CED eNewsletter

Cambridge Electronic Design Limited

#83 ~ February 2014

Contents

News

Future meetings and events

Back to top

Biophysical Society Annual Meeting 2014
Moscone Center
San Francisco, CA
USA
February 15th – 19th 2014

Society for The Neural Control of Movement Annual Meeting 2014
NH Grand Hotel Krasnapolsky
Amsterdam
The Netherlands
April 21st – 25th 2014

Experimental Biology 2014
San Diego Convention Center
San Diego, CA
USA
April 26th – 30th 2014

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 8.00a	12/13	Signal version 6	12/13
Spike2 version 7.12c	09/13	Signal version 5.09	05/13
Spike2 version 6.17	10/12	Signal version 4.10	10/12
Spike2 demo	12/13	Signal demo	12/13

Scripts Spotlight

Back to top

In the last few spotlight features we have looked at examples of how to use the `View()` family of functions to locate, show and keep track of any open files that the script writer needs to work with. Often you will also need a way to get values or arrays of data from the channels themselves, especially if you are writing custom analysis routines. For Signal data files and memory views, and Spike2 result views, this is easily achieved using the:

```
View(vh%, c%).[];
```

construct, which gives you access to the channel data as an array where `vh%` is the view handle of the file and `c%` is the channel of interest.

For Spike2 data files however there is a dedicated command that will fill an array with data values from a waveform or RealWave channel, or with event times from all other channel types:

```
ChanData(chan%, arr%[], sTime, eTime);
```

For this command `chan%` is the channel to read data from and `arr[]` is either a real or integer array. Real arrays collect the waveform values in user units as set by the y-axis, so you would most often use a real array here. `sTime` and `eTime` set the start and end times to collect the data from in seconds.

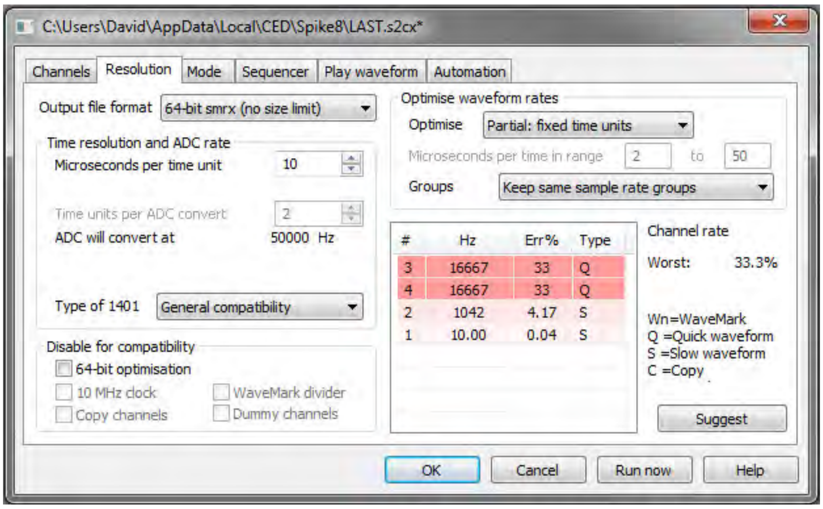
Back to top

Show me How!

Click here to watch a video on optimising sample rates in Spike2

Q. I'm new to Spike2 and trying to set up a configuration for sampling four waveform channels with my Micro1401-3. I've set 25kHz as the ideal sample rate for two of my channels, but when I click OK, the configuration only shows an ADC rate of just over 16.6kHz.

A. The actual sampling rates achievable for waveform channels depends on your time resolution and ADC rate settings. The `Resolution` tab in the sampling configuration contains a number of options that you can use to optimise these settings and get as close as possible to the sample rates you require.



C:\Users\David\AppData\Local\CED\Spike8\LAST.s2cx*

Output file format: 64-bit smrx (no size limit)

Time resolution and ADC rate
Microseconds per time unit: 10
Time units per ADC convert: 2
ADC will convert at 50000 Hz

Type of 1401: General compatibility

Disable for compatibility
□ 64-bit optimisation
□ 10 MHz clock
□ WaveMark divider
□ Copy channels
□ Dummy channels

Optimise waveform rates
Optimise: Partial: fixed time units
Microseconds per time in range 2 to 50
Groups: Keep same sample rate groups

#	Hz	Err%	Type
3	16667	33	Q
4	16667	33	Q
2	1042	4.17	S
1	10.00	0.04	S

Channel rate
Worst: 33.3%

Wn=WaveMark
Q =Quick waveform
S =Slow waveform
C =Copy

Times measured by Spike2 are relative to a master clock in the 1401, and the `Microseconds per time unit` field sets this clock tick period. If you are using version 8 and recording to a 64-bit file, you can usually set this to 1, giving a wide choice of sample rates and no limitations on run time.

