

Contents

- [News](#) – Updates & meetings
- [Spike2](#) – Large data files
- [Signal](#) – Single channel patch clamp analysis
- [Spike2 Script](#) – Overdraw channels from multiple result files
- [Signal Script](#) – Trigger sampling between two levels
- [Did you know...?](#) – Sequencer updates during sampling
- [Recent questions](#) – Keyboard shortcuts for optimizing y-axis display
- [Customer contribution](#) – Dynamic clamp implementation with Spike2

News

US Training Days

Our US training days are now confirmed for the 15th and 16th of October at the Embassy Suites Chicago Downtown hotel, prior to the Society for Neuroscience annual meeting. The training days are offered at \$250 per day, including continental breakfast buffet and lunch as well as coffee and tea, and are suitable for both existing and prospective users of Spike2 and Signal. Places are filling up quickly so if you or any of your colleagues would like to attend, please follow the links to view the [training day program](#) and complete the [registration form](#). Please give a short description of the type of research that you do and analysis that you are interested in. This will allow us to tailor sessions for specific requirements.

Meetings and events

[Southampton Neuroscience Group \(SoNG\) Annual Meeting](#)

University of Southampton
Southampton, UK
September 25th 2009

[Neuroscience 2009 - 39th Annual Meeting](#)

McCormick Place
Chicago, IL, USA
October 17th – 21st 2009

[8th Biennial conference of the Chinese Society for Neuroscience \(CSfN\)](#)

Guangzhou Huatai Hotel
Guangzhou, China
November 7th – 10th 2009

[Federation of European Physiological Societies 2009](#)

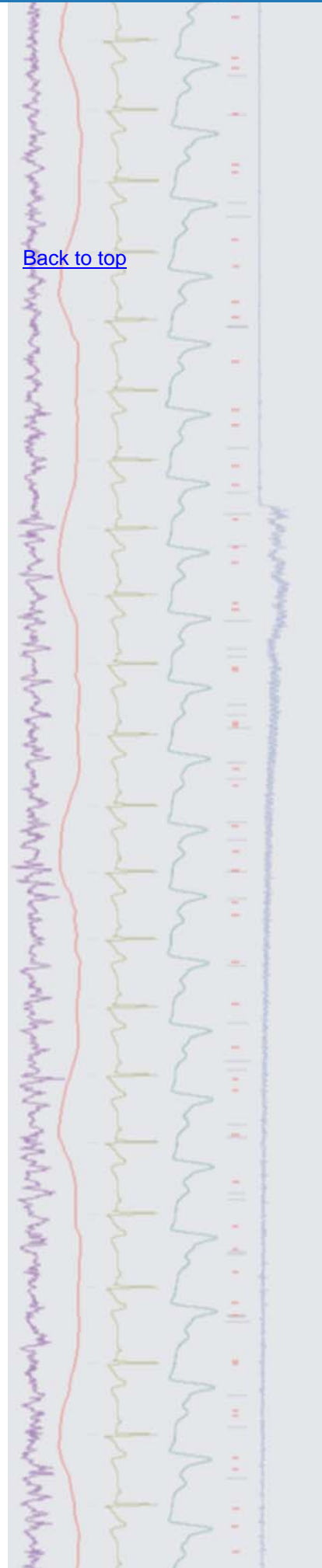
University Medical Centre Ljubljana
Ljubljana, Slovenia
November 12th – 15th 2009

Latest versions of Spike2 and Signal

Updates for Spike2 and Signal are available from the [CED downloads](#) page, or by clicking on the links in the table below. Demonstration versions of the latest software are also available.

Spike2 downloads	Signal downloads
Spike2 version 7.01	Signal version 4.05
Spike2 version 6.12	Signal version 3.12
Spike2 demo	Signal demo

[Back to top](#)

