

Contents

- [News](#) – Updates & Meetings
- [Scripts Spotlight](#) – Sequencer interaction
- [Spike2](#) – Sequencer sections
- [Spike2 Script](#) – Output spike times
- [Signal](#) – Channel processing with Virtual Channels
- [Signal Script](#) – Text output of idealised trace events
- [Did you know...?](#) – Power1401 ADC channels
- [Recent Questions](#) – Trigger output sequencer from event channel

News

European Training Days 2014

Join us in Cambridge on June 5th - 6th for our annual training days and learn how to use Spike2 and Signal more efficiently to customise and streamline your analysis. [Register](#) on our website to reserve your place on the course.

Future meetings and events

CED European Training Days 2014

St. John's Innovation Centre,
Cambridge,
UK
June 5th – 6th 2014

CED Chinese Training Days

Suzhou City,
Shanghai,
China
June 22nd – 24th 2014

Symposium for Chinese Neuroscientists Worldwide 2014

Suzhou City,
Shanghai,
China
June 24th – 28th 2014

Physiology 2014

Queen Elizabeth II Conference Centre,
London,
UK
June 30th – July 2nd 2014

9th FENS Forum of Neuroscience

MiCo – Milano Congressi,
Milan,
Italy
July 5th – 9th 2014

[Back to top](#)



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	Released	Signal	Released
Spike2 version 8.01c	03/14	Signal version 6.01a	03/14
Spike2 version 7.13	03/14	Signal version 5.10	03/14
Spike2 version 6.17	10/12	Signal version 4.11	02/14
Spike2 demo	12/13	Signal demo	03/14

Scripts Spotlight

The output sequencer in Spike2 is used to generate digital and waveform outputs during sampling. These can be used to trigger and control a wide range of devices such as electrical and magnetic stimulators, motors or visual stimulus generators. Often the researcher needs to be able to set, or change, these stimulus parameters 'on-the-fly' during a recording and, while it is possible to load in replacement sequencer files during sampling, it can be much more convenient to use a script that allows the user to input various stimulus parameters with a dialog. The script can then pass these new values to the sequencer to update the experiment protocol in real time.

As easy as this sounds, many script writers come unstuck when it comes to re-calculating user entered values as sequencer values. The sequencer uses a 'low-level' programming language and expects DAC values to be expressed as 32-bit numbers. For example, if a user has requested a DAC output of 3V in a dialog, this will need converting to a 32-bit value for use by the DAC command in the sequencer.

Luckily there is a script available to make these calculations as stress-free as possible. `Seqlib.s2s` is stored in the `include` folder of your Spike2 and Signal installation and contains a library of functions to help with the calculation of sequencer variables. These include a function to convert user entered units into 32-bit numbers for use by the DAC output commands. This script is intended for use as an include file so you can simply enter `#include <seqlib.s2s>` at the top of your own script.

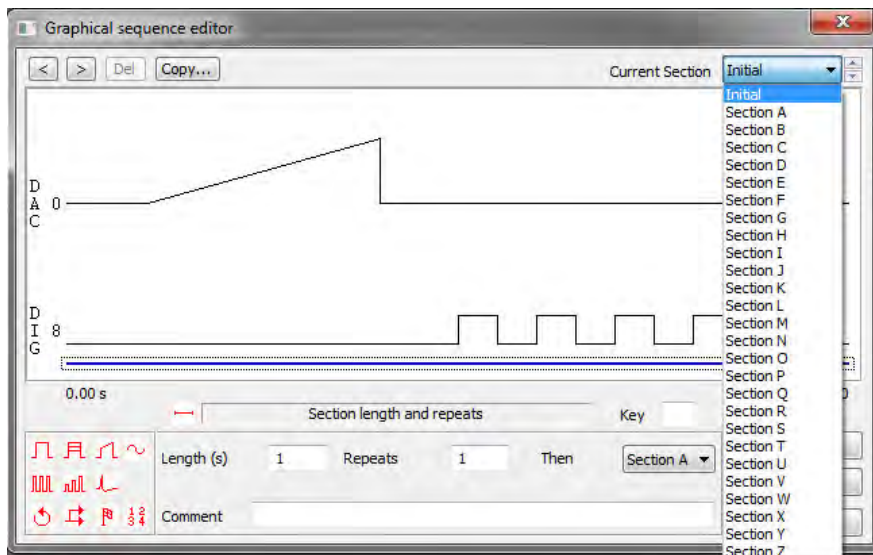
The logo for Spike2, featuring the word "Spike2" in a stylized, cursive font with a red-to-yellow gradient and a drop shadow effect.

- Q. How do I set-up multiple pulse output protocols in the Spike2 sequencer?
- A. The graphical editor gives a user-friendly display of selected outputs for fast and easy creation of stimulus protocols. Pulse outputs can be defined in separate sections in the output sequencer which can then be linked together for automated stimulus sequencing.

