

## Contents

### Welcome

### Training days

### Latest software

### Here to help

### Script Spotlight

- BinSize()



- Cursors
- Script – Sleep Spindles

## Signal

- XY views from cursor feature detection

### Scripter's corner

- XY Views

### Recent questions

- Import axon files

### Contact Us

## Welcome

Thank you for downloading our latest newsletter. This month brings us to the December edition of our newsletter and we can start the countdown for 2022!

Joe Kenyon joined Cambridge Electronic Design in October 2021. He is the newest member of our software team, and he comes to us from UEA. Joe will be bringing all his knowledge gained over the last years to CED, to pursue even greater experiences with our professional software team.

Versions 10.12a, 9.15 and 8.23 of Spike2 are now available to download. For more information visit our [website](#).

If you wish to change to the latest version, please email us at [info@ced.co.uk](mailto:info@ced.co.uk) or call us at **+44 (0) 1223 420186**, to upgrade to version 10!

On Monday 8<sup>th</sup> of November, **SfN 2021** took place virtually! It was lovely to see some familiar names, but also some new ones visiting our 'booth'. We hope you found all the information that you needed during the meeting, if you would like any more information, please feel to contact us at [Marjorie@ced.co.uk](mailto:Marjorie@ced.co.uk)

## Training

Full recordings from this year's earlier online training sessions may be viewed on our [website](#). We have more sessions planned for later, for which details will be shared nearer the time.

We also offer remote training sessions by Skype/Zoom/Teams 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 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: [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

<u>Spike2</u>	<u>Released</u>	<u>Signal</u>	<u>Released</u>
<a href="#">Version 10.12a</a>	10/2021	<a href="#">Version 7.06</a>	04/2021
<a href="#">Version 9.15</a>	10/2021	<a href="#">Version 6.06</a>	04/2021
<a href="#">Version 8.23</a>	10/2021	<a href="#">Version 5.12a</a>	02/2018
<a href="#">Demo</a>	10/2021	<a href="#">Demo</a>	04/2021

[Back to contents](#)

### Here to help

We know access to laboratories has been erratic for the 18 months, but with the easing of lockdown measures we hope that conditions will continue to improve. We are now able to offer site visits when necessary, however we will continue to support remotely in the first instance. CED will also continue to do all in our power to support you for increased home working. Should you require any help or wish to discuss your system, email [Marjorie@ced.co.uk](mailto:Marjorie@ced.co.uk) and we can arrange for a video call via Skype/Zoom/Teams.

We also have tutorial videos for both Spike2 and Signal available on our [website](#) for you to peruse at your leisure. There are new videos and updates in the pipeline. If you have a particular topic you think could benefit from a tutorial video, please let us know. All these videos are also available through our YouTube channels: [Spike2](#) / [Signal](#).

Try the [CED Forums](#) bulletin board for further software and hardware support. Ask your questions and receive valuable input from our super users.

[Back to contents](#)

### Script Spotlight – BinSize()

The script command `BinSize({chan%})` is used to obtain information on a file's time resolution or a channel's sampling interval.

- In a time view, the sampling interval between points is returned for Waveform, WaveMark and RealWave channels or a negative number if the channel does not exist. For event channels, the command returns the underlying time resolution. If you omit the `chan%` argument, the function also returns the file time resolution in seconds.
- In a result view, `chan%` is ignored, and should be omitted. The command instead returns the x axis increment per bin.

`BinSize()` is used in a variety of ways, for example use the calculation `1/BinSize(chan%)` to return the sampling rate of a channel. In a result view, `MaxTime()` returns the number of bins, therefore the time width of the histogram can be calculated with `MaxTime()*BinSize()`.

[Back to contents](#)

The logo for Spike2, featuring the word "Spike2" in a stylized, orange, cursive font.

## Cursors

In a time view, vertical cursors can be active or static. An active cursor can seek to a position based on the position of other cursors and data in a channel. They can also automate data analysis, leading to new data channels, XY trend plots or tabulated output. Learning how to use cursors both in static and active mode is imperative to make the most out of Spike2's broad range of analysis functions.

