## Contents

## News

A little late this second newsletter. ANY comments about format or whether on some emailers the images are not visible would be very helpful. I'm feeling my way with this!

Version **5.01** of Spike2 is now released. A demonstration version of the software is available from www.ced.co.uk/upu.shtml

Version **4.16** of Spike2 is freely downloadable for registered v4 users.
Version **2.13** of Signal is also freely downloadable for registered v2 users.

Version 5
We have been working on curve fitting within Spike2 and should have a 5.02 release soon.

User group. See end of newsletter.

UK training days in Cambridge on Monday June 16th and Tuesday 17th. There will also be the usual USN training days later in the year.

## SPIKE2

Q.    I would like to record EMG activity during a repetitive cycle e.g. rowing or walking and occasionally produce an evoked response to neural stimulation at 15 second intervals. To visualise the data a little better I would like to be able to use Spike2 as an oscilloscope.

A.    The answer is to set up the display with Display Trigger active. This can be found in the view pulldown and opens a dialog detailing the trigger channel to trigger upon or 'No Channel' which would give a traditional ECG/EKG paged scope type display. The Display Trigger is applied to the current window so it is possible to duplicate the data window using the Window pulldown – duplicate function then apply the trigger settings. This way we can have a chart and scope on the screen simultaneously.

Other functions of the dialog include pre-trigger display time and minimum display hold which will 'store' the display even though other triggers are occurring in this time. Cursor(0) can also be set to inactive or to jump to the new data section. This is an aid to taking measurements where other 'Active' cursors will then locate features and produce an XY plot of latencies, amplitudes etc.
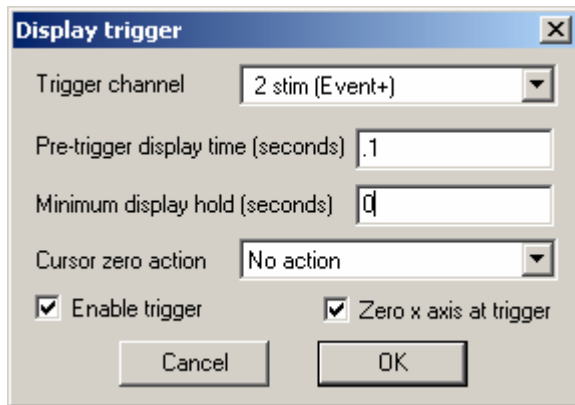
Once the Display Trigger is enabled then the data view contains two more buttons at the bottom left of the window.



These allow you to step left or right through the triggers. To change the x axis width to suit the short period needed for evoked response we should either drag or double click the x axis changing the width to say 800ms.

Stimulation can be either created externally and recorded as an event or generated by Spike2. To generate a pulse use the Sequencer section of the sampling configuration. Here we can either call an existing text sequence file such as demo.pls from the sequence folder of Spike or we can graphically generate the output pulses and load this into the 1401 when acquisition starts.

Choose Use Graphical Sequencer from the drop down list. Uncheck all DAC and Digital output check boxes except bit 8 then hit Graphical Editor. What we should now see is a stretchable dialog showing a 'frame' of output. We are in the 'initial' state which will go active as soon as we start recording and will output a pulse if it exists which may be a bit of a shock! To avoid this we can choose one of another 26 'states' by pulling down the Current Selection. Select 'Key A' . We should have two horizontal bars on the screen one blue and one black. The blue line is our frame section length and by clicking on it we change it's length and whether it repeats. We also give it a 'Key' label in this case uppercase A.
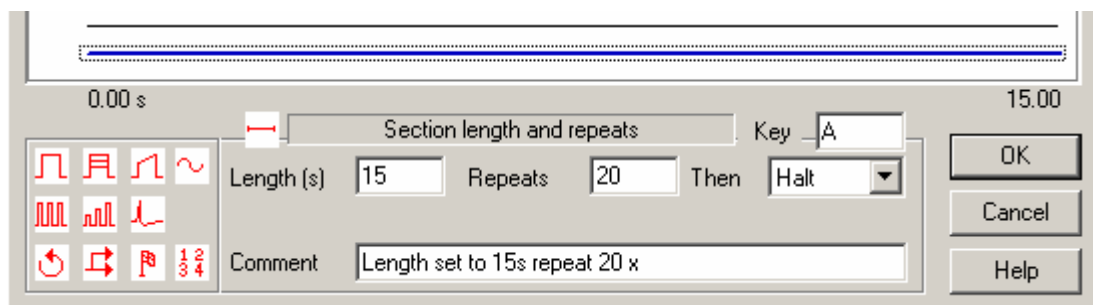


By changing 'Then' from Halt to another key we can produce a more complex protocol.

