CED eNews

The eNewsletter from Cambridge Electronic Design

#8 ~ August 2004

Established 1970

Contents

News

The latest version of Spike2 is **5.07**, which is available for registered v5 users. This revision includes Virtual Channels which allow you to generate waveforms that are algebraic functions of existing channels (for example differences, products and ratios), and waveforms from event channels. See the Spike2 feature in this newsletter for more information about Virtual Channels.

A full list of new features is available from the updates section of the website.

A demonstration version of the current software is also available.

Version **4.20** of Spike2 is downloadable for registered v4 users.

Version **2.15** of Signal is downloadable for registered v2 users.

Version **3** of Signal is nearing release. This new version will include tools for single channel Patch clamp analysis, direct access to the text sequencer for maximum flexibility of pulse outputs and many other new features. See the Signal feature in this newsletter for further information about Signal version 3.

Our pre-Neuroscience training days this year will take place on Thursday 21st and Friday 22nd of October in San Diego. The venue has yet to be confirmed, but is likely to be the US Grant Hotel, near the convention centre. Following feedback from our recent UK and US training days, the program of events has been revised to provide more 'Workshop' sessions which will allow us to focus more on individual problems or queries, and users to bring examples of their own data to the sessions. Full details of the training day sessions and a booking form are available. Please feel free to contact us with any suggestions or comments about the revised timetable.

We have had a number of requests for a more 'up-to-date' script to perform analysis of EEG data, this project has been started as a script for manual sleep-scoring, but we hope to implement a number of other features common to EEG research and analysis. If you think that an EEG script would be useful for your research, please feel free to contact us with any suggestions or requirements that you would like to see included.

Spike2

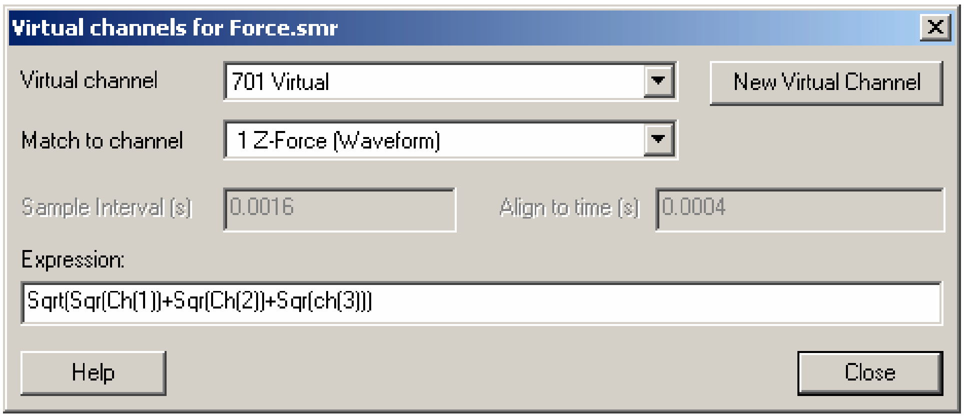
Q. I would like to be able to perform channel arithmetic between multiple channels in a data file, is there a script available for this? Ideally I would like to be able to do this both on and off-line.

A. Virtual channels are new in Spike2 5.07 and are created as arithmetic combinations of waveform, event or RealWave channels, defined by a user-supplied expression. Virtual channels can be used off-line or during sampling, and will update dynamically from the sampled data based on any supplied expression. You can match the sample interval and data alignment of the virtual channels to an existing channel, or specify your own settings. The menu command to create new Virtual Channels or edit existing settings is available from the Analysis menu pull-down.

The Expression field is used to set the expression that defines the virtual channel. Expressions are composed from a combination of numbers, arithmetic operators and channel data. Waveform channel data is accessed from the special function **`Ch(n)`**, where **`n`** is a waveform or RealWave channel. This expression returns the data from channel `n`.

As a simple example, if channels 1 and 2 hold waveform or RealWave data, then the virtual channel expression **`Ch(2) - Ch(1)`** displays the difference between channel 2 and channel 1.

For more complex calculations, Virtual channel expressions can make use of the special mathematical commands, **`Abs()`**, **`Sqr()`** and **`Sqrt()`** which will calculate the absolute value, square and square root of supplied data.



