

## ScienceMode

---

### RehaMove3

#### Description and Protocol

Version	3.2.4
Date	2018-04-21
Author	Bjorn Kuberski
Email	kuberski@hasomed.de

## Table of Contents

1	ScienceMode General Description .....	3
1.1	Introduction.....	3
1.2	RehaMove3 .....	3
1.3	Electrode Error Detection.....	3
1.4	Stimulation Modes .....	3
1.5	Stimulation Channels.....	3
2	Protocol General.....	4
2.1	Serial Connection.....	4
2.2	Packet Structure .....	4
2.3	Commands and Responses.....	5
3	General Commands .....	6
3.1	Overview.....	6
3.2	Commands – Direction RehaMove3.....	6
3.3	Responses – Direction External Device .....	7
4	Low-Level Commands.....	9
4.1	Overview.....	9
4.2	Commands – Direction RehaMove3.....	10
4.3	Responses – Direction External Device .....	11
5	Protocol Mid-Level Commands .....	12
5.1	Overview.....	12
5.2	Commands – Direction RehaMove3.....	12
5.3	Responses – Direction External Device .....	13
6	Result and Error Values .....	14
7	Examples.....	15
7.1	Low-Level.....	15
7.2	Mid-Level.....	15

# 1 ScienceMode General Description

## 1.1 Introduction

ScienceMode is a communication protocol between an external device (e.g. a PC) and the electrical stimulator RehaMove3, which is produced by HASOMED (see <http://www.rehamove.com>). Using the ScienceMode commands, the external device can control the stimulation to execute complex stimulation patterns for a wide range of research applications.

## 1.2 RehaMove3

RehaMove3 is a 4-channel electrical stimulation unit with the following specifications:

- Up to 4 channels (one current source)
- Current  $\pm 130$  mA
- Pulse width: 20 – 16000  $\mu$ s
- Frequency: 1 – 500 Hz
- Individual pulse forms using up to 16 points
- Low latency for real time applications
- Electrode error detection
- Galvanic isolated FTDI USB connection
- Rechargeable battery
- Weight: 280 g
- Size: 50 x 73 x 32 mm

## 1.3 Electrode Error Detection

The stimulator provides a skin resistance check for safety reasons. The resistance is determined by analyzing the effect of a small test impulse which is sent before each stimulation pulse. If the resistance is not inside normal ranges the stimulation pulse will not be executed.

## 1.4 Stimulation Modes

There are two modes for the stimulation: low and mid-level.

In the low-level mode every single stimulation pulse needs to be sent. Therefore the stimulation frequency is generated by the external device. This allows irregular frequencies and a fine tuned stimulation execution.

In mid-level mode the pulse frequency is generated by the stimulation device. Each channel can have its own frequency, which also can be updated regularly.

## 1.5 Stimulation Channels

The 4 stimulation channels are coded in the following way:

- Red – Channel number 0
- Blue – Channel number 1
- Black – Channel number 2
- White – Channel number 3

## 2 Protocol General

### 2.1 Serial Connection

Connect an external device and the RehaMove3 (micro-USB) using the USB connection cable. RehaMove3 uses an integrated FTDI-Chip to generate a virtual serial port interface on the external device. Therefore the FTDI driver needs to be installed.

The needed Windows driver can be downloaded from [www.ftdichip.com](http://www.ftdichip.com). The COM port for the virtual serial port can be found in the device manager after its installation. It is important to adjust the standard driver settings for optimized timings. In the Device Manager open the USB serial device window and continue to open the properties window. Navigate to the port settings and then open the advanced window. Set the latency timer to 1 ms.

Linux and MacOS include a kernel FTDI-Module that can be used. The non-standard baud rate can be set in Linux using the functions `cfsetospeed()` and `cfsetispeed()`. MacOS users can use the function `ioctl(fd, IOSSIOSPEED, speed)`.

The serial settings for the RehaMove3 ScienceMode protocol are listed in Table 1.

Parameter	Value
Baud rate	3 000 000
Data bits	8
Stop bits	2
Parity	None
Flow control RTS/CTS	Yes

Table 1: Serial settings

### 2.2 Packet Structure

The protocol packet structure is shown in Table 2. Every packet starts with the start byte and ends with the stop byte (see constants in Table 3).