From the palette drag the simple square pulse onto the black 'output' line. This produces a square pulse of editable duration and start time. If we set the start time to 0 then we have a pulse which by default will be 0.1 s in duration and then 14.9 seconds of inactivity. We can of course change the inactive time to accommodate the pulse width so that we have a true 15 second window.

Once we have designed the output and start sampling we should see the sequencer controls on the screen. Clicking the 'A' button will initiate the output.

This is a v4 or v5 function.

**SIGNAL**

Q      How can I make Signal displays better for publication

A      Signal contains a number of features for customizing the display of data. Since its release you have been able to:

Show and hide specific channels

Set the range displayed in both X and Y axes

Duplicate views in order to show either different frames or time sections simultaneously

Display the data in different draw modes

Through its development many new features have been added, resulting in an extremely flexible display system. Such improvements include:

Far greater control over axis display including tick spacing, display of independent features (eg channel numbers, units etc) and option to display only a scale bar
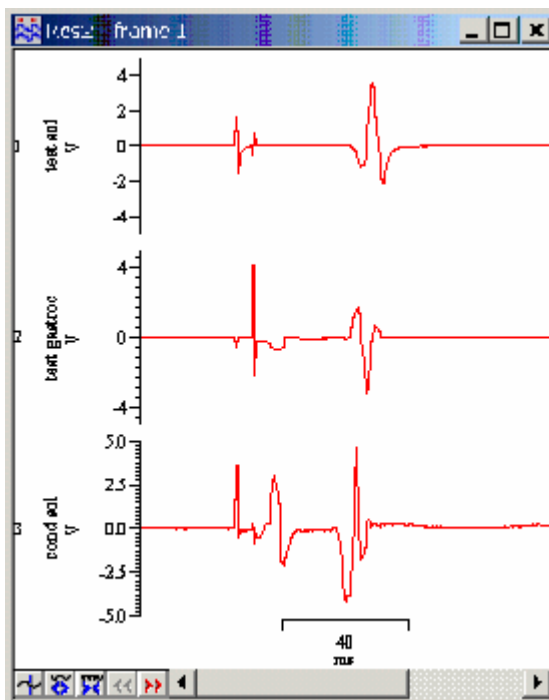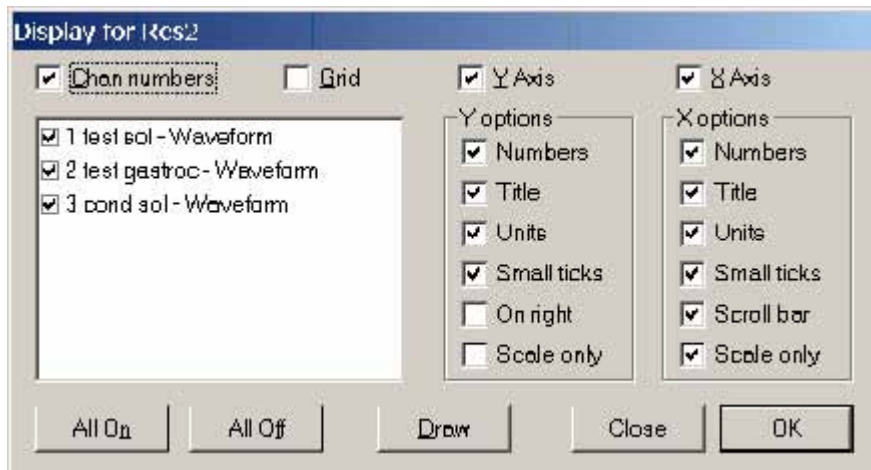
Ability to re-order the channel display. This enables grouping of associated channels and even display of multiple channels on a single axis. Channels can be moved by clicking on and dragging the channel number to the required position and releasing the mouse button. When the mouse pointer indicates there the channel will be positioned. The diagram below demonstrates dragging channel 1 between channels 2 and 3.
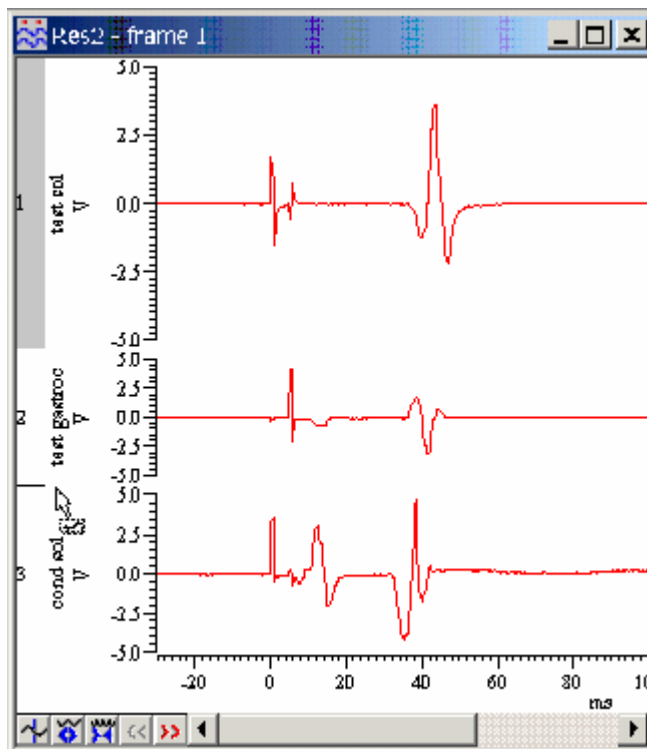