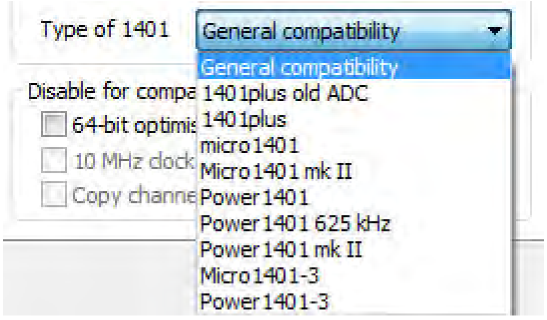
If you are recording to the old 32-bit format, you have to balance the desire for timing accuracy against how long you need to sample for. With a 32-bit file, the maximum sample time is just over 2.1 billion (2,147,483,647) ticks of the

`Microseconds per time unit`. The `Longest run time` field displays the maximum file duration, based on the current time units. If this period is shorter or longer than you require for your work you can adjust `Microseconds per time unit` to give you at least the duration that you need.

There is also a file length limit for 64-bit files but it is so large that it is of no practical restriction; With 1 microsecond resolution you could sample to a single data file for 256,000 years!

Type of 1401

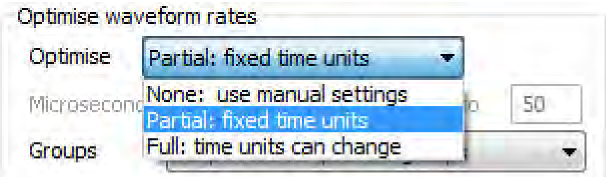
Members of the 1401 family have different capabilities, so you would generally select the model that you are using to sample, here. Spike2 version 8 no longer supports the 1401*plus* or the original micro1401 for sampling, but they are included in case you need to match previous configuration settings.


Type of 1401: General compatibility

If you replace a 1401 with a more recent model or upgrade Spike2 from before version 6.05 you may get different waveform and WaveMark sampling rates. The new rates will be closer to the requested rates, but if you are half way through a study, compatibility with earlier data may be more important. This section of the dialog disables new features so you can match the original rates.

Optimise waveform rates

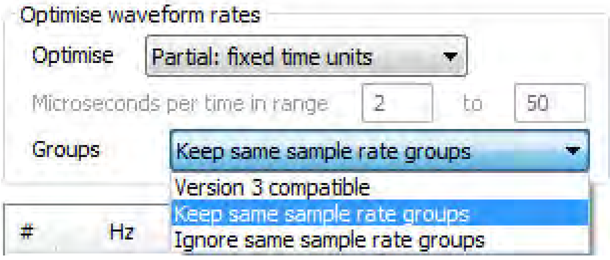
The `Optimise` drop-down sets the parameters Spike2 changes to minimise sample-rate error. With the default setting of `Partial: fixed time units`, you set the `Microseconds per time unit` and Spike2 adjusts the `Time units per ADC convert` setting to minimise the sampling rate errors. If there is more than one solution, Spike2 chooses the one with the slowest ADC convert rate.


Optimise waveform rates
Optimise: Partial: fixed time units
None: use manual settings
Partial: fixed time units
Full: time units can change
Microsecond
Groups
50

With `Full: time units can change`, Spike2 sets both the fields to give the best overall settings for your combination of channels. `None: use manual settings` gives you control over both the values in the Microseconds per time unit and Time units per ADC convert fields. `Time units per ADC convert` sets the ADC clock interval in the units set by `Microseconds per time unit`. Each time the ADC is clocked either one sample is taken (non-burst mode) or a group of samples is taken (burst mode).

The `ADC will convert at` field shows the aggregate sample rate in Hz. This is then split between the waveform channels to get as close as possible to the requested rates.In this mode, click the `Suggest` button to change the fields to the values that minimise sample rate errors.

The `Groups` drop-down places additional restrictions on how Spike2 maps the requested sample rates for all the waveform and WaveMark channels into achievable sampling patterns.



Optimise waveform rates

Optimise: Partial: fixed time units

Microseconds per time in range 2 to 50

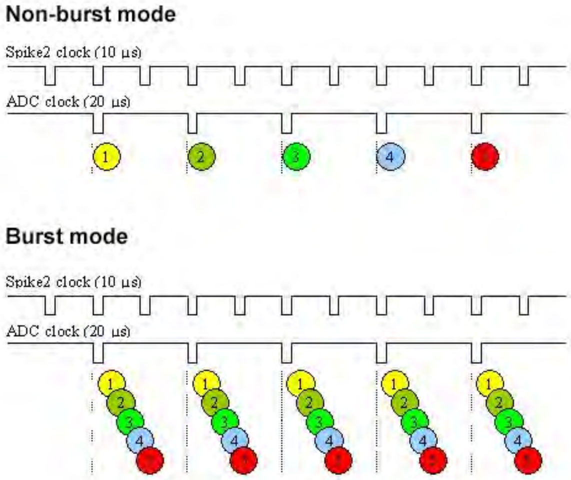
Groups: Keep same sample rate groups

- Version 3 compatible
- Keep same sample rate groups
- Ignore same sample rate groups

`Keep same sample rate groups` makes sure that all waveforms channels with the same ideal sampling rate have the same actual rate. You will normally use this option. `Ignore same sample rate groups` gives the smallest sampling rate errors, but channels with the same ideal sampling rates may get actual sampling rates that are different. `Version 3 compatible` should only be used if you need to match sampling rates to old files recorded with Spike2 version 3.