Virtual channels for Force.smr

Virtual channel: 701 Virtual

Match to channel: 1 Z-Force (Waveform)

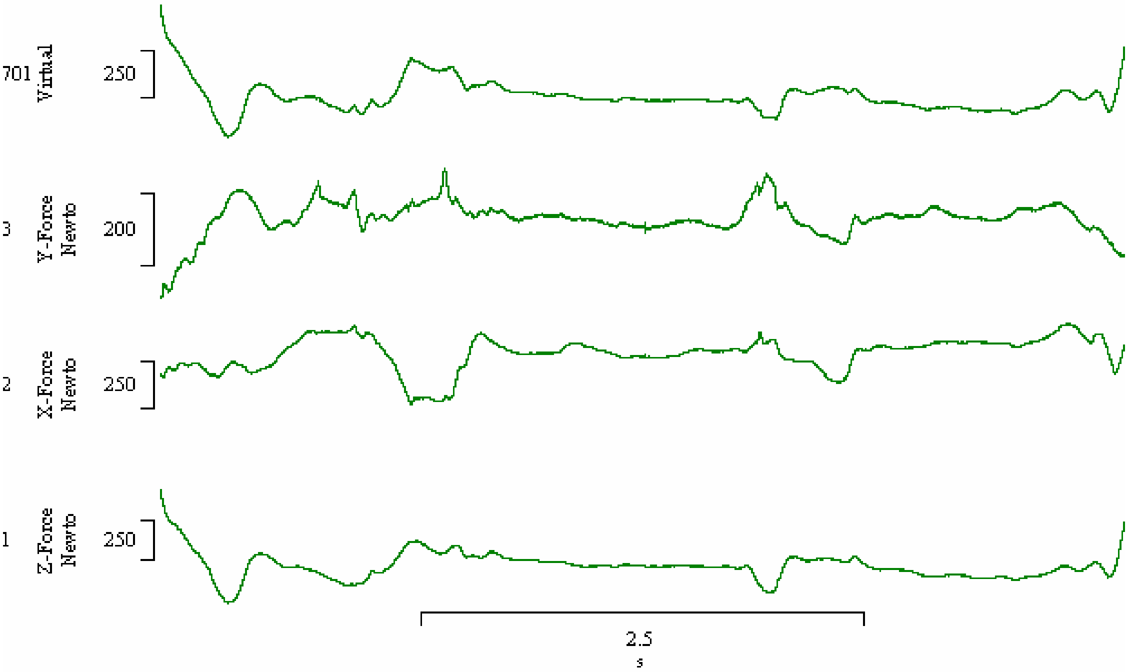
Sample Interval (s): 0.0016

Align to time (s): 0.0004

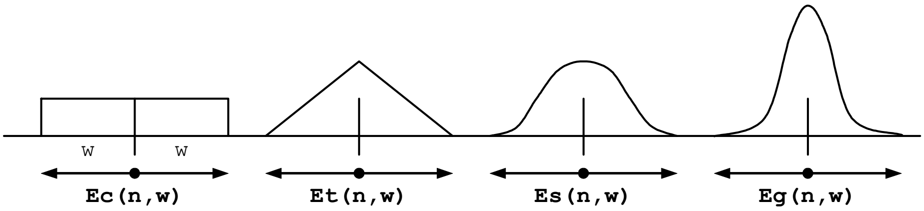
Expression:

```
Sqrt(Sqr(Ch(1))+Sqr(Ch(2))+Sqr(ch(3)))
```

The expression in this example calculates the square root of the sum of squares of the 3 channels, to display the magnitude of 3 perpendicular forces in virtual channel 701, below.



Virtual channels can also be used to convert an event channel into a waveform using the **`Ec(n,w)`**, **`Et(n,w)`**, **`Es(n,w)`** and **`Eg(n,w)`** functions, where **`n`** is the event channel to use and **`w`** is the smoothing period in seconds. The conversion replaces each event by a symmetrical shape, centred on the event time. For an event at time `t`, the shape extends from `t-w` to `t+w` seconds. The shape used is defined by the function; `Ec()` simply counts the events in the time range, the `Et()` function weights the events with a triangle function, the `Es()` function weights the events with a raised cosine and the `Eg()` function uses a Gaussian curve extending to 4 sigmas.