The protocol uses byte stuffing. Every constant from Table 3, except the stuffing key, is escaped with the stuffing byte. The value is then XORed with the stuffing key, e.g. if the command data has a byte which is identical to the start byte value, this start byte (0xF0) is stuffed to 0x81, 0xA5.

The packet length (2 Byte) is the full length of the packet including the start and the stop byte. It is always stuffed and therefore uses 4 Bytes. The maximum packet length including stuffing is 1200 Bytes.

The check sum is the CRC-CCITT 2 Byte (16 Bit) and is generated from the stuffed packet data (see Table 2). It is always stuffed and therefore uses 4 Bytes.

The packet number and command number combined use 2 Bytes. The packet number ranges from 0 to 63 and is echoed by the stimulator response. It can be used for debugging purposes. The command is represented by a number for the different commands, responses and information packets, whereas the command data is reserved for the parameter of the command.

The ScienceMode protocol uses little endian byte order.

Start byte	Packet length	Check sum	Packet number (6 Bit) Command number (10 Bit)	Command data	Stop byte
1 Byte	4 Byte	4 Byte	2 Byte	n-Byte	1 Byte
			<- Packet data ->		
			<- Stuffing ->		
			<- Check sum ->		
<- Packet length ->					
<- Packet (Header, Data, Footer) ->					

Table 2: Packet structure

Constants	Hex	Binary
Start byte	0xF0	11110000
Stop byte	0x0F	00001111
Stuffing byte	0x81	10000001
Stuffing key	0x55	01010101

Table 3: Byte constants

### 2.3 Commands and Responses

The stimulator functions can be controlled by sending a request command. Most of these request commands have a corresponding stimulator response (ending with ack), contained the result, the requested data and sometimes additional information. The commands, its responses and their parameters are described in the next sections.

There are three layers: general, low and mid-level. The general layer contains commands for requesting basic information like battery state, device id or general status. The low and mid-level layers contain the commands for the corresponding stimulation modes.

### 3 General Commands

#### 3.1 Overview

Command number	Command	Description	Direction
050	Get_version_main	Requests the Firmware and ScienceMode library version	RehaMove3
051	Get_version_main_ack	Response containing the Firmware and ScienceMode library version	Ext. device
052	Get_device_id	Requests the device ID	RehaMove3
053	Get_device_id_ack	Response containing the device ID	Ext. device
054	Get_battery_status	Requests the rechargeable battery status	RehaMove3
055	Get_battery_status_ack	Response containing the rechargeable battery status	Ext. device
058	Reset	Requests a restart of the stimulator	RehaMove3
059	Reset_ack	Acknowledge of the reset command (currently not send)	Ext. device
062	Get_stim_status	Requests the status of the stimulation	RehaMove3
063	Get_stim_status_ack	Response containing the status of the stimulation	Ext. device
066	General_error	Send when a general error occurs	Ext. device
067	Unknown_cmd	Send when the command number cannot be processed	Ext. device

#### 3.2 Commands – Direction RehaMove3

Command	Details
Get_version_main	Packet length: 12 No parameter
Get_device_id	Packet length: 12 No parameter
Get_battery_status	Packet length: 12 No parameter
Reset	Packet length: 12 No parameter
Get_stim_status	Packet length: 12 No parameter