[Back to top](#)

[Back to top](#)





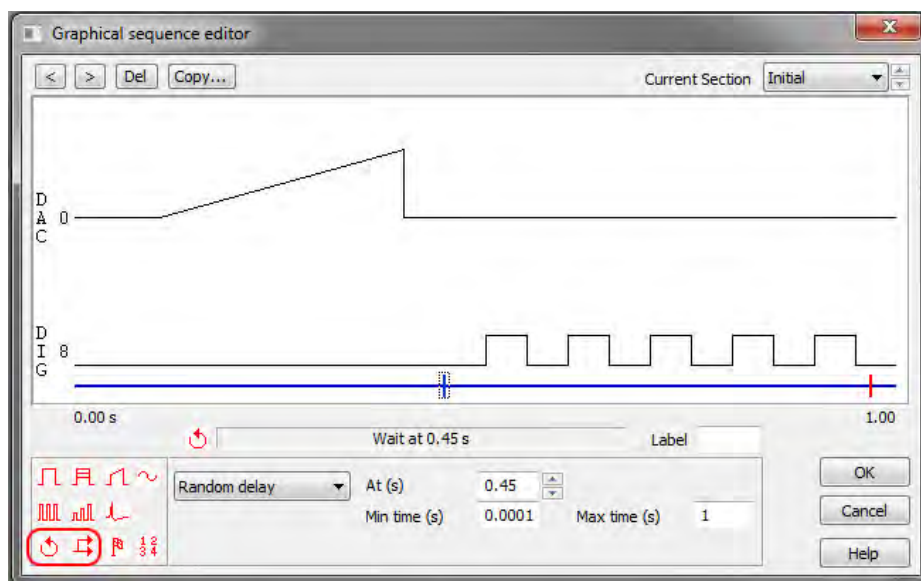
Graphical editor showing available sections

There is a list of up to 27 sections that you can define: the Initial section (which runs automatically when the sequence starts) plus 26 more from Section A, through to Section Z (named Key A to Key Z prior to Spike2 version 8).

The duration of each section is set by clicking on the blue 'control track' at the bottom of the graphical window. Other section parameters include a repeat count, an optional key that can jump control to the section and an action to take when the end of the section (and any repeats) are completed.

If you assign a key to a section, the section can be started by pressing the assigned key on the keyboard or by clicking the corresponding button in the sequencer control toolbar. Sections will run linearly from start to finish, executing the pulse outputs that you have configured in time order, but you can also make use of control statements such as loops and branches to automate the sequencing. These options can be dragged onto the control track and used to pause the section while waiting for a time or condition to be met, or branch to a new section.

Hint: There are limits on what can be achieved with the graphical editor. For example, the graphical editor does not have access to the sequencer table. In most complex cases you can use the graphical editor to generate the majority of the code, then use the text editor for the final stages



Control track showing a loop command and branch as dragged from the palette

Scripts: Spike2

[Back to top](#)

- Q. I would like to be able to play a series of spike times from a pre-recorded data file as a stimulus during sampling.
- A. The Spike2 output sequencer supports a user defined table of 32-bit values that can be used to transfer large quantities of information rapidly between a script and a sequence. The attached script, [PlayTrainTable.s2s](#) and sequence, [PlayTrain.pls](#) is an example of using the table to output a series of pre-recorded events during sampling. Before using the script, load the sequence file into the sampling configuration. Then open a new data file ready for sampling and run the script. It will prompt the user to open a data file containing an event-type channel (Event, Marker or WaveMark) and select a time range of events to write to the sequencer table. During sampling, pressing key 'a' on the keyboard or sequence toolbar will play the copied data out of Digital Output 0 on your Micro1401 or Power1401. The sequencer table is not supported by the 1401 *plus*.

Signal




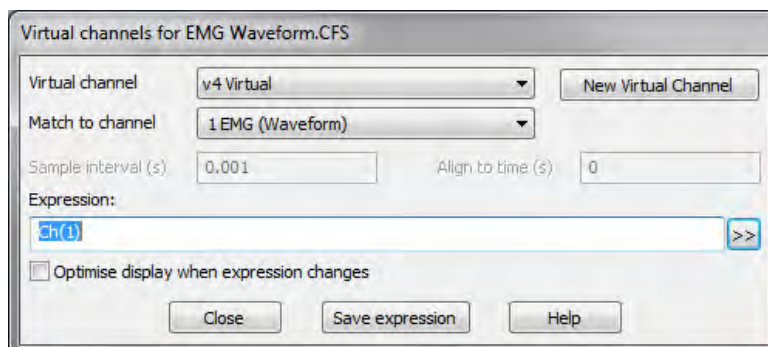
Show me How!

