Manual Schedule Files

Version: 2022-05-05

Schedule files can be created to automate the control of the tissue culture setup, for example to repeatedly run defined stimulation protocols or to define presets of stimulation parameters that can be quickly submitted via the software main window (Combobox *Stimulation Presets*).

A schedule file is a plain text file (*.txt) or comma separated file (*.csv) that can be created, edited and saved with any text editor. Microsoft Excel, Libre Office Calc, or similar Software can also be used. This may be especially helpful with complex schedules, because formulas can be used with these programs.

Note that **only semicolon or tab are allowed as field separators**. Thus, when saving as .CSV, semicolon must be used. This is the case by default with German language settings. If you do not have German language settings, **we recommend to save the files as tab-separated text files**. Using commas as separators will result in errors.

General syntax of schedule file commands

- Schedule files may contain one or several commands.
- Commands are case insensitive.
- Each command is written on a separate line.
- Each command begins with a time specification at which the command is to be executed.
- The time is specified either in seconds after the schedule start or in system daytime (HH:MM:SS).
- There are **commands, which do not require a channel, nor a parameter**, which follow this pattern:

  `[time]; [command]`

- There are **commands which do not require a channel, but a parameter**, which follow this pattern:

  `[time]; [command]; [parameter]`

- There are **channel-specific commands**, which follow this pattern:

  `[time]; [command]; [channel]; [parameter]`

- And there are **list commands**, which follow this pattern:

  `[time]; [command]; list; [parameters for channels 1-8]`

- Semicolons can be replaced by tab:

  `[time]  [command]  [channel]  [parameter]`

- All spaces are ignored, except within filenames and comments: Here, only the spaces before or after the file name or comment are ignored.
- Every text behind `//` will be ignored. Thus, `//` can be used to write notes or comments into the schedule file

  `[time]; [command]; [channel]; [parameter] //this is a comment`

- Empty lines can be inserted after commands for better readability.

Examples

Example 1: **Command without parameter or channel**
Save current rocker settings for later restore command, 400 seconds after schedule start:

```
400; saveRocker
```

Example 2: **Command which does not require a channel, but a parameter**
Load the subordinate schedule file "MySchedule.txt", 5 seconds after schedule start:

```
5; load; MySchedule.txt
```

Example 3: **Channel-specific command**
Set stimulation of channel 5 to 60 bpm, 10 seconds after schedule start, other channels won't change:

```
10; stimFrequency; 5; 60
```

Example 4: **Channel-specific command, using the *all* keyword**
Stimulate all channels with a current of 50 mA, at system daytime 13:53:20:

```
13:53:20; stimCurrent; all; 50
```

Example 5: **List command affecting all 8 channels**
Stimulate channels 1-4 with a current of 50 mA, channels 5-8 with 75 mA, 30 seconds after schedule start:

```
30; stimCurrent; list; 50; 50; 50; 50; 75; 75; 75; 75
```

Example 6: **List command affecting only some channels**
Stimulate channels 2, 4, 6 and 8 with a current of 60 mA, but do not change other channels, 40 seconds after schedule start:

```
40; stimCurrent; list; ;60; ;60; ; 60; ; 60
```

Important notes regarding list commands:

- List commands are especially useful to create schedule files with a spreadsheet program (e.g. Microsoft Excel), because they allow for the definition of all 8 channels in specified columns.
- A channel is not changed if the corresponding field is empty.
- Entries after the 8th channel column, i.e. after the 8th separator, will be ignored

Definition of the command time

- The time can be specified either **as runtime** of the schedule (in seconds, counting from the time point the schedule was started) **or as 24-hour daytime** (in HH:MM:SS format). The seconds must be provided, even if zero.
- Examples:

```
08:00:00; stimFrequency; all; 100 //stimulate all channels with 100 bpm
14:00:00; stimFrequency; all; 30 //stimulate all channels with 30 bpm
```

- When using daytime, the schedule file starts the following day if any daytime command is in the past of the current day. For example, if a schedule is started by the user on a Monday at 16:30, but the first command in the schedule file is defined for 11:30, then the whole schedule file will start the next day, i.e., on Tuesday, at 11:30. If, in the same scenario, the schedule is started on Monday at 10:00, then the schedule will start on Monday at 11:30.
- When using runtime commands, the schedule will always start as defined, no matter the daytime.
- Do not mix daytime and runtime commands in one schedule file.
- A main (parent) schedule can call subordinate (child) schedule files, using the `load` keyword.
- The parent schedule can be defined in daytime, while the child schedule files can be defined in runtime. This is handy when stimulation protocols shall be executed on a daily basis at a certain daytime and can be automated using the `repeat` command.