### 3.3 Responses – Direction External Device

Command	Description																														
Get_version_main_ack	Packet length: 19																														
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Byte</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="4">Result</td> <td rowspan="4">1 Byte</td> <td> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table> </td> </tr> <tr> <td>FW-Major</td> <td>1 Byte</td> <td>Firmware-Version Major</td> </tr> <tr> <td>FW-Minor</td> <td>1 Byte</td> <td>Firmware-Version Minor</td> </tr> <tr> <td>FW-Revision</td> <td>1 Byte</td> <td>Firmware-Version Revision</td> </tr> <tr> <td>SMPT-Major</td> <td>1 Byte</td> <td>ScienceMode-Version Major</td> </tr> <tr> <td>SMPT-Minor</td> <td>1 Byte</td> <td>ScienceMode-Version Minor</td> </tr> <tr> <td>SMPT-Revision</td> <td>1 Byte</td> <td>ScienceMode-Version Revision</td> </tr> </tbody> </table>	Parameter	Byte	Description	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value	Description	0	Successful	1	Transfer error	FW-Major	1 Byte	Firmware-Version Major	FW-Minor	1 Byte	Firmware-Version Minor	FW-Revision	1 Byte	Firmware-Version Revision	SMPT-Major	1 Byte	ScienceMode-Version Major	SMPT-Minor	1 Byte	ScienceMode-Version Minor	SMPT-Revision	1 Byte	ScienceMode-Version Revision
	Parameter	Byte	Description																												
	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value			Description	0	Successful	1	Transfer error																				
			Value	Description																											
			0	Successful																											
			1	Transfer error																											
	FW-Major	1 Byte	Firmware-Version Major																												
	FW-Minor	1 Byte	Firmware-Version Minor																												
FW-Revision	1 Byte	Firmware-Version Revision																													
SMPT-Major	1 Byte	ScienceMode-Version Major																													
SMPT-Minor	1 Byte	ScienceMode-Version Minor																													
SMPT-Revision	1 Byte	ScienceMode-Version Revision																													
Get_device_id_ack	Packet length: 23																														
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Byte</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Result</td> <td rowspan="3">1 Byte</td> <td> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table> </td> </tr> <tr> <td>Device-ID</td> <td>10 Byte</td> <td>Device-ID coded as chars</td> </tr> </tbody> </table>	Parameter	Byte	Description	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value	Description	0	Successful	1	Transfer error	Device-ID	10 Byte	Device-ID coded as chars															
	Parameter	Byte	Description																												
	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value			Description	0	Successful	1	Transfer error																				
Value			Description																												
0			Successful																												
1	Transfer error																														
Device-ID	10 Byte	Device-ID coded as chars																													
Get_battery_status_ack	Packet length: 16																														
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Byte</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Result</td> <td rowspan="3">1 Byte</td> <td> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table> </td> </tr> <tr> <td>Rechargeable Battery level</td> <td>1 Byte</td> <td>[0 .. 1 .. 100] %</td> </tr> <tr> <td>Rechargeable Battery voltage</td> <td>2 Byte</td> <td>[0 .. 1 .. 65535] mV</td> </tr> </tbody> </table>	Parameter	Byte	Description	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value	Description	0	Successful	1	Transfer error	Rechargeable Battery level	1 Byte	[0 .. 1 .. 100] %	Rechargeable Battery voltage	2 Byte	[0 .. 1 .. 65535] mV												
	Parameter	Byte	Description																												
	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value			Description	0	Successful	1	Transfer error																				
			Value	Description																											
0			Successful																												
1	Transfer error																														
Rechargeable Battery level	1 Byte	[0 .. 1 .. 100] %																													
Rechargeable Battery voltage	2 Byte	[0 .. 1 .. 65535] mV																													
Reset_ack	Packet length: 13 Byte																														
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Byte</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td rowspan="3">Result</td> <td rowspan="3">1 Byte</td> <td> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	Parameter	Byte	Description	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value	Description	0	Successful	1	Transfer error																		
	Parameter	Byte	Description																												
Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value	Description			0	Successful	1	Transfer error																					
		Value	Description																												
		0	Successful																												
1	Transfer error																														

Command	Description															
Get_stim_status_ack	Packet Length: 15 Byte															
	Parameter	Byte	Description													
	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Successful</td> </tr> <tr> <td>1</td> <td>Transfer error</td> </tr> </tbody> </table>	Value	Description	0	Successful	1	Transfer error							
			Value	Description												
0			Successful													
1	Transfer error															
Stim-Status	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No Level initialized</td> </tr> <tr> <td>1</td> <td>Low-Level initialized</td> </tr> <tr> <td>2</td> <td>Mid-Level initialized</td> </tr> <tr> <td>3</td> <td>Mid-Level running</td> </tr> </tbody> </table>	Value	Description	0	No Level initialized	1	Low-Level initialized	2	Mid-Level initialized	3	Mid-Level running				
		Value	Description													
		0	No Level initialized													
		1	Low-Level initialized													
2	Mid-Level initialized															
3	Mid-Level running															
High-Voltage-Level	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Off</td> </tr> <tr> <td>2</td> <td>30 V</td> </tr> <tr> <td>3</td> <td>60 V</td> </tr> <tr> <td>4</td> <td>90 V</td> </tr> <tr> <td>5</td> <td>120 V</td> </tr> <tr> <td>6</td> <td>150 V</td> </tr> </tbody> </table>	Value	Description	1	Off	2	30 V	3	60 V	4	90 V	5	120 V	6	150 V
		Value	Description													
		1	Off													
		2	30 V													
		3	60 V													
		4	90 V													
5	120 V															
6	150 V															
General_error	Packet Length: 13 Byte															
	Parameter	Byte	Description													
	Result	1 Byte	see section Error Values													
Unknown_cmd	Packet Length: 13 Byte															
	Parameter	Byte	Description													
	Result	1 Byte	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>11</td> <td>Unknown command</td> </tr> </tbody> </table>	Value	Description	11	Unknown command									
Value	Description															
11	Unknown command															