Click here to watch a video on channel processing in Signal

[Back to top](#)

- Q. Is there any way to add channel modifications, such as rectify and smooth, to a copy of an existing channel so that the original data is not permanently modified?
- A. Virtual channels can be used to carry out a wide variety of channel arithmetic tasks (for example, sums and differences of channels), generate new waveforms based on sine, square or triangle envelopes, and add channel processes such as DC offset removal, rectification and smoothing to copies of raw data channels both on-line and off-line. Virtual channels are created from the **Analysis** menu **Virtual channel** command. The new virtual channel requires a sampling rate and can be matched to an existing waveform channel using the drop-down list or you can specify a sample interval and align time.

There are a number of expression functions that can be typed into the **Expression** field of the dialog to generate virtual channels (a full list of these can be found in the on-line help) or you can build expressions using the  button and selecting the operation to perform from the menu. For this example, select **Waveform** from **channel** from the menu and select the channel you want to process from the resultant dialog. This creates the expression $Ch(n)$, where $Ch()$ is the function to copy a waveform channel and n is the channel number. To add a channel process to the channel, select the $Ch(n)$ text in the **Expression** field by clicking and dragging the text caret over the $Ch(n)$ entry.

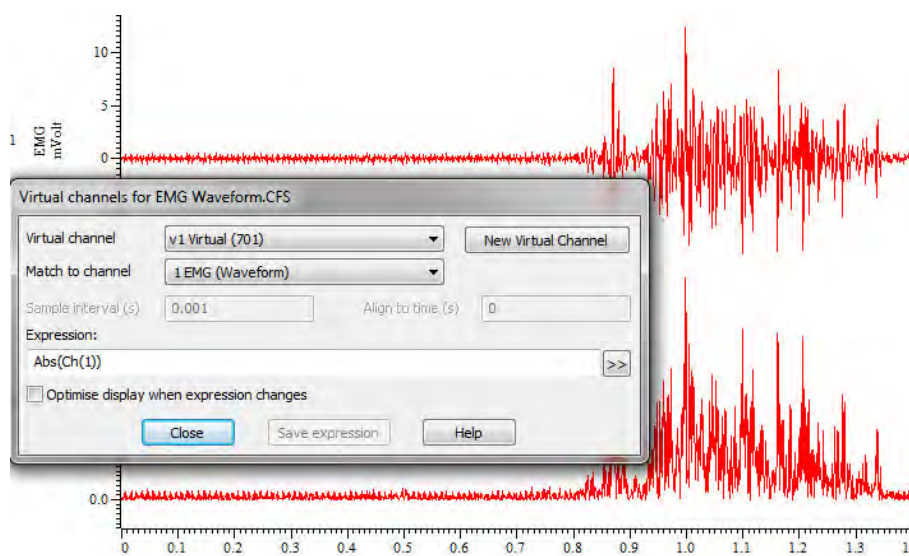


Ch(1) text selected in the Virtual channels dialog

Hint: You can also edit text expressions manually or copy and paste to insert an expression in the dialog



Next, click **>>** and select **Rectify** from the **Channel process functions** sub-menu. This encloses $Ch(n)$ within the $Abs()$ expression, creating $Abs(Ch(n))$ in the dialog and generating the result in the virtual channel displayed in the data file.



Virtual channel showing rectified EMG data from channel 1

Channel process functions can be applied in sequence. For example, to add a 3 point smooth to the rectified data, select the $Abs(Ch(n))$ text in the dialog and click on the 3 point smooth option from the **Channel process functions** menu. This creates the expression $Smth3(Abs(Ch(1)))$ which applies a 3 point smooth to the rectified EMG data.

Scripts: **Signal**

- Q. I record single channel patch clamp data and use the SCAN method in Signal to generate idealised traces. I export the idealised trace as text which gives me most of the required information including a flag value, however I would like to know what these flags represent (for example, open time, closed time, first latency etc.).
- A. The attached script [OpClTextOutput.sgs](#) prints a text file which lists the event times and flag value representations for each event. Open a Signal data file containing idealised trace data then run the script. It currently prints all of the data for each idealised trace frame in the file, but could easily be modified to allow you to select a time range if required.

Did you know...?

All Power1401 models have 16 waveform input channels as standard. Of these, 8 are available as BNC inputs on the front panel with the further 8 available as pin inputs on the rear analogue expansion port. We supply patch panels and break-out boxes that allow connection to the rear input ports via BNC connections if required, contact us for details.

Recent Questions

- Q. I have used the graphical editor in Spike2 to setup a sinewave output and now I would like this to be triggered in response to an event input during a recording.

[Back to top](#)

[Back to top](#)

[Back to top](#)

- A. The output sequencer has a number of commands that can be used to control sequencing and how it reacts to incoming data including external triggers. The bottom row of icons in the graphical editor 'pulse palette' include loop and branch commands. These can be dragged onto the blue 'control track' in the main window. The loop command can be used to generate random wait times or monitor channel data until a certain condition is met. When a loop is selected on the control track, select `Next event` from the drop-down list and then set the `Channel` field to the event channel being used to record the trigger pulse.

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, or wish to unsubscribe from the mailing list for this newsletter, please notify sales@ced.co.uk.

