

## Contents

### Welcome

### Training days

### Latest software

### Future meetings

### Script Spotlight

- Activity from video



- Memory channels
- Script – Generating events from threshold crossings



- Automation in experiments
- Script – Importing stimulator instructions from text

### Did you know...?

- Scale bars

### Recent questions

- How to create long trains of pulses

### CED user forums

## Welcome

Thank you for downloading the third issue of our newsletter. These are difficult times for all of us and we hope you, your family, and colleagues are all staying safe and well. It is times such as this that we all need to come together to support one another, and we are confident that the scientific community will adapt to these changes and continue to thrive.

Be assured that CED is still open for business, and we are continuing our daily operations. Our software team is still developing and providing updates for Spike2 and Signal, and our hardware team is still at hand for any help you need. Our customer support teams are available to answer any technical questions and provide remote training via Skype, Zoom or TeamViewer. We may be contacted using all the usual channels; however, we ask that contact be done via email where possible. If you do need to speak to someone directly, please send us an email to request a call back in the first instance. Up to date information can always be found on our website, and the CED user forums are also available for technical help from our super users.

We are also endeavouring to add more video tutorials to our existing library and would like your input on topics you think would be useful to cover. Please get in touch at [Marjorie@ced.co.uk](mailto:Marjorie@ced.co.uk) and let us know your ideas!

---

## Training Days

Due to the ongoing COVID-19 outbreaks, all current training events have been put on hold. We do however offer remote training sessions via Skype either one-to-one or with groups.

Join us and learn how to make the best use of Spike2 and Signal to save hours of repetitive analysis. Our remote sessions are free to arrange and are suitable for both existing and prospective users of our data acquisition and analysis systems. If you would like to schedule a session, please get in touch.

If you are interested in hosting a training event in your local area once social distancing measures have been eased, please get in touch: [Marjorie@ced.co.uk](mailto:Marjorie@ced.co.uk).

---

If you see these buttons in our newsletters, it means a file or script relating to the section is available to download:



## Latest versions of Spike2 and Signal

Spike2	Released	Signal	Released
<a href="#">Version 10.04</a>	03/2020	<a href="#">Version 7.05a</a>	02/2020
<a href="#">Version 9.10</a>	02/2020	<a href="#">Version 6.05b</a>	10/2019
<a href="#">Version 8.19a</a>	11/2019	<a href="#">Version 5.12a</a>	02/2018
<a href="#">Demo</a>	03/2020	<a href="#">Demo</a>	02/2020

[Back to contents](#)

## Future meetings and events

### [Europhysiology 2020](#)

Berlin,  
Germany  
September 11<sup>th</sup> – 13<sup>th</sup> 2020

### [Neuroscience 2020](#)

Washington, DC,  
USA  
October 24<sup>th</sup> – 28<sup>th</sup> 2020

### [ISAC XXI 2020](#)

Lisbon,  
Portugal  
November 3<sup>rd</sup> – 6<sup>th</sup> 2020

[Back to contents](#)

## Script Spotlight

Our software team has developed a script which converts motion in a video recorded with the Spike2 multimedia facility (.avi or .mp4) into an "Activity" trace in the corresponding time view. The script does not attempt to track moving objects, but instead converts the video to grey-scale and interprets differences in intensity between frames, pixel by pixel, in a user-defined area of the image (for example as due to motion rather than flashing lights). The result is qualitative, with the output units as 'mean intensity change per pixel between video frames'. The maximum activity value is 100, where all pixels are undergoing maximum possible grey-scale change between frames; in practice activity values will be lower.

This approach is not sensitive to colour. Therefore, movement in areas of high contrast and many contours will register disproportionately and changes in light intensity will register as movement. Despite these caveats, the script will be useful for applications such as sleep scoring of rats or mice moving around in relatively homogeneous environments. For example, an animal, or up to 6 animals in adjacent cages, can be videoed from above. The video is then processed offline with this script to create an activity trace for each animal. The area of the video frame to analyse for each animal may be specified by clicking and dragging a rectangle over the relevant area of the video frame to select it. The activity traces for all the animals can then be generated simultaneously.