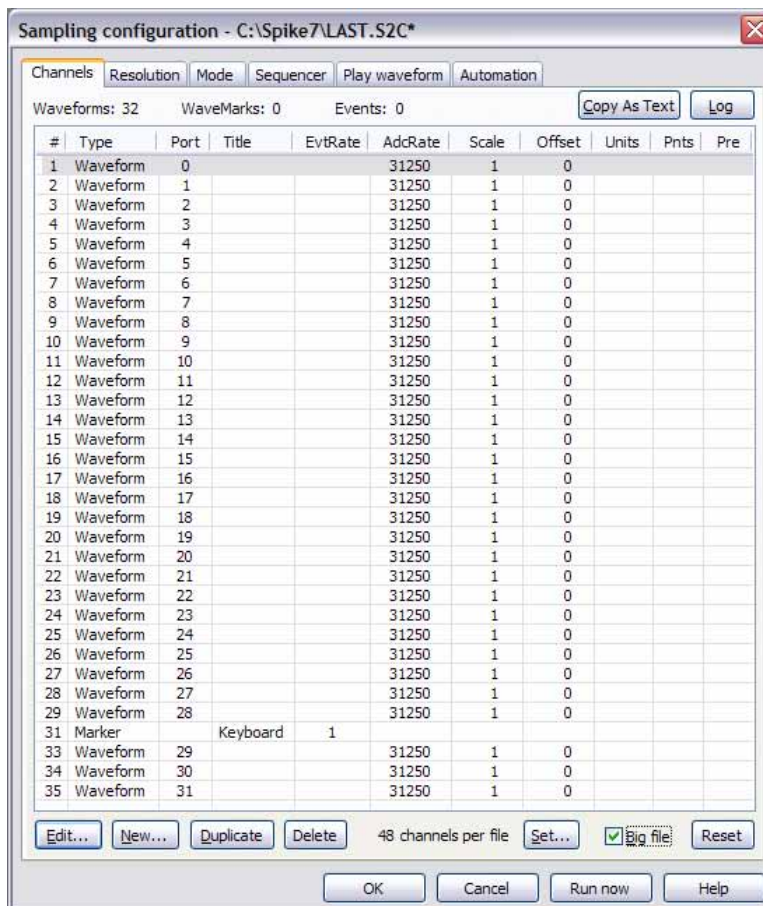




[Back to top](#)

- Q. Is there any way to create data files larger than 2GB when sampling?
- A. Spike2 now supports data file sizes of up to 1TB when using the Big file format option in the sampling configuration Channels tab.

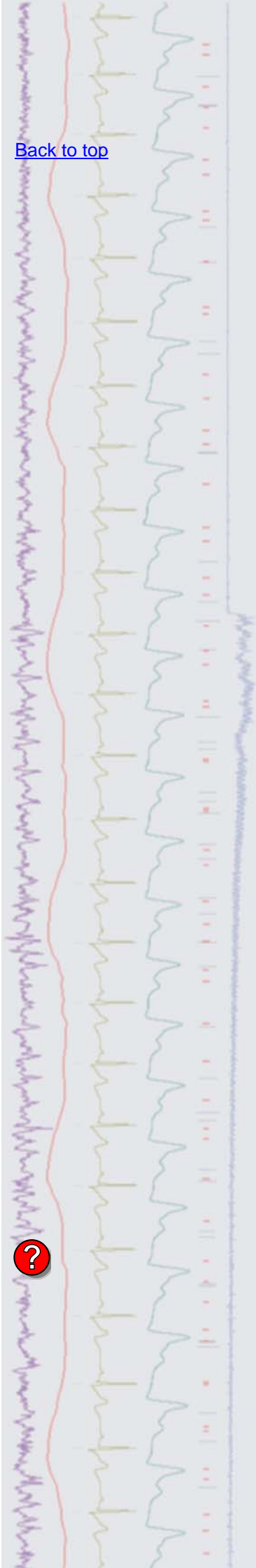
There are two limits to the size of a Spike2 data file: the number of clock ticks in the file (limited to 2147483647) and the physical size of the file. The physical size before version 7 was limited to 2147483647 bytes, or 2 GB. For most users of Spike2, these limits are not a problem. However, it is possible to run out of disk space before you run out of clock ticks. For example, if you set the Waveform sample rate to be one clock tick and sample one channel, each sample point uses 2 bytes of data. If you set Burst mode sampling in the Resolution tab with n channels, each taking one sample per clock tick, then each clock tick is using 2n data bytes. This means that your maximum possible run time is limited to $2147483647/2n$ clock ticks.



Sampling configuration with Big file option enabled

Without using Big file mode, if you were sampling 32 channels of data at 31.25 kHz each (the maximum possible with a Power1401 mk II), in non-burst mode you would need to set the clock tick to 1 microsecond, which means that the clock tick limit would be some 35 minutes of sampling. However, the file size limit would be hit in around 17 minutes. If you changed to burst mode sampling, the clock could be run at 32 microseconds per tick, which extends the clock limit to 19 hours, but the file size limit is still 17 minutes.

If you check the Big file option, the maximum file size is increased by a factor of 512 allowing sampling for the full time allowed by the clock tick period.