## 4 Low-Level Commands

### 4.1 Overview

Command number	Command	Description	Direction
000	Ll_init	Requests the initialization of the low level mode; Switches on the high voltage	RehaMove3
001	Ll_init_ack	Response for Ll_init (response requires around 40 ms)	Ext. device
002	Ll_channel_config	Requests the execution of a stimulation pulse; The stimulator can buffer up to 10 commands in its internal buffer. This is useful for the execution of stimulation pulses with a high frequency.	RehaMove3
003	Ll_channel_config_ack	Response for Ll_channel_config; The command is send after the stimulation has been executed. It contains information about possible errors.	Ext. device
004	Ll_stop	Requests the stop of low-level mode; Switches off the high voltage	RehaMove3
005	Ll_stop_ack	Response to Ll_stop (response requires around 40 ms)	Ext. device

#### 4.2 Commands – Direction RehaMove3

Command	Details				
Ll_init	Packet length: 13 Byte				
	<b>Parameter</b>	<b>Bit</b>	<b>Bytes</b>	<b>Description</b>	
	Reserved	4 Bit	1 Byte	Should be set to 0	
	High Voltage	3 Bit		0 – Standard (150 V) 1 – Off            4 – 90 V 2 – 30 V            5 – 120 V 3 – 60 V            6 – 150 V	
Reserved	1 Bit	Should be set to 0			
Ll_channel_config	Packet length: 17 – 75 Byte				
	<b>Parameter</b>	<b>Bit</b>	<b>Byte</b>	<b>Description</b>	
	Execution Stimulation	1 Bit	1 Byte	0 – No Stimulation 1 – Stimulation will be executed	
	Channel selection	2 Bit		Channel selection (0 – 3)	
	Reserved	1 Bit		Should be set to 0	
	Number of points	4 Bit		Number of points (1 – 16; Values 0 – 15) 0 -> 1 Point 1 -> 2 Points, etc.	
For each point (maximum 16 points)					
	Point	32 Bit	4 Byte	See point definition	
Ll_channel_config (Point definition)	<b>Parameter</b>	<b>Bit</b>	<b>Byte</b>	<b>Range</b>	<b>Values</b>
	Duration	12 Bit	4 Byte	[0 .. 1 .. 4095] µs	4096
	Current	10 Bit		[-150 .. -149,5 .. 150] mA transfer function: $f(x) = 2x + 300$ 0 -> -150 mA 1 -> -149,5 mA 2 -> -149 mA, etc.	600 (1024)
	Reserved	10 Bit			
Ll_stop	Packet length: 12 Byte No Parameter				

### 4.3 Responses – Direction External Device

Command	Details			
LI_init_ack	Packet length: 13 Byte			
	<b>Parameter</b>	<b>Bytes</b>	<b>Description</b>	
	Result	1 Byte	<b>Value</b>	<b>Description</b>
			00	Successful
			01	Transfer error
			02	Parameter error
			04	Timeout Stimulation
LI_channel_config_ack	Packet length: 14 Byte			
	<b>Parameter</b>	<b>Byte</b>	<b>Description</b>	
	Result	1 Byte	<b>Value</b>	<b>Description</b>
			00	Successful
			01	Transfer error
			02	Parameter error
			04	Timeout Stimulation
		07	Stimulation not initialized	
		10	Electrode error	
	Electrode error channel	1 Byte	Contains electrode error channel, if result == electrode error	
LI_stop_ack	Packet length: 13 Byte			
	<b>Parameter</b>	<b>Byte</b>	<b>Description</b>	
	Result	1 Byte	<b>Value</b>	<b>Description</b>
			00	Successful
			01	Transfer error