You can change the space allocated to any channel. Hold down Shift and click the mouse between two channels. Keep holding down Shift and move the mouse to change the spacing between the channel and the one above. Hold down Ctrl and release Shift (after clicking) to squeeze/stretch all channels.

Hold down Ctrl+Shift to squeeze/stretch only channels with y axes.
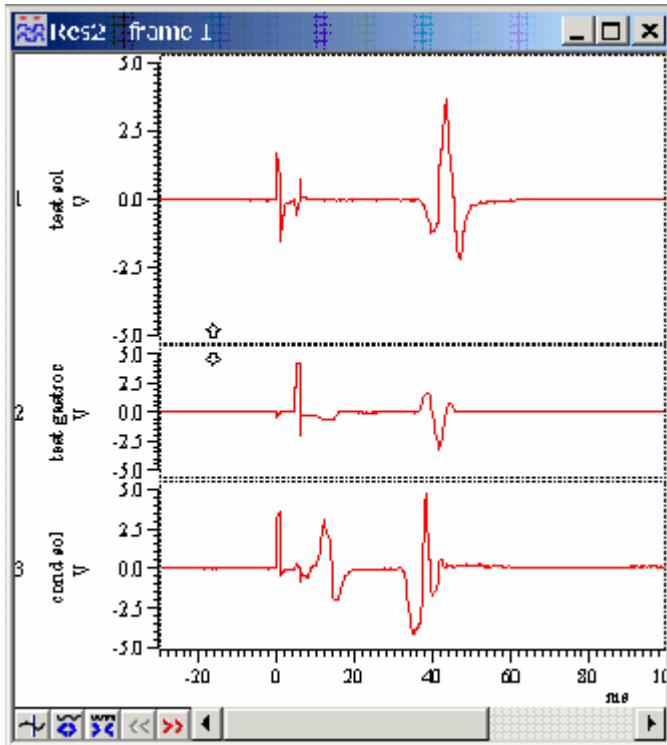
### Scripts. Spike2

This script was written as a simple demonstration of a method to simultaneously rerun a data file and play the sound from one channel through the computer sound card. As there is currently no built-in script mechanism for playing out waveforms off-line the user must first run the script then use the menu driven waveform output facility.

When the script is run it requires the user to open a data file and then set the display width to show during the rerun. When this is done, select O*utput Waveform* from the *Sample* menu. From the waveform output menu set the channel to output and the time range to rerun and ensure *Sound* is selected as the output. When *Play* is pressed, the sound should be played through the system speakers and the data file rerun.

```
**********************************************
var data%;                  'Data file handle
var w;                      'Display width

ToolbarSet(1,"Quit");
ToolbarSet(3,"Open file",Open%);
Toolbar("",1023);

Func Open%()
data% := FileOpen("",0,2);      'Open data file
WindowVisible(1);           'Bring to front
ToolbarSet(0,"",Idle%); 'Start idle routine
return 1;
end;

Func Idle%()
View(data%);                    'Make sure we are looking at data view
w := XHigh()-XLow();    'Get display width
Draw(Cursor(0) - w,w);  'Display with cursor 0 at right hand end of window
return 1;
end;
**********************************************
```