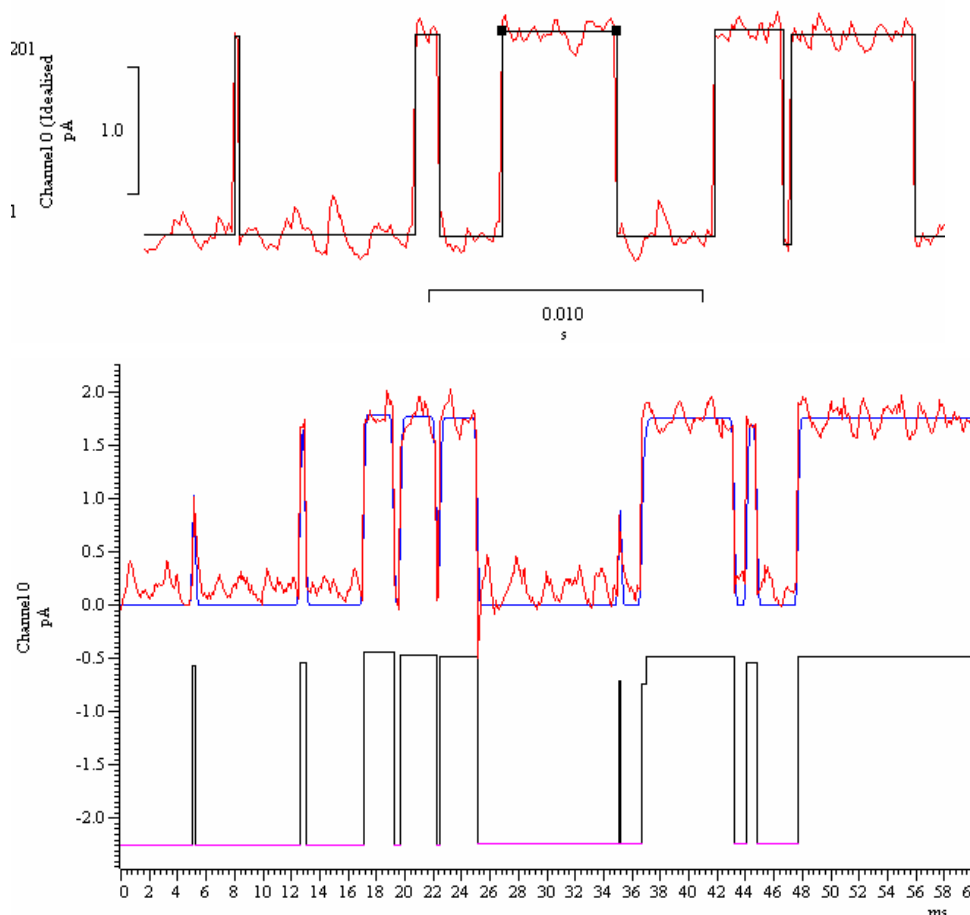


Signal

[Back to top](#)

- Q. Is Signal suitable for analysis of single channel patch clamp data?
- A. There are a number of features for single channel patch clamp data analysis available, including idealized trace formation and generation of dwell time and amplitude histograms. An idealized trace is formed by converting the raw data trace into a series of open/closed times (events). Each time has an amplitude, a duration and a set of 'flags' to identify it.

There are two methods of creating a new idealised trace from the Analysis menu Open/Closed times command: Threshold and SCAN. The Threshold method involves setting an Open and Closed threshold level to detect the opening of ion channels for inward and outward currents. This method is more suitable if you have more than one channel in your patch and the resultant trace is drawn overlaying the raw data. The SCAN analysis method (originally developed and written by professor David Colquhoun at University College London) assumes that a Gaussian filter was used to remove noise from the original signal and generates a high time resolution calculation of the original, unfiltered waveform using a form of reverse convolution of the idealized trace with the step response function of the filter. The result is drawn with the convoluted trace overlaying the data and the idealized trace drawn below.



Idealised traces generated using the threshold method (top) and SCAN analysis method (bottom)

Once an idealised trace has been generated from the data, you can view and edit details of the events by right-clicking on the trace and selecting Show event details from the Edit menu pull-down. You can also click and drag the selected level on the trace to adjust amplitude and dwell times. Open/Closed time histograms, Amplitude histograms of the open/closed time durations and burst histograms can all be created from the Analysis menu. The settings for all histograms allow you to include or exclude data based on the associated 'flags'.



Further details on idealized trace generation and analysis can be found in the on-line help.