Save and restore parameters

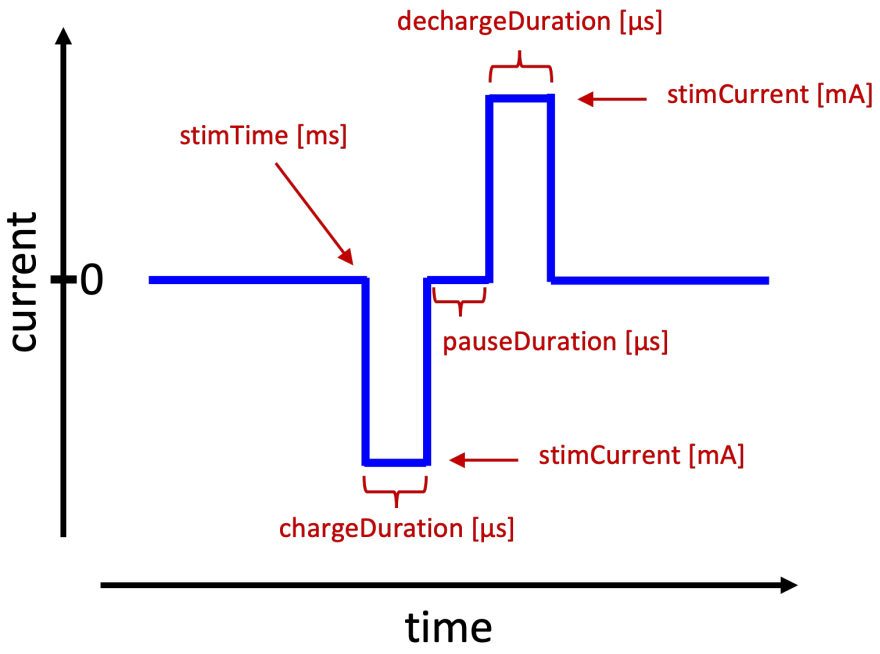
- It is possible to save current parameter settings with a corresponding save command
- The saved parameters can be restored later, for example when the schedule file defining a certain stimulation protocol has ended and the (baseline) parameters before schedule start, which may be unknown when the schedule file is created, need to be restored
- The `save` commands store the current settings in a stack, which are then retrieved and removed from the stack when the restore command is executed (last in first out principle).
- Therefore, make sure that each restore command is preceded by a matching save command earlier in the schedule file, i.e.

```
saveAll – restoreAll
saveRocker – restoreRocker
saveStimSequence – restoreStimSequence
saveStimPulses – restoreStimPulses
```

- Otherwise, the restore command will have no effect or may cause unpredicted behavior by restoring unintended parameters.

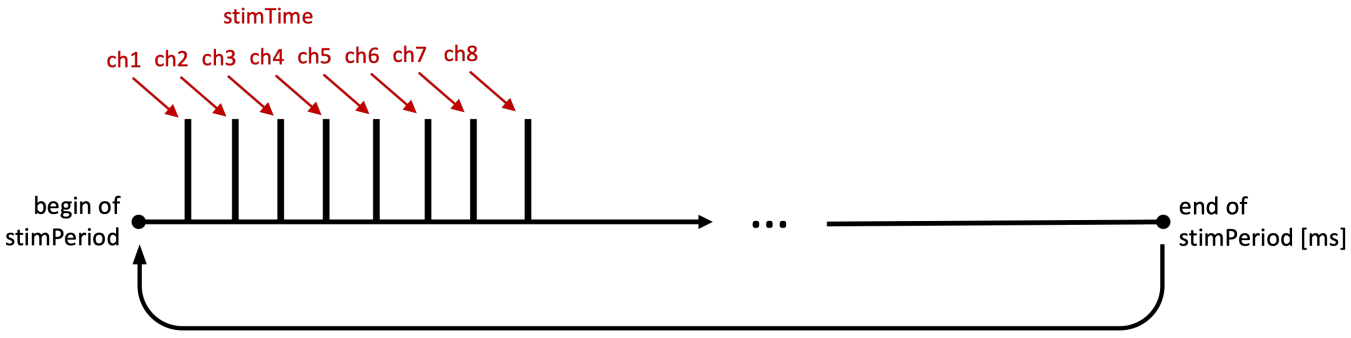
Definition of stimulation pulses

To define stimulation pulses in schedule files, it is important to first understand how the principle of slice stimulation in the cultivation chambers. The following schematic shows a typical stimulation pulse and the corresponding parameters that can be defined:



- `stimTime` defines when the pulse occurs within the stimPeriod (see below) in ms
- `stimCurrent` defines the amplitude (positive and negative) of the pulse in mA
- `chargeDuration` defines the duration of the first (negative) component of the pulse in µs
- `pauseDuration` defines the duration of the pause between negative and positive component
- `dechargeDuration` defines the duration of the second (positive) component of the pulse in µs

StimTime refers to the relative position of the stimulation pulse within a stimulation period. The stimPeriod command defines the duration the stimulation period in ms, after which all stim pulses are repeated:



Important notes:

- Only one channel can be stimulated at a time. When defining stimulation pulses and stimulation times, make sure that there is no overlap and at least 1 ms (better 2-3 ms) pause between successive pulses, regardless of the channel. This means that you need the take into account the total pulse duration, which is chargeDuration + pauseDuration + dechargeDuration.
- It is possible to define several stimulation times for the same channel within on period. For example, with a stimPeriod of 2000 ms, channel 1 can be stimulated at 0ms and 1000ms, yielding a

pacing frequency of 1 Hz, while channel 2 is stimulated only at 10 ms, yielding a pacing frequency of 0.5 Hz.

- By default, the pulse parameters (current, charge, pause and decharge durations) are applied to all pulses of the corresponding channel. However, in order to define different parameters for one channel within the same stimPeriod, you can use extra pulses.

Definition of extra pulses

To allow for more sophisticated stimulation schemes, we introduced the concept of extra pulses.
Extra pulses can be defined using the # sign. This way, it is possible to define up to 9 additional parameter settings (current, charge, pause and decharge) for each channel.
For example:

```
0; stimCurrent; 2; 25 //set current of the default pulse in channel 2 to 25 mA
0; chargeDuration; 2; 3000 //set chargeDuration to 3000 µs = 3 ms
0; pauseDuration; 2; 1000 //set chargeDuration to 1000 µs = 1 ms
0; dechargeDuration; 2; 3000 //set dechargeDuration to 3000 µs = 3 ms

1; stimCurrent #1; 2; 25 //set current of extra pulse 1 in channel 2 to 25 mA
1; chargeDuration #1; 2; 10000 //set chargeDuration to 10000 µs = 10 ms
1; pauseDuration #1; 2; 1000 //set chargeDuration to 1000 µs = 1 ms
1; dechargeDuration #1; 2; 10000 //set dechargeDuration to 10000 µs = 10 ms

2; stimPeriod; 2000 // stimulation period of 2000 ms
2; stimTime; 2; 0; 1000 //apply the default pulse at 0 ms and 1000 ms
2; stimTime #1; 2; 200; 1200 //apply the extra pulse at 200 ms and 1200 ms
```

This can be done in an analogous way for additional extra pulses #2 through #9 and for all channels.
#0 refers to the default pulse and can be omitted.

Alphabetical list of all schedule file commands

- The command key words are case insensitive and presented in alphabetical order.
- If the channel column is empty, the command does not require a channel.
- If the parameter column is empty, the command does not require a parameter

Keyword (case insensitive)	channel	Parameter [range]	Description
chargeDuration [#1 - #9]	[all, 1-8 list]	µs [0-15000]	Sets the charge pulse duration (negative voltage) to the provided time in microseconds; to avoid electrolysis, it is important to keep chargeDuration and dechargeDuration the same. It is therefore recommended to use pulseDuration, which sets both at the same time. [#1 - #9] can be used to define the parameter for extra pulses Examples: `08:00:00; chargeTime; all; 1000 //set chargeDuration of all channels to 1000 µs at 08:00 daytime`

comment	[all, 1-8]	free text [56 characters]	Writes the given text as comment to the log file; limited to 56 characters (longer comments will be cut off).
			`10; comment; all; medium exchanged //applies to all channels`
			`60; comment; 4; removed tissue slice //applies to channel 4`

comment		free text [56 characters]	Writes the given text as comment to the log file; limited to 56 characters `10; comment; started schedule file //general comment`

stimCurrent [#1 - #9]	[all, 1-8 list]	mA [0-80]	Sets the stimulation current of the provided channel; to set the stimulation current of all channels at once, use “all”
			[#1 - #9] can be used to define the parameter for extra pulses
			Examples:
			`50; stimCurrent; 1; 60 //stimCurrent of channel 1 to 60 mA`
			`100; stimCurrent; all; 70 //stimCurrent of all channels to 70 mA`

stimFrequency	[all, 1-8 list]	bpm [10-720]	Sets the stimulation frequency of the provided channel; to set the stimulation frequency of all channels at once, use “all” or do not provide a second parameter
			Examples:
			`10; stimFrequency; all; 60 //stimFrequency of all channels to 60 bpm`
			`30; stimFrequency; 3; 45 //set stimFrequency of channel 3 to 45 bpm`