### Static mode

You can add up to 10 vertical cursors to a data or result view with the cursor icon or the cursor pull down menu. Vertical cursors are numbered 0 to 9 and pressing Ctrl+n (where n is a number key between 0 and 9) will add or fetch that numbered vertical cursor within the visible data area. The default setting for cursors is Static, that is, they will not move position unless you move them. Cursors can be labelled in different ways using the Cursor menu Label mode option: number, position, and position and number, or user defined.

The *Cursor* menu > *Display Y values* option displays measurements at the cursor position in a new window. For waveforms, the result is the amplitude of the nearest data point. For event data, the measurement depends on the channel display mode. The cursor times and y values can be set relative to any cursor by checking the Time zero and Y zero boxes respectively and selecting the reference cursor to use with a radio button.

Measurements can be made between pairs of cursors with the *Cursor* menu > *Cursor regions* dialog. A new cursor measurements window is created in which one of three types of measurement can be made between adjacent cursors. These measurements may be made relative to a zero-region defined by checking the Zero region box and selecting a cursor region with radio button.

You can also add up to 9 horizontal cursors to your data or result view. These are numbered 1 to 9. Horizontal cursors are attached to a channel, but can be dragged between channels, as required. They are used to indicate y axis values. Most dialogs in Spike2 will accept cursor positions in place of numerical entries (e.g. Cursor(1) instead of 30s, or HCursor(1) instead of 4.0V).

### Active mode

Vertical cursor 0 is special; we have discussed it several times in previous newsletters for both Spike2 and Signal. It always exists, but it can be hidden. It is the iterator for XY view and data channel measurements; a movement of cursor 0 causes all other active cursors to recalculate their positions.

Cursor 0 can be driven by Spike shape dialogs, by Multimedia replay, Offline Waveform Output, by user dragging and by Measurement processing. The Spike shape dialogs, and Multimedia replay windows can be set to track cursor 0, so changes made by active cursor searches can affect them.

Open the Cursor mode dialog with the *Cursor* menu > *Active cursor modes*. The *Search method* field and the selected cursor determine the dialog contents. An active cursor has an associated *Search channel* and start/end positions that define the data to search to locate the new cursor position. However, Cursor 0 does not have start and end positions. Instead, it has a Minimum step; searches start at this distance from the cursor 0 position and continue to the file end for a forward search, or to the file start for a backward search.

Cursor 0 has a range of search methods: Peak find, Trough find, Rising threshold, Falling threshold, Within thresholds, Outside thresholds, Peak slope, Trough slope, Turning point, Data points and Expression.

The search positions can be a fixed time, but more usually they will be expressions that involve the positions of other active cursors. Active cursor positions are evaluated in sequence from cursor 0 to cursor 9. For cursor n, an expression that refers to an active cursor less than n uses the new position. An expression that refers to a cursor greater than or equal to n uses the old position.

If the End position for search is less than the Start position for search, searches are backwards. If a search is backwards, read previous for next and last for first in the descriptions of the cursor modes. Where search modes depend on data values, for example maximum and minimum, and the data channel does not have a y axis, then the value is taken as the interval between data items. The Position if search fails field is described here.

Several modes use a slope. These modes have the value Width for slope measurement, in seconds and use the data points from Width/2 before the current position to Width/2 after to calculate the slope unless there are more than 200 points, in which case 100 points before and after are used. The contribution of each point to the slope is proportional to the distance of the point from the current position. Because the slope at any point requires data around it, the slope within Width/2 of the ends of the data and any gaps is not to be relied on as the missing data is assumed to be zero.

The ... button next to the cursor number opens the Cursor Label dialog where you can set a user-defined label (sometime useful to make it clear which cursor is in use).

The active cursor modes are:

Static	Peak find	Trough find	Rising threshold	Falling threshold
Threshold	Outside thresholds	Within thresholds	Maximum value	Minimum value
Maximum excursion	Steepest rising	Steepest falling	Steepest slope (+/-)	Slope peak
Slope trough	Repolarisation %	Turning point	Slope%	Data points
Expression				

See the in-software help topic for full details of each mode. The new position is obtained by evaluating the Position field, which normally holds an expression based on cursor positions, for example  $\text{Cursor}(0)+1$  or  $(C0+C1)/2$ .