## 5 Protocol Mid-Level Commands

### 5.1 Overview

Befehlsnr.	Command	Description	Direction
030	MI_init	Requests the initialization of the mid-level mode; The high voltage is enabled.	RehaMove3
031	MI_init_ack	Response to MI_init_ack	Ext. device
032	MI_update	Requests the start or the update of the stimulation with the parameters; The stimulation has an automatic timeout of 2 s. To keep the stimulation alive, you need to send the keep-alive-signal (MI_get_current_data) or a MI_update. After a timeout the stimulation can be started again with MI_update.	RehaMove3
033	MI_update_ack	Response to MI_update	Ext. device
034	MI_stop	Requests the stop of the stimulation; The high voltage is switched off.	RehaMove3
035	MI_stop_ack	Response to MI_stop	Ext. device
036	MI_get_current_data	Requests the live data and keep alive signal	RehaMove3
037	MI_get_current_data_ack	Response to MI_get_current_data containing the live data	Ext. device

### 5.2 Commands – Direction RehaMove3

Command	Description		
MI_init	Packets length: 13 Byte		
	Parameter	Bytes	Description
	Reserved value	1 Byte	Should be set to 0

Command	Description											
MI_update	Packet length: 13–301 Byte											
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Bit</th> <th>Bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Channel activation</td> <td>4 Bit</td> <td rowspan="2">1 Byte</td> <td>Bitwise channel activation (channels 0 – 3)</td> </tr> <tr> <td>Unused</td> <td>4 Bit</td> <td>Should be set to 0</td> </tr> </tbody> </table>	Parameter	Bit	Bytes	Description	Channel activation	4 Bit	1 Byte	Bitwise channel activation (channels 0 – 3)	Unused	4 Bit	Should be set to 0
	Parameter	Bit	Bytes	Description								
	Channel activation	4 Bit	1 Byte	Bitwise channel activation (channels 0 – 3)								
	Unused	4 Bit		Should be set to 0								
	For each active channel 0 – 3											
	Number of points	4 Bit	1 Byte	1 – 16; Values 0 – 15 1 Point -> 0 2 Points -> 1, etc.								
	Ramp	4 Bit		0 – 15; Number of linear increasing lower current pulse pattern until the full current is reached; The ramp is executed if the channel is enabled.								
	Period	15 Bit	2 Byte	0,5–16383 ms -> Values 1–32767 Time between two points transfer function $f(x) = 2x$ 0,5 ms -> 1 1 ms -> 2, etc.								
	Reserved	1 Bit		Should be set to 0								
For each point (maximum 16 points)												
	Point definition	4 Byte	see Low-Level point definition									
MI_stop	Packet length: 12 Byte No parameter											
MI_get_curr ent_data	Packet length: 13 Byte											
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Additional Data</td> <td>1 Byte</td> <td>Set value to 0x2 for stimulation data</td> </tr> </tbody> </table>	Parameter	Bytes	Description	Additional Data	1 Byte	Set value to 0x2 for stimulation data					
Parameter	Bytes	Description										
Additional Data	1 Byte	Set value to 0x2 for stimulation data										

### 5.3 Responses – Direction External Device

Command	Description					
MI_init_ack	Packet length: 13 Byte					
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Result</td> <td>1 Byte</td> <td>See section Result and Error values</td> </tr> </tbody> </table>	Parameter	Bytes	Description	Result	1 Byte
Parameter	Bytes	Description				
Result	1 Byte	See section Result and Error values				
MI_update_ack	Packet length: 13 Byte					
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Result</td> <td>1 Byte</td> <td>See section Result and Error values</td> </tr> </tbody> </table>	Parameter	Bytes	Description	Result	1 Byte
Parameter	Bytes	Description				
Result	1 Byte	See section Result and Error values				
MI_stop_ack	Packet length: 13 Byte					
	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Bytes</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Result</td> <td>1 Byte</td> <td>See section Result and Error values</td> </tr> </tbody> </table>	Parameter	Bytes	Description	Result	1 Byte
Parameter	Bytes	Description				
Result	1 Byte	See section Result and Error values				