The activity traces could then be used as part of the input to a semi-automated sleep staging script (e.g. OSD4.s2s) based on a combination of activity level and banded power spectra. Other applications might include assessing time spent by a single animal in different parts of a behavioural arena. We appreciate any feedback for this script, and hope it is of benefit to you.

[Download the script from our website.](#)

[Back to contents](#)



## What is a memory channel for, and how do I use them?

Spike2 stores memory channels in system memory. They are used for easily modifying data which cannot be achieved in disk channels, as these are optimized for recording large quantities of unchanging data. Memory channels are widely used within scripts to create new channels of copied data or even entirely new script generated data. There is also the option to manually enter data into memory channels.

Memory channels can be of any channel type and are displayed in the same way as other channels, i.e. a memory based level channel will appear the same as a disk based level channel. When you have finished modifying the data in the memory buffer it can be written to the data file as a disk channel or discarded; if you do not write memory channels to disk they are lost when the data file is closed.

To create a new memory buffer, navigate to the analysis menu and select *Memory Buffer -> Create New buffer*. You're presented with a simple dialog to choose the buffer type using the drop-down menu. Click *OK* to create an empty channel or click *Import* to copy data from a source channel.

If you choose to import from a Waveform or RealWave channel when creating an event or marker buffer you are presented with the option to choose the mode of import. The modes available are *Peaks*, *Troughs*, *Peaks and Troughs*, *Data rising through level*, *Data falling through level*, and *Rise and fall through level*. The events or markers are created based on the *Mode* chosen, with the *Size* of the peak/trough or *Level* being supplied in Volts. The *minimum interval* between events in seconds also needs to be supplied, and the range of data to process can be specified using the *start time* and *end time* fields. For example, if one needed to import *Peaks* at least 100 ms apart which exceeded 1V in amplitude as an event I would choose the settings on the right:

Import to Demo.smr[32-bit] memory buffer

Buffer	m1 Memory (Event-,2001)	seconds
Channel	1 Sinewave (Waveform)	seconds
Start time	0.0	seconds
End time	MaxTime()	seconds
Mode	Peaks	
Minimum Interval	0.1	seconds
Size	1.0	Volts

Help Close Apply OK

Creating a direct copy in memory of a disk channel is simpler. Navigate to the *Analysis* menu and select *Memory buffer -> Create Channel Copy*. Alternatively, right click the channel you wish to copy in the data and select *Copy as memory channel*. Specify the start and end time for the data to copy in the new dialog and click *OK*. If your source channel has been modified you will be presented with a warning.

To manually add a new event item to an existing memory channel, navigate to the analysis menu and select *Memory buffer -> Add items*, or right click the memory channel and select *Add Items* from there. Choose the time of the data to add, the marker code if adding a marker, the value of the wave if adding a waveform, and click *OK*. Manually deleting an item from the memory buffer follows the same format as adding an item; from the *Delete Items* dialog you can choose to delete items around a time point, between a time range, or delete all items.

Memory channels are useful for holding your data to manipulate. However, there is a limit to the quantity of data that can be held in memory and accessed quickly. 32-bit applications are limited to 4GB of memory. 64-bit applications can address much more. There is limited physical memory in your PC, maybe 8 or 16 GB. Once this is exhausted the operating system 'swaps out' what it thinks is currently unused physical memory to disk to make more room. This means that if you generate huge quantities of data in memory channels, all operations get slow as the system spends its time saving and restoring memory from disk.