### Horizontal active cursors

Normally, horizontal cursors are static; they stay where they are put. Using the Active horizontal cursor dialog, horizontal cursors in time views can be made active; they will move to a level found by a measurement (normally taken from the channel on which the cursor is located). This search is repeated whenever cursor 0 is iterated or a vertical cursor is moved on which they depend. This repositioning is normally carried out after all the vertical cursors whose position is used have been repositioned. Active horizontal cursors can be used as a simple way of marking levels within your data; they are also useful for extending the capabilities of analyses that generate measurements from data files.

[Back to contents](#)

# Scripts: Sleep Spindles

Sleep spindles are brief bursts of 12-15Hz oscillations in the EEG during slow wave sleep. They occur in all the mammalian species that have been tested and are thought to be implicated in sensory processing and in the consolidation of long-term memories. Thus, there is significant research interest in the effect of learning, memory retrieval and pharmacological agents on the frequency of sleep spindles.

**Preferences**

Default Y-Axis ranges:

Raw EEG (+ $\mu$ V)

Filtered EEG (+ $\mu$ V)

Power in Band ( $\mu$ V<sup>2</sup>)

Default Time Ranges:

Standard (s)

Zoomed in (s)

Enable Reset button

Our script writers have produced the [SleepSpindle 04.s2s](#) script, available on our website to detect, mark, and analyse sleep spindles in your EEG data.

Sleep spindles are relatively small and somewhat variable and so detecting them automatically is not a trivial task. This script attempts this by detecting bursts of EEG power in the relevant frequency band that exceed a user-defined threshold. These events are then revised to exclude episodes that are shorter than a user-defined duration (usually 0.5s) or to amalgamate bursts that are close together into a single event.

**Results**

Epoch channel

Event marker

Start time

End time

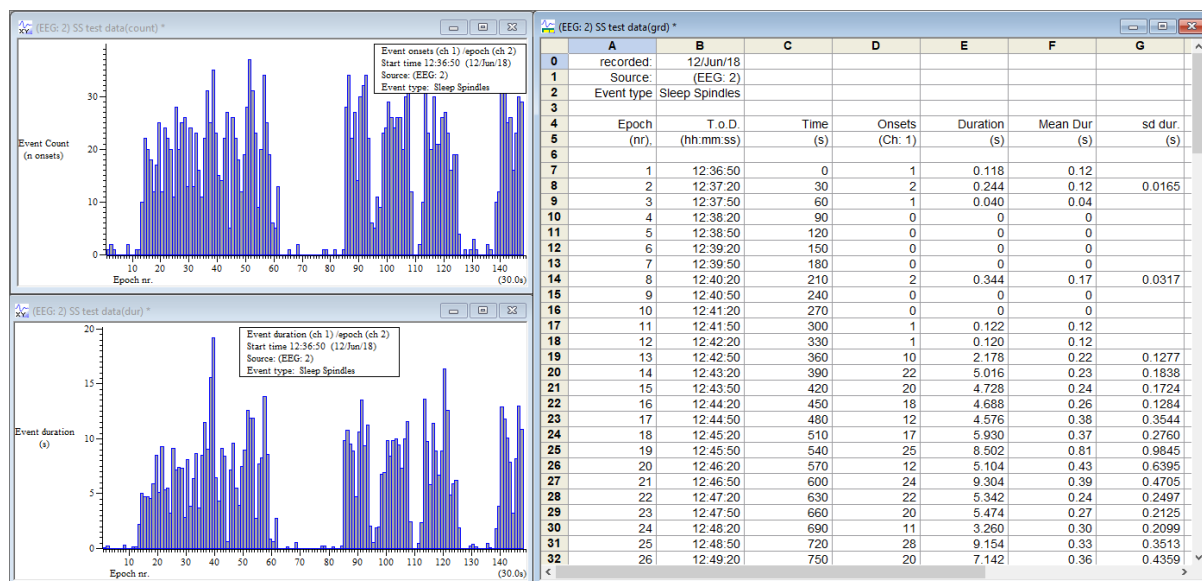
Event Onsets /epoch

Event duration

Tabulate

The script can process multiple EEG channels in a time view, and the features detected are marked as coloured bars in State channels. There are also options to exclude spindle-like events that exceed a set duration or exceed a threshold power level. The flexibility in setting the scripts parameters means that the script could also be useful for other EEG features such as seizures.