MI_get_current_data_ack	Packet length: 15-28 Byte			
	Parameter	Bit	Bytes	Description
	Result		1 Byte	See section Result and Error values
	Additional data echo		1 Byte	Set to 0x2
	Unused	3 Bit	1 Byte	
Stimulation status	1 Bit	0 – No Stimulation 1 – Stimulation execution		
Electrode error	4 Bit	Bits 0–3 for each channel 0 – No electrode error 1 – Electrode error		

## 6 Result and Error Values

Value	Result	Description
00	No Error	The command was executed or the execution started.
01	Transfer error	The check sum and/or length included in the packet do not match with the calculated value.
02	Parameter error	Any of the following conditions is true: <ul style="list-style-type: none"> <li>At least one parameter has an invalid value.</li> <li>The packet contains too few parameters.</li> </ul>
04	Stimulation Timeout	Timeout with the stimulation controller
07	Not initialized	The command cannot be executed, because <ul style="list-style-type: none"> <li>The level (mode) is not initialized.</li> <li>Another level is currently initialized.</li> </ul>
10	Electrode error	Electrode error during the execution of the stimulation
11	Unknown Command	The command number is unknown and cannot be processed.

## 7 Examples

### 7.1 Low-Level

Command	Parameter	Packet (Hex)														
LI_init	Packet number: 0	F0 81 55 81 58 81 55 81 55 00 00 00 0F														
LI_channel_config	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Channel</td> <td>red</td> </tr> <tr> <td>Number of points</td> <td>3</td> </tr> <tr> <td>Point 1</td> <td>250 µs, 20 mA</td> </tr> <tr> <td>Point 2</td> <td>100 µs, 0 mA</td> </tr> <tr> <td>Point 3</td> <td>250 µs, -20 mA</td> </tr> <tr> <td>Packet number</td> <td>1</td> </tr> </tbody> </table>	Parameter	Value	Channel	red	Number of points	3	Point 1	250 µs, 20 mA	Point 2	100 µs, 0 mA	Point 3	250 µs, -20 mA	Packet number	1	F0 81 55 81 4E 81 D3 81 AF 04 02 82 81 5A A5 50 00 06 44 B0 00 81 5A A4 10 00 0F
	Parameter	Value														
	Channel	red														
	Number of points	3														
	Point 1	250 µs, 20 mA														
	Point 2	100 µs, 0 mA														
	Point 3	250 µs, -20 mA														
Packet number	1															
LI_stop	Packet number: 2	F0 81 55 81 59 81 9C 81 78 08 04 0F														

### 7.2 Mid-Level

Command	Parameter	Packet (Hex)								
MI_init	Packet number: 0	F0 81 55 81 58 81 75 81 29 00 1E 00 0F								
MI_update	<table border="1"> <thead> <tr> <th>Parameter</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Channel red</td> <td>3 Points biphasic, 200 µs, ± 20 mA, 100 µs break, period 20 ms, ramp 3</td> </tr> <tr> <td>Channel blue</td> <td>3 Points biphasic, 100 µs, ± 10 mA, 100 µs break, period 10 ms, ramp 3</td> </tr> <tr> <td>Packet number</td> <td>1</td> </tr> </tbody> </table>	Parameter	Value	Channel red	3 Points biphasic, 200 µs, ± 20 mA, 100 µs break, period 20 ms, ramp 3	Channel blue	3 Points biphasic, 100 µs, ± 10 mA, 100 µs break, period 10 ms, ramp 3	Packet number	1	F0 81 55 81 7E 81 5D 81 42 04 20 03 23 00 50 0C 85 50 00 06 44 B0 00 0C 84 10 00 23 00 28 06 45 00 00 06 44 B0 00 06 44 60 00 0F
	Parameter	Value								
	Channel red	3 Points biphasic, 200 µs, ± 20 mA, 100 µs break, period 20 ms, ramp 3								
Channel blue	3 Points biphasic, 100 µs, ± 10 mA, 100 µs break, period 10 ms, ramp 3									
Packet number	1									
MI_get_current_data	Packet number: 2	F0 81 55 81 58 81 16 81 94 08 24 02 0F								
MI_stop	Packet number: 3	F0 81 55 81 59 81 14 81 18 0C 22 0F								