Scripts: Spike2

- Q. I would like to be able to take channels from multiple waveform averages and merge them into a single file so that I can display them overlaid. Is there a script available that could do this?
- A. The attached script, `MergeODResultViews.s2s`, creates a new XY window displaying overdrawn data from multiple result views selected in turn by the user. By default the script takes data from the first channel of the result view and displays each channel (up to a maximum of 256) overdrawn and in separate colours in the new window.

Scripts: Signal

- Q. I would like to be able to trigger sampling in Signal when a designated channel is between two threshold levels. Peri-trigger mode allows me to trigger when there is a transition from within to outside a pair of levels, but not when the waveform is between levels. Is there any way to achieve this?
- A. The attached script, `TrigInLevels.sgs`, and sequence file, `TrigInLevels.pls`, waits until a sampled waveform is between two levels on ADC port 0 (channel 1) before triggering sampling and then waits for a defined time before arming the next sweep. The threshold levels are defined using horizontal cursors and the script uses *peri-triggered sampling* mode and *free run without restarts* mode for the outputs. This sequence could easily be modified to generate additional outputs for the sweep or respond to multiple states conditions if required.

Did you know...?

You can now update the Spike2 output sequencer by loading sequence files during sampling. The new sequencer file replaces the previous one.

Recent questions

- Q. Are there keyboard shortcuts in Spike2 for optimising the data in all channels in a file? This would be much easier than opening the Y-axis dialog and setting all channels to optimise.
- A. There are many functions in both Spike2 and Signal that have associated 'hotkey' keyboard shortcuts. The `Ctrl+Q` shortcut optimises all channels in a file.
- A full list of keyboard shortcuts can be found in the on-line help for both Spike2 and Signal.

Customer contribution – 4 channel dynamic clamp implementation using Spike2 and 1401plus

Dynamic clamp methods are a standard in electrophysiology that increase flexibility in the manipulation of membrane processes—e.g., addition and subtraction of ion channels, synapses, and additional component neurons in a network—however, these methods' time resolution and stability are limited by computing speed and model complexity. In this application, we used the built-in real time output capability of Spike2 with a 1401plus interface to implement a dynamic clamp to arbitrarily rewire a motor pattern generator in vitro.

Using up to 4 intracellular recordings from different neurons, we added and subtracted electrical couplings between these neurons. This was done using a sequencer file containing a loop that calculated currents to inject into each neuron at precise, small timesteps, i.e. in "real time". These currents were solutions to Ohm's Law between neurons and Kirchoff's Current Law at each node of the network (where each neuron was treated as a node). The attached script (`4ChanDynClamp.s2s`) and sequence file (`4NeuronCoupling.pls`) is a basic example of implementing a 4 channel dynamic clamp using user-defined settings that define the connectivity of the network (coupling of neurons) and a global conductance value, G , that defines the strength of every connection. Editing of these settings and sampling of data is controlled via the buttons in the script toolbar.

[Back to top](#)

Right-click the script icon and save to disk.

If you have any problems opening the embedded scripts in this newsletter please let us know.

[Back to top](#)

Right-click the script icon and save to disk.

If you have any problems opening the embedded scripts in this newsletter please let us know.

[Back to top](#)

[Back to top](#)

[Back to top](#)

Right-click the script icon and save to disk.

If you have any problems opening the embedded scripts in this newsletter please let us know.

The 1401*plus* sequencer has a minimum sequencer clock tick of 50 μ s, meaning that it can complete one instruction in this time, e.g., addition of two variables or reading a voltage from an analog input, so efficient coding and our choice of linear electrical couplings rather than nonlinear synaptic couplings was important- the tradeoff is between model complexity and stability of the dynamic clamp. Briefly, large oscillations on the electrode occur when the sequencer loop runs too slowly. Our most useful sequencer files had ~20 instructions and completed one loop every ~1 ms, reading voltages from each neuron, calculating currents to inject, and writing command voltages to be sent to the current clamp amplifier via the 1401 DAC outputs.

While 1 ms is slow compared to nominal update intervals of 50 to 100 μ s used in contemporary dynamic clamp experiments, we found that 1 ms was sufficiently fast to show a nice effect in our experiments while maintaining stability. This was possible partly because we injected relatively small magnitudes of current and balanced the bridge often. A modern Power1401 interface, which has a minimum sequencer step time of 10 μ s would allow speeding up of sequencer execution by a significant factor.

Fred Sieling
Laboratory for Neuroengineering
Georgia Institute of Technology

User group

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

If you have any comments about the newsletter format and content, or wish to unsubscribe from the mailing list for this newsletter, please notify sales@ced.co.uk.

[Back to top](#)