The results can also be gated to include only spindle-like events occurring within a specified sleep state. You can also print the results to a spreadsheet or plot spindle count and spindle duration per epoch.



# Signal

You can copy data from an XY view in spreadsheet format and paste it to a text file or directly into a spreadsheet document. With the XY view open, use the *Copy as Text* command from the *Edit* menu. This copies the data to the clipboard from where you can paste it to your text file or spreadsheet document.

The *Copy as Text* command only copies visible data. You can show and hide channels in the view using the *Customise display* option in the *View* menu, and select the time range and Y range to copy by zooming in or out of the data with the mouse

To show all the data and channels to copy, click *All* on in the *Customise display* dialog, then open the *X Axis range* dialog from the *View* menu and click *Show all*. Finally, open the *Y Axis range* dialog from the *View* menu and click *Optimise*.

Text output configuration

Interpolation method: None

	Data	Time	Headings
Waveform	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Errors (SD)	<input type="checkbox"/>		
Marker	<input type="checkbox"/>		<input type="checkbox"/>
Fitted data	<input type="checkbox"/>		<input type="checkbox"/>
Idealised	<input type="checkbox"/>		<input type="checkbox"/>
Real Marker	<input type="checkbox"/>		<input type="checkbox"/>

Include channel units

Decimal places (data, time): 6 8

Field width for all items: 0

String delimiter: "

Separator: Tab

Add header to text block

OK Cancel

You can also copy data views to the clipboard as text. However, text representations of Signal data can be very large and awkward to manipulate with the clipboard; as a better alternative you can write the text output to a file with the *File* menu *Export As* command. The command leads to dialogs that set the text output format and the data to copy:

This dialog sets the text output format. The first section sets an interpolation method for Waveform channels. This is because the Waveform channels will be output in columns with each row showing data sampled at a particular time. If the data is not sampled in burst mode then the data points in the file will not all be sampled at the same time. This will result in a small error as data will appear shifted slightly to bring all the points into line. This can be overcome by using interpolation to estimate where the waveform on a given channel would have been at the time a point on another channel was sampled. Linear and Cubic Spline interpolations are available.

In the next section you can choose whether to output Data, Times values (where appropriate) and Headings for each of Waveform, Errors, Marker, Fitted, Idealised trace and Real Marker data types by checking the relevant boxes. Error data is always the standard deviation and is not generated unless waveform data is dumped.

All output is written in fields that are either numeric or text. A text field is a sequence of characters that may include spaces. Text fields may hold numbers, but numeric fields cannot hold text.

Export from Actions.CFS

Channels: All channels

Start time (s): Mintime()

End time (s): Maxtime()

Frames: Current frame

Force exported data X range to start at zero

Export Cancel

[Back to contents](#)

## Scripter's Corner – XY Views

An XY view is a way of plotting any X value against any other Y value. The points can be plotted using a variety of symbols and colours. The data is divided into channels and the data points within each channel may be joined or unjoined.

An XY view always has at least one data channel; when you create a view you also create a channel. The following script code makes an XY view:

```
'XY.s2s
var xy%; 'handle for the XY view
xy% := FileNew(12,1); 'type 12=XY, 1=make visible now
```

To add additional channels use the XYSetChan() command. You can also use this command to set a channel to a particular state. The following code sets channel 1 to show data points joined by lines with no limit on the number of data points, drawn in the standard colour:

```
XYSetChan(1, 0, 0, 1); 'chan 1, no size limit, no sort, joined
XYDrawMode(1, 2, 0); 'set a marker size of 0 (invisible)
```

To add data points to a channel use the XYAddData() command. You can add single points or pass an array of x and y co-ordinates. The following code adds three points to draw a triangle:

```
XYAddData(1, 0, 0); 'add a point to channel 1 at (0, 0)
XYAddData(1, 1, 0); 'add a point at (1,0)
XYAddData(1, 0.5, 1); 'add a point at (0.5, 1)
```

You will notice that the result of this draws only two sides of a triangle. We could complete the figure by adding an additional data point at (0,0), but it is easier to change the line joining mode to "Looped", and the figure is completed for you:

```
XYJoin(1,2); 'set looped mode
```

In addition to the above commands there are a host of other commands to use with XY views. These are:

- XYColour() Gets or sets the colour of a channel.
- XYCount() Gets the number of data points in a channel.
- XYDelete() Deletes a range of data points or all data points from one channel.
- XYGetData() Gets data points between two indices from a channel.
- XYInCircle() Gets the number of data points inside a circle.
- XYInRect() Returns the number of data points in a channel.
- XYKey() Gets or sets the display mode and positions of the channel key.
- XYRange() Gets the range of data values of a channel or channels.
- XYSize() Gets and sets the limits on the number of data points of a channel.
- XYSort() Gets or sets the sorting method of a channel.