Keyword (case insensitive)	channel	Parameter [range]	Description
dechargeDuration [#1 - #9]	[all, 1-8 list]	µs [0-15000]	Sets the decharge pulse duration (positive voltage) to the provided time in microseconds; to avoid electrolysis, it is important to keep chargeDuration and dechargeDuration the same [#1 - #9] can be used to define the parameter for extra pulses Examples: `18:00:00; dechargeDuration; all; 2000 //set dechargeDuration of all channels to 2000 µs at 6:00 PM`
load		file name	Loads and executes a second schedule file. Make sure that commands after the load command do not overlap in time with the loaded schedule file. If no directory is provided, the file will be searched in the folder of the calling schedule file. Example: `12:00; load; MyStimulationProtocol.txt //loads the stimulation protocol MyStimulationProtocol.txt schedule file`
pauseDuration [#1 - #9]	[all, 1-8, list]	µs [0-15000]	Sets the pause duration between the negative and positive pulses to the provided time in microseconds; 1000 µs is the recommended value [#1 - #9] can be used to define the parameter for extra pulses `120; pauseDuration; 7; 1000 //set pauseDuration of channel 7 to 1000 µs at 2 mins after schedule start`
pulseDuration [#1 - #9]	[all, 1-8 list]	µs [0-15000]	Sets the charge AND decharge pulse durations to the provided time in microseconds; [#1 - #9] can be used to define the parameter for extra pulses Examples: `08:00:00; pulseDuration; all; 3000 //set charge- and dechargeDuration of all channels to 3000 µs at 08:00 daytime`
polarity [#1 - #9]	[all, 1-8, list]	mode [0-2]	Modifies the pulse polarity pattern (0 is default): • 0: each stimulus is biphasic, with a first negative voltage (charge), then pause, then positive voltage (decharge) • 1: alternating stimulus pattern: one stimulus is positive, the following stimulus is negative, then positive, then negative, etc. • 2: biphasic and alternating, with a negative-pause-positive, then positive-pause-negative, and so on [#1 - #9] can be used to define the parameter for extra pulses `25; polarity; 6; 2 //set polarity of channel 6 to biphasic and alternating`
repeat			Repeats (reloads) the schedule file; use this command at the end of a schedule file if you want it to repeat itself over and over. Note that commands after the repeat command will never be reached. Example 1: `23:59:00; repeat //reloads the schedule file at 23:59`

Keyword (case insensitive)	channel	Parameter [range]	Description
			Example 2: `10; repeat //reloads the schedule file 10 sec after schedule start`
restoreAll			Restores the previously saved rocker, stimulation pulse and stimulation sequence settings
restoreRocker			Restores the previously saved rocker settings (power and speed)
restoreStimPulses			Restores the previously saved stimulation pulse settings (chargeDuration, dechargeDuration, pauseDuration and stimCurrent) of all channels
restoreStimSequence			Restores the previously saved stimulation sequence (always affects all channels)
rockerPower		AU [60-80]	Sets the force and power of the rocker, 80 is recommended and default
rockerSpeed		rpm [1-90]	Sets the rocker frequency in round per minute; 60-70 is recommended Rocker Speed can be set to 0 for up to 10-20 seconds to avoid rocking artefacts. This improves the signal-to-noise ratio and is recommended for data analysis, especially if contraction amplitudes are low. Note that rockerSpeed 0 for more than 10-20 seconds may reduce oxygen supply of the cardiac slices.
saveAll			Saves rocker settings, stimulation pulses and stimulation sequence

recordAnalysis		file name	*Not included in basic version of MyoDish Software. Requires the analysis software package.*
			
			Saves the contraction analysis data into the provided file name. The file will be saved into the folder of the current data file. If no file name is specified, the data file name will be used and extended by "_analysis.txt".
			Example:
			`10; recordAnalysis //starts analysis, makes sense to wait 10 beats, if, for example, 10 beats are averaged`
			`20; recordAnalysis; analysisData.txt //saves the data into analysisData.txt`
			`21; stopAnalysis //stops analysis to save computation power`

saveRocker			Saves the current rocker parameters, making it possible to load them again later with the restoreRocker command Example: `120; restoreRocker`