There are also new additions to the `EventToWaveform()` script command which allow you to define your own weighting functions, such as asymmetric shapes. This has been asked for by a number of users to analyse spike density function.

Signal

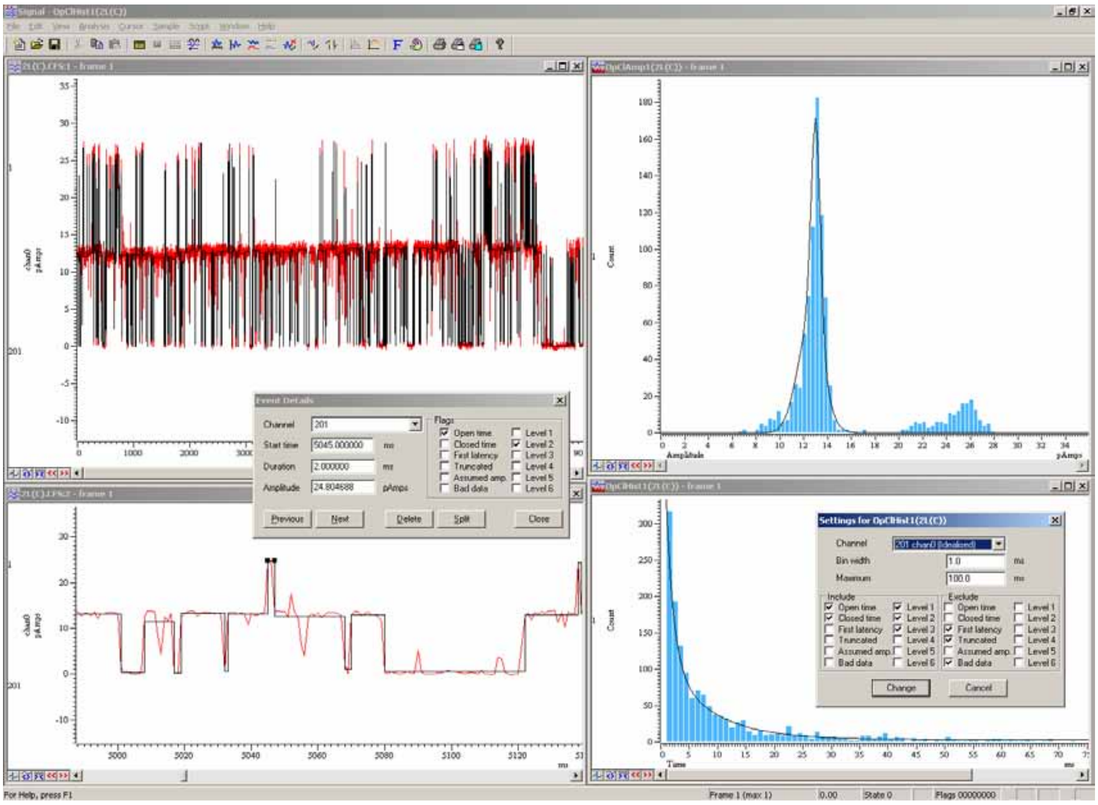
Signal version 3 is now in the final stages of testing, and will be a major upgrade improving and expanding on existing version 2 functions and adding many new features. These will include:

Sampling

- Direct access to the text sequencer for pulses, giving maximum flexibility and control of pulse and waveform outputs while sampling.
- Enhanced amplifier telegraph data support and usability
- External convert sampling input for arbitrary or varying sample rates
- Optional trigger polarity inversion

Analysis

- Single channel patch clamp analysis generates idealized current trace
- Analysis and editing of idealized traces, open/closed times and amplitudes. Import idealized traces generated with DOS Patch software
- Copy and paste data to and from XY views
- Improved curve fitting: polynomial, exponential, sine and Gaussian functions for fitting to single or multiple frame data
- Matrix functions built into script language
- Enhanced auto-average measurements and improved amplitude histogram analysis



Event Details

Channel: 201

Start time: 5045.000000 ms

Duration: 2.000000 ms

Amplitude: 24.804688 pAmps

Flags

- ☑ Open time
- □ Closed time
- □ First latency
- □ Truncated
- □ Assumed amp.
- □ Bad data
- □ Level 1
- ☑ Level 2
- □ Level 3
- □ Level 4
- □ Level 5
- □ Level 6