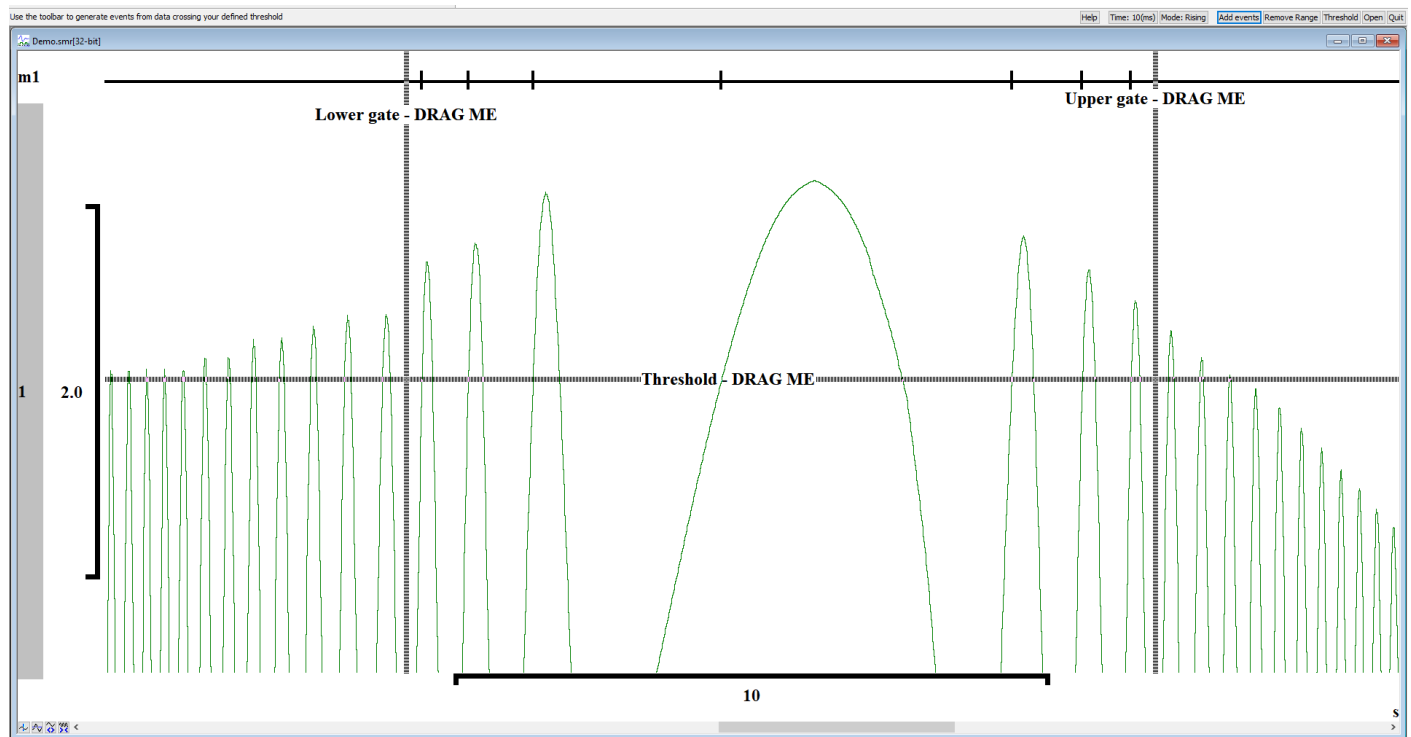
[Back to contents](#)

## Scripts: Spike2



Following on from the introduction to memory channels above, we have combined many of the elements we have discussed in the past three newsletter issues to create a script which generates events at the point which waveform data crosses a threshold. This script creates a memory channel which uses the *import channel* function to generate these events. An idle function continually updates the memory buffer each time the threshold or time range is altered.

Upon running the script, use the *Open* button in the script generated toolbar to load your data. Use the *Help* button to display guidance. We have used the *Demo.smr* file for our example here; this data file is stored in your user data folder, usually located in C:\Users\Username\Documents\Spike2\Data.



After opening your file, you need to choose the channel you wish to generate events from by clicking the channel number in the Y axis (the script will show these for you if you have hidden them previously). You then need to add the threshold and optionally a time range before using the *Add events* button to create the memory channel.