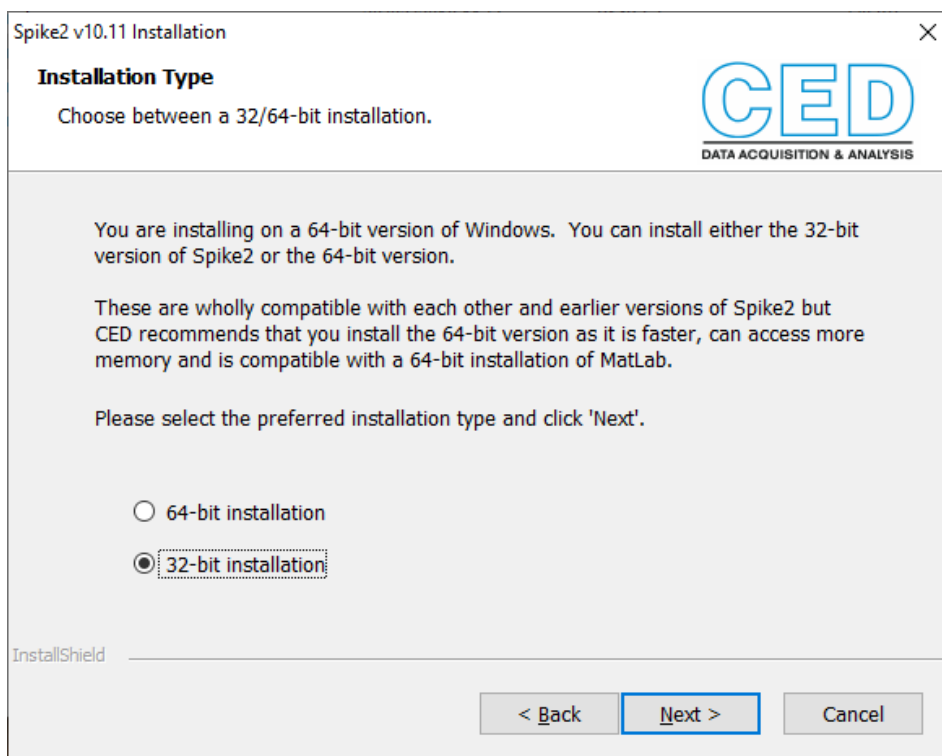
A good example of an XY view is the clock script that ships with both Spike2 and Signal (usually C:\Users\User\Documents\Spike10\Scripts or Signal7\Scripts). In this script the clock face is drawn using a single channel of 12 data points that are not joined and drawn as large solid blocks. Each of the clock hands is drawn using one channel drawn using invisible points that are looped.

[Back to contents](#)

## Recent Questions – *Why can't I import axon files?*

Axon instruments by Molecular Devices have yet to update their file system to 64-bit and are only available in 32-bit format. Therefore, the importer is only available on 32-bit versions of Spike2 and Signal. You can check the details of your installation in the Help > About Spike2/Signal dialog, however if you cannot see the *Axon Instrument (\*.DAT; \*.ABF)* option available in your File menu > Import dialog, then you likely have a 64-bit installation.

To install your software as 32-bit, insert your installation CD or USB drive and during the installation process select a Custom Install. You'll then be asked which version of the software to install:



The only other 32-bit only importer is for files from Data Sciences International.

[Back to contents](#)

## Contact Us

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).

### Contact us:

#### In the UK:

Technical Centre, 139 Cambridge Road,  
Milton, Cambridge, CB24 6AZ, UK  
Telephone: (01223) 420186

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

International Tel: [44] 1223 420186

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 © 2021 Cambridge Electronic Design Ltd, All rights reserved.