Burst mode

Burst mode sampling can be useful when you sample more than 1 waveform or WaveMark channel. If you have *n* waveform and WaveMark channels, You may be able to run the Spike2 clock (set by Microseconds per time unit) *n* times slower. This allows longer sample durations if you sample to a 32-bit file (not a consideration for 64-bit files). You may also be able to achieve sample rates per channel that are *n* times higher for the same Spike2 clock rate.
The disadvantage of burst mode is that waveforms are sampled at times that may fall between Spike2 clock ticks. Channels are sampled at precisely the correct interval, but each channel is shifted sideways by up to half a Spike2 clock tick from when it was sampled.



In many applications this will not matter, and the benefits of a better sampling rate may outweigh this. In version 8, where you can have a very small clock tick, this may not matter at all. Burst mode is most effective when you have a large number of waveform and WaveMark channels.

Scripts: Spike2

Back to top

Q. I am recording EMG data for force feedback using Spike2 and would like to know if there is any way to have the incoming data displayed in the centre of the x-axis time range during a recording. This is important for me as it is necessary for the subject to be able to see 'ahead' of the current time to clearly know when to start responding.

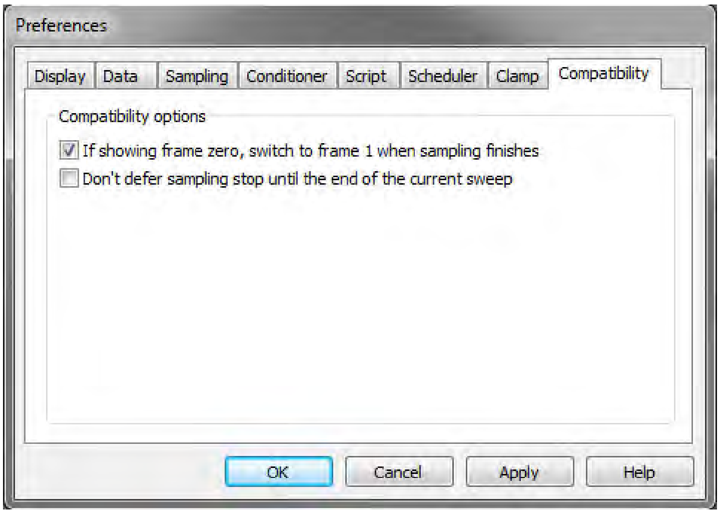
A. The attached example script, `SampleWindow.s2s`, utilises an idle routine to display the incoming data in the middle of the X-axis time range. The user can specify the length of the displayed time range before sampling. Although the script works well when displaying a few channels at relatively low sample rates, such as EMG recordings, the smoothness of the scrolling display will be affected by an increase in the number of channels and the sample rate. The exact effects will depend on your PC's hardware configuration and graphics card.

Signal

Back to top

Q. When sampling with Signal, the data file always displays the last recorded frame when sampling is stopped. I would much prefer that the file displayed the first saved frame by default.

A. The `Edit` menu `Preferences` dialog in Signal is a tabbed dialog that includes a range of different options for the display, sampling and export of data. The `Compatibility` tab of the dialog contains a checkbox to set the option of displaying Frame 1, rather than Frame 0, once sampling is stopped.



Compatibility options

☑ If showing frame zero, switch to frame 1 when sampling finishes

□ Don't defer sampling stop until the end of the current sweep

Signal Preferences dialog

Scripts: **Signal**

Back to top

Q. I would like to export selected frames of a file as images to use in publications.

A. The script, `ExportMetaFrames.sgs`, will export a range of selected frames from a data file as individual images in the Windows metafile (.wmf) format. These new .wmf files are saved to the same location as the selected data file.

Back to top

Did you know...?

Modern 1401 interfaces such as the Micro-3, Power1401-3 and Power1401 mkII, can be set to use +/-5V or +/-10V inputs and outputs via the `Try1401` diagnostic software installed in the Signal and Spike folders. With a 1401 connected and switched on, run the Try1401 application and select `1401 options` from the `File` menu pulldown. This opens a dialog where you can select the input/output range to set for the 1401.

Back to top

Recent Questions

Q. I have some data collected by a colleague using a third party acquisition system and would like to analyse it in Spike2 using waveform correlation, but the channels I want to analyse have been sampled at different rates.

A. In Spike2 you can resample waveforms to user-defined sample rates with the `Interpolate` process, or match the sampling rate of another channel in Spike2 using the `Match to channel` function from the `Channel Process` dialog, which is available from the `Analysis` menu. One example of the use of these functions would be waveform correlation analysis between two channels initially sampled at different rates.

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.