ScienceMode overview/comparison (September 2017)

Device	**RehaStim1**	**RehaStim2**	**RehaMove3 Science**
Protocol	ScienceMode1	ScienceMode2	ScienceMode3
Stim. channels	8	8	4
Current (at 1 kΩ load)	0 – 126 mA (2 mA steps)	0 – 126 mA (2 mA steps)	0 – 130 mA (0,5 mA steps)
Pulse width	0; 20 – 500 µs (1 µs steps)	0; 20 – 500 µs (1 µs steps)	0; 10 – 65520 µs (1 µs steps)
Frequency	up to 100 Hz (using all 8 channels); higher rate using lesser channels	up to 40 Hz (using all 8 channels)	up to 500 Hz using 4 channels; depends on the pulse width; higher rate using lesser channels, short pulse width
Pulse shape	Biphasic pulse		Biphasic pulse; Individual pulse shape using 16 points
Compatibility	ScienceMode1, ScienceMode2 and ScienceMode3 are incompatible		
Stimulation commands (from PC to the stimulation device)	**1. Initialize Stimulation:** • contains parameters that must be adjusted before start (frequency, activated channels, activated channels with partial frequencies) • The Stimulator initializes the values and is waiting for the start of the stimulation. **2. Update/Start Stimulation:** • contains parameters pulse width and current; the settings can be adjusted • The Stimulator starts the stimulation (as start) respectively adapts the transferred parameter (as update). • enables doublets/triplets (multiple impulses instead of only one) **3. Stop Stimulation:** • The Stimulator stops the Stimulation. **4. Single Impulse:** • contains as parameter one channel, one current and one pulse width. • The Stimulator has a single output of the impulse with adequate parameters		**Two Stimulation Levels** • **Mid-level:** RehaMove3 generates the set frequency • **Low-level:** Each stimulation impulse can be generated by the control program **Mid-Level** • **MI_Init:** Initializes stimulation • **MI_Update:** contains all stimulation related parameters • **MI_Stop:** deinitializes stimulation • **MI_Get_Status:** Get status information and live signal **Low-Level** • **LI_Init:** Initializes stimulation • **LI_Channel_Config:** contains individual pulse shape incl. current and pulse width • **LI_Stop:** deinitializes the stimulation More commands exists to get status information like battery, firmware version etc.
Typical scenario of the commands	Initialize, start, update, update, …, stop		**Low-Level** • LI_init, LI_ch_config, LI_ch_config, …, lI_stop **Mid-Level** • MI_init, MI_update, MI_get_status, …, MI_stop
Latency in the execution of stim. commands	2 ms	10 - 24 ms	1 ms

Electrode error detection/feedback	Yes/No	Yes/Yes	Yes/Yes
Emergency stop availabl./feedback	Yes/No	Yes/Yes	No/No
MOTOmed control	No	Yes, with 12 commands: **1. Start, Stop arm/leg trainer:** • set resistance, passive speed, flywheel mass, spasticity detection and more **2. Information from MOTOmed** • angles of the pedals, speed, torque • values of the particular phases of training: active performance, active distance, passive distance, symmetry (balance)	No
ScienceMode together with other programs (RehaMove, Sequence Mode)	Yes, but only specific combinations are possible.	It is not possible to use ScienceMode2 and other programs together.	Not applicable
C-Library and API	No	No	Yes
Simulink Control*	There is an open source Simulink-block. http://sourceforge.net/projects/sciencestim/	There is an open source Simulink block. http://sourceforge.net/projects/sciencestim/ (MOTOmed commands are not implemented)	An open source Simulink-block should be available soon at http://sourceforge.net/projects/sciencestim/
Simulink hardware and operation system support of open source block	• Simulation with Sync-Block (soft real-time) Win32/64, Linux32/64 Not yet supported • Real-Time Windows Target (hard real-time) • Eclipse IDE / Embedded Coder (soft real-time) not possible • xPC-Target	• Simulation with Sync-Block (soft real-time) Win32/64, Linux32/64 • Eclipse IDE / Embedded Coder (soft real-time) Win32/64, Linux32, Linux64 (not tested) Not yet supported • Real-Time Windows Target (hard real-time) • xPC-Target	To be determined

*Simulink is a block diagram environment for multidomain simulation and Model-Based Design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems. Simulink provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems. It is integrated with MATLAB, enabling you to incorporate MATLAB algorithms into models and export simulation results to MATLAB for further analysis.