPE_HIGH 0x01	LEN_LOW 0x02	LEN_HIGH 0x00	DATA0 LUT_SIZE_LSB	DATA1 LUT_SIZE_MSB	LRC

## 10.23 Read Freewheeling Velocity Threshold

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xC0	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4..7	Byte 8
TYPE_LOW 0xC0	TYPE_HIGH 0x01	LEN_LOW 0x04	LEN_HIGH 0x00	DATA0..3 Threshold	LRC

## 10.24 Read Force Freewheeling

### Command

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
TYPE_LOW 0xC1	TYPE_HIGH 0x01	LEN_LOW 0x00	LEN_HIGH 0x00	LRC

### Reply

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
TYPE_LOW 0xC1	TYPE_HIGH 0x01	LEN_LOW 0x01	LEN_HIGH 0x00	DATA FORCE	LRC

### Force

FORCE\_FREEWHEELING\_DISABLED    0x00  
 FORCE\_FREEWHEELING\_ENABLED    0x01

## You still didn't find the answer to your question?

---

If you need additional assistance or have recommendations for changes, please don't hesitate to get in touch with us:

support@xeeltech.com  
+43 5552 93081-0  
www.xeeltech.com

Or visit [www.xeeltech.com/hapticore-support](http://www.xeeltech.com/hapticore-support) where you can find helpful FAQs and video tutorials.

Here you will also find our other HAPTICORE software solutions, allowing you to create even more personalized haptic feedback modes.

 **HAPTICORE**  
by  **XEELTECH**