Much of the script is concerned with correctly selecting and positioning views and error checking. The main script element of interest is `MemImport(chan%,inCh%,start,end{,mode%,time,level{,code%}})`; this is contained in the `Events%` toolbar function where `start`, `end`, and `level` are specified by the cursors on screen. The *Mode* is toggled using the associated button in the toolbar, and the minimum *Time* between events can also be specified via its button. The idle function continually checks for the current positions of the placed horizontal cursor (`level`) and the difference between the two vertical cursors (`start` and `end`) and compares them to their last known values. If these values differ the script runs the `Update%` function which uses `MemDeleteItem(chan%{,index%{,num%}})` to remove all items from the memory channel, and imports again with `MemImport()` using the new positions of the threshold and time gates.

[Download the script](#) and try it out for yourself. We hope you find it useful.

[Back to contents](#)



The automation tab of the sampling configuration provides this functionality. The tab controls three limits that will cause sampling to stop automatically: *Number of Frames*, *Sampling duration*, and *File size*. These limits only cause the sampling to stop, not finish; this means you have the choice of flipping through the frames and deciding whether to continue sampling or finish and save the data file. Clicking the *More* button in the sampling controls toolbar once sampling reaches a pre-set limit allows another limits-worth of sampling to occur before stopping again. For example, if the frame limit were set to 25, clicking *More* would cause sampling to continue until the data file had 50 frames. Clicking *More* again would give 75 frames and so on.

Each option has a check box to enable the limit, plus a field for entry of the limit value. If the check box for a limit is clear, or if the corresponding limit value is set to zero, then that limit is ignored.

Say for example we wished to obtain exactly 20 frames of evoked response data; we enable the Number of Frames limit by checking the box and enter 20 into the dialog field. We create an evoked response experiment using multiple frame states, of which 10 of the 20 frames will have a single pulse delivered to a stimulator and the remaining 10 will have a paired pulse delivered. Upon reaching the limit, we check through the frames of data to see if the appropriate evoked response was achieved. In doing so we find

anomalies on a few frames that we make note of and now wish to repeat.

We first uncheck the *Write to disk* box and select *Pause at end*, swap to using *Manual* state control for the experiment and select the *Idle* state to prevent firing the stimulator. By clicking *More*, Signal is now ready to begin capturing more frames manually. We select the desired state in the state bar and click *Continue* to trigger another sweep. If the frame is good, we can choose to *Accept* this new frame to write it to disk. You can capture as many

more frames as the original limit, so in this case another 20 to give a total of 40 frames. However, we can *Stop* early once we have gotten the correct frames and *Finish* once we are happy.

The [sampling configuration](#) attached to this newsletter lets you try this out for yourself.

Another automation tool within Signal is the *Artefact rejection* dialog. This is also accessed in the *Automation* tab of the sampling configuration. When enabled, Signal automatically examines newly sampled data. You have the choice to set Signal to either reject or tag new frames for when data has reached the ADC limits. Coupling automatic artefact rejection with your sampling limits is a useful aid in capturing the exact data you require.

[Back to contents](#)



## Scripts: Signal



In our last newsletter, we discussed setting up protocols for controlling stimulators from Signal. It can be sometimes be useful to import a previously generated protocol through other means into Signal. Rather than copying the protocol verbatim, this script enables the user to generate a protocol for a Magstim BiStim stimulator as a text file and import the settings as needed. The script will condense the protocol into the minimum number of states required to play out all steps of the protocol. The correct timings of the stimulator firing and the intensity are kept for each step.

The format required for the text file protocol is as follows:

POWER A% POWER B% PULSE INTERVAL(ms) SWEEP INTERVAL(s)

The power % can be either a value of Maximum Stimulator Output (MSO) from 0-100%, or as a percentage of motor threshold (MT) %. If you specify the latter, you must also provide the MT % value; you are given the choice to specify either working in MSO or MT when importing the protocol.