Keyword (case insensitive)	channel	Parameter [range]	Description

saveStimPulses			Saves the current stimulation pulse settings (chargeDuration, dechargeDuration, pauseDuration, stimCurrent) of all channels for later restoreCommands; note that save commands can be executed repeatedly several times
			Example:
			`0; saveStimPulses //stim pulses 1 saved`
			`//some commands`
			`100; saveStimPulses //stim pulses 2 saved`
			`//some commands`
			`200; restoreStimPulses //now stim pulses 2 are restored`
			`//some commands`
			`300; restoreStimPulses //now stim pulses 1 are restored`

saveStimSequence			Saves the current stimulation sequence for later use of restoreStimSequence commands
startParallelRecording		file name	Starts parallel recording into the file name provided; use this option to record certain experiments or the whole schedule (if used as the first command) in an additional data file Example: `0; startParallelRecording; schedule1.mdd //now saves the data additionally into the provided file`

Keyword (case insensitive)	channel	Parameter [range]	Description
stimPeriod		ms [100-10000]	Sets the stimulation sequence period. **Important note**: This command deletes all previously defined stim times. Thus, it requires additional information on the channel stimulation times on subsequent lines. Example: `30; stimPeriod; 1000 //stimulation period 1000ms`

stimTime
[#1 - #9]
	[1-8, list]	ms
		[0-stimPeriod-1]
			Sets the stimulation times within a stimulation period.
			[#1 - #9] can be used to define the parameter for extra pulses
			Example with 1 Hz stimulation of channels 1, 2, 3, 7 and 8, 2 Hz stimulation of channel 5, 3 Hz stimulation of channel 6, and no stimulation of channel 4:
			
			`30; stimPeriod; 1000 //stimulation period 1000ms`
			`30; stimTime; 1; 30`
			`30; stimTime; 2; 60`
			`30; stimTime; 3; 90`
			`30; stimTime; 5; 150; 650`
			`30; stimTime; 6; 180; 513; 846`
			`30; stimTime; 7; 210`
			`30; stimTime; 8; 240`
			`31; rockerSpeed; 70 //ends the block`
			
			The previous block is equivalent to:
			`30; stimPeriod; 1000 //stimulation period 1000ms`
			`30; stimTime; list; 30;60;90; ;150;180;210;240`
			`30; stimTime; list;  ;  ;  ; ;650;513;  ;`
			`30; stimTime; list;  ;  ;  ; ;  ;846;  ;`
			
			Channels are stimulated serially and an interval of at least 10ms between each stimulation time is required. This means that channel stimulation times must not be closer than 10 ms.
			Examples:
			`30; stimPeriod; 1000`
			`30; stimTime; 1; 10; 11; 12 //wrong`
			
			`30; stimPeriod; 1000`
			`30; stimTime; 1; 10`
			`30; stimTime; 2; 10 //wrong`
			
			`30; stimPeriod; 1000`
			`30; stimTime; 1; 10;`
			`30; stimTime; 2; 11 //wrong`
			
			`30; stimPeriod; 1000`
			`30; stimTime; 1; 10;`
			`30; stimTime; 2; 25 //okay`

startAnalysis			*Not included in basic version of MyoDish Software. Requires the analysis software package.* Starts the contraction analysis (Fmax, dFdtmax, dFdtmin, etc.)

Keyword (case insensitive)	channel	Parameter [range]	Description
stopAnalysis			*Not included in basic version of MyoDish Software. Requires the analysis software package.* Stops contraction analysis
stopParallelRecording			Use this command to stop the parallel recording, for example at the end of the schedule file Example: `600; stopParallelRecording //end`

See our example schedule files. A typical schedule file may look like this:

```
//example schedule file1.txt
0; saveAll //saves the current settings (rocker, stim pulses, stim sequence)


0; stimPeriod; 2000 //sets the stimulation period to 2000 ms
0; stimTime; 1; 0; 1000 //stimulate channel 1 at 0 and 1000ms within the period
0; stimTime; 2; 25; 1025 //1 Hz
0; stimTime; 3; 50 //channel 3 is stimulated with 0.5 Hz
0; stimTime; 4; 75 //channel 4 is stimulated with 0.5 Hz
0; stimTime; 5; 100; 1100 //1 Hz
0; stimTime; 6; 125; 1125 //1 Hz
0; stimTime; 7; 150; 1150 //1 Hz
0; stimTime; 8; 175; 1175 //1 Hz
30; saveRocker
30; rockerSpeed; 0 //stop Rocker to reduce shaking artifacts
40; restoreRocker
//some more commands
60; restoreAll //restores the rocker, stimpulse and stim sequence settings
//end of file
```

Another schedule file, file2.txt, could call file1.txt at a certain time of the day.
Example:

```
//example schedule file2.txt
01:00:00; load; file1.txt //executes the schedule defined in file1.txt at 1:00 AM


02:00:00; repeat //repeats file2, i.e. will load file1.txt again at 01:00 daytime the
next day
```