Settings for OpClHist1(2L(C))

Channel: 201 chan0 (idealized)

Bin width: 1.0 ms

Maximum: 100.0 ms

Include

- ☑ Open time
- ☑ Closed time
- □ First latency
- □ Truncated
- □ Assumed amp.
- □ Bad data
- ☑ Level 1
- ☑ Level 2
- ☑ Level 3
- □ Level 4
- □ Level 5
- □ Level 6

Exclude

- □ Open time
- □ Closed time
- ☑ First latency
- ☑ Truncated
- □ Assumed amp.
- ☑ Bad data
- □ Level 1
- □ Level 2
- □ Level 3
- □ Level 4
- □ Level 5
- □ Level 6

Patch clamp analysis features in Signal version 3

Control

- Multiple monitor support
- Channel information dialogs
- Overdrawn channels with optional locked Y axis and offset
- Cubic-spline draw mode for result views
- New Window dialog for easy selection and clean-up of files

This new version of Signal will be available to current users at a discount over the new price, and future revisions of Signal version 3 will be freely available to download for all registered version 3 users. Please contact us if you would like further details.

Scripts: Spike2 *(Double click icon and save embedded script file to disk)*

Q. It would be useful to have a skeleton script that opens a series of files one after another to perform user defined analysis functions on each file; applying channel processes to multiple files for instance. Is there a script available that would do this?

A. The attached script, `BatchProcess.s2s` is an example script that simply attempts to open each data file in a directory in succession, and show all channels and data in the file. It then closes the file and moves on to the next after displaying a message. This could easily be modified to apply channel processes or generate result view data from each file in the directory.

If your directory contains a lot of files, this can take an appreciable amount of time. You can break out of the script by pressing `Esc`, or supply a user interact command in place of the message, to allow you to stop analysis after each file. The script works with Spike2 version 4 and version 5.

Scripts: Signal *(Double click icon and save embedded script file to disk)*

Q. I would like to be able to create a duplicate view or channel from my data file holding modified data, for example, rectified or smoothed, alongside the raw data.

A. The attached script, `ProcessedView.sgs` is another example script. This prompts you to open a data file and then creates a duplicate holding the rectified data, and displays it alongside the raw data file. This could be modified to perform a number of the Modify Channel commands in place of rectification in the script. The script works with Signal version 2.

Did you know…?

You can now take measurements quickly from a data file by using the interactive measurement feature in Spike2. Simply hold down `Alt` and left click, then drag the cursor. The measurements show the values that correspond to the X axis units of the data file, and the Y axis units of the channel being measured.

This can be useful for estimating level settings when setting up peak and trough searches for the Active Cursor modes, or extracting events from waveforms using the memory channels. The interactive measurements feature is available in version 5.

Recent questions

Q. I am writing a script that takes various measurements from a data file using cursors positioned by the user, and writes these values to a text file. I have a level channel in the data file representing a light being switched on or off. I need to know the value of the level channel at the time of the positioned cursor (either 1 or 0), but `ChanValue()` only returns the time of the next event in the channel.

A. The `LastTime()` and `NextTime()` script commands can be used to find the transition (low to high, or high to low) of a level channel at the time of an event using the optional `&val` argument. This returns 0 if the transition is low to high, and 1 if the transition is high to low. One way to find the level value at a Cursor position is to use `LastTime()` to find the event preceding the cursor, and record the transition. If the transition is low to high (0), this tells us that the level at `Cursor(n)` is high (1), and vice versa. The following code will find the event preceding `Cursor(1)` on channel 2 (our theoretical level channel) and return the transition (or a level if there is no transition), then assign the level value at `Cursor(1)` position to the integer variable `level%.`

```
Var value;    'variable to hold level data
Var level%;   'variable to display level

'Find the event before Cursor(1)
if LastTime(2,Cursor(1),value) > 0 then 'If an event, record transition
   level%:= not value;     'Convert value to return level at Cursor(1)
else
   level%:= value;         'Otherwise, record value at time
endif;
Message("Level at Cursor(1) is %d",level%);    'Display as a message
```

User Group

You can now find our eNewsletters archived on our website.

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