In the text file, each element **MUST** be separated by a tab for the script to correctly read each setting. A text file using POWER% as a percentage of MT, providing three instructions, would look something like the example below. We have also included a few example text files to download with the script to try out for yourself:

80	120	15	7
80	120	3	6
80	0	0	0



To use the script, ensure your BiStim's Trigger Out is connected to the Trigger port of your 1401. You must also set the BiStim to Independent Triggering mode (also known as IBT mode), connect Digital Output 0 to the master stimulator (Power A), and connect Digital Output 1 to the secondary stimulator (Power B). When running the script you need to first use *Load Protocol*. After selecting your protocol, the script calculates how many states are required and the order of the states needed to play out each instruction. The Sample button is then enabled, when clicked the script asks you to provide a trigger before sampling will begin. You may progress the script to this point before setting up the stimulator, but your stimulator must be connected to the 1401 as described above before continuing. When ready, manually fire a pulse from the BiStim to provide the trigger to the 1401, which then prompts the script to check all is OK before finishing the set up. Sampling then proceeds using the loaded protocol.

Whilst this script only currently functions with a Magstim BiStim, it is possible to modify the script to work with other Signal supported devices. If this is something you are interested in, please get in touch: [info@ced.co.uk](mailto:info@ced.co.uk).

[Back to contents](#)

## Did you know...?


Scale bars are available to use within Spike2 and Signal to help create useful figures for your reports. To enable scale bars, right click the data and click *Show/Hide channels* for Spike2 or *Customise display* for Signal. Here you are given options for customising the Y and X axis. Ticking the *Scale bar* option under either axis list will enable it. It's best to play around with your other draw settings as well to better portray your data, but if you get lost you can always revert to the default settings by navigating to the *View* menu and selecting *Standard display*.

[Back to contents](#)

## Recent Questions

### ***How do I create a long train of several thousand pulses?***

There are several ways one could create a large train of pulses within Spike2 and Signal. The *graphical pulse editor* may be used to create a pulse train of up to 1000 pulses, the repeat field would then be used to repeat the train however number of times is needed to create the long train. However, this method less efficient as each pulse is stored as an instruction to be played out. It is preferred to specify a smaller number of pulses in one section and repeat the required number of times. It is even more efficient to delve into the sequencer language to create a single



pulse which is repeated the desired number of times, as you can then use the minimum number of sequencer instructions. The sequencer is limited to 8191 instructions, although you are unlikely to reach this it is still preferable to use the least amount of instructions per section as it leaves more room for further instructions should your experiment evolve and require more later.

In the sequencer language, a train of 5000 pulses of 1ms length and 1ms interval could be created as this:

```
                SET      0.1,1,0          ;Get rate & scaling OK

                VAR      V45,Loop=5000    ;Define variable for section loops

0000            HALT
0001 LA:        'A    DIGOUT [...1]      ;Loopback point
0002            DELAY   s(0.0009)-1
0003            DIGOUT [...0]
0004            DELAY   s(0.0008)-1
0005            DBNZ    Loop,LA           ;Repeat required times
0006            HALT                    ;End of this sequence section
```

[Back to contents](#)

## CED User forums

Try the [CED Forums](#) bulletin board for software and hardware support.

If you have any comments about the newsletter format and content, please get in touch: [Marjorie@ced.co.uk](mailto:Marjorie@ced.co.uk).

To adjust your subscription preferences, please visit our website: [www.ced.co.uk/upgrades/subscribeenews](http://www.ced.co.uk/upgrades/subscribeenews).

[Back to contents](#)

### Contact us:

#### In the UK:

Technical Centre, 139 Cambridge Road,  
Milton, Cambridge, CB24 6AZ, UK

**Telephone:** (01223) 420186

**Fax:** (01223) 420488

**Email:** [info@ced.co.uk](mailto:info@ced.co.uk)

**International Tel:** [44] 1223 420186

**International Fax:** [44] 1223 420488

**USA and Canada Toll Free:** 1 800 345 7794

**Website:** [www.ced.co.uk](http://www.ced.co.uk)

All Trademarks are acknowledged to be the Trademarks of the registered holders.  
Copyright © 2020 Cambridge Electronic Design Ltd, All rights reserved.