## Scripts Signal

We have been asked to produce a script to help export Signal data. Normally the data would be exported in frames with some arbitrary gap in between. To get around this the script below between the * marks forms a concatenated version of the original file into a new memory view. Effectively this is forming one frame from many

```
***********************************************
'Script to amalgamate all frames of one file into a memory view

Var FrameCt%;      'Number of frames variable
Var List%[32];     '# channels
Var Bins;          'Points per frame
Var Data%;         'Data view handle


Data%:=FrontView();          'Get the name of the current data view
FrameCt%:=Framecount(); 'Get frame count

ChanList(List%[],1);    'Get list of wave channel

BinS:=BinSize(List%[1]);      'Whats the sample rate
BinS:=(Maxtime()+MinTime())/bins;   'Number of points in a frame


Doit%();

Func Doit%();
Var arr[Bins];
Var MoveArr[Bins];
Var Res%;
Var I%;
Var J%;
Var Begin%:=0;

Res%:=SetMemory(List%[0],FrameCt%*Bins,BinSize(List%[1]),0,0,0,0,"Make me
one","S");   'Create memory view
WindowVisible(1);

For i%:=1 to frameCt% do

View(data%);
Frame(i%);


For J%:= 1 to List%[0] do
 ArrConst(MoveArr[],View(Data%,list%[j%]).[]);  'Get data from main view
into array
 ArrConst(View(Res%,list%[j%]).[Begin%:Bins],Movearr[]);    'Copy array to
new memory view
 next
Begin%:=Begin%+Bins;    'Need to move through the data
next
Return 1
End

*******************************************
```

**Did you know**

Hot Keys are available in the Signal Program a list of these are given below.

Keyboard Shortcuts for Signal

Windows software is usually orientated towards control by means of the mouse and menus, but it is often convenient to use the keyboard instead. For interactive adjustments of the data or display, keyboard control can also be much faster. With this in mind, Signal includes keyboard shortcuts to handle most display manipulation and analysis requirements:

Keyboard display control

| | |
|---|---|
| Scroll data down / up | *Cursor down / cursor up* |
| Decrease / increase Y range | *Ctrl cursor down / Ctrl cursor up* |
| Optimise Y range | *End* |
| Show all Y range | *Home* |
| Y axis dialog | *Ctrl Y* |
| Scroll left / right | *Cursor left / right* |
| Decrease / increase X range | *Ctrl cursor left / Ctrl cursor right* |
| Show all X range | *Ctrl Home* |
| X axis range dialog | *Ctrl X* |
| Next frame / previous frame | *PgUp / PgDn* |
| First frame / last frame | *Ctrl PgDn / Ctrl PgUp* |
| Hide selected channels | *Del* |
| Customise display | *Ctrl Del* |

Keyboard analysis control

Channel arithmetic

| | |
|---|---|
| Zero channels | *Shift Z* |
| Negate data | *Shift N* |
| Rectify data | *Shift R* |
| Subtract DC level | *Shift O* |
| Differentiate data | *Shift D* |
| Integrate data | *Shift I* |
| 3-point smooth | *Shift 3* |
| 5-point smooth | *Shift 5* |
| Shift 1 point left | *Shift <* |
| Shift 1 point right | *Shift >* |

Frame buffer operations

| | |
|---|---|
| Toggle display of frame buffer | *Ctrl B* |
| Clear buffer data | *Ctrl 0 (zero)* |
| Add frame to buffer (average) | *Enter* |
| Add frame to buffer | *+* |
| Add buffer data to frame | *Ctrl +* |
| Subtract frame from buffer (average) | *Ctrl Enter* |
| Subtract buffer data from frame | *-* |
| Subtract frame from buffer | *Ctrl -* |
| Copy frame data to buffer | *Insert* |
| Copy buffer data to frame | *Ctrl Insert* |
| Exchange buffer and frame data | *Shift Insert* |
| | |
| Multiple frames dialog | *Ctrl M* |

All of these shortcuts are documented with the appropriate menu commands.

**Recent questions**

Q.      How can I make a movie of Spike2 or Signal to show my students how to operate them.

A.      There are a number of frame grabbers that do this for you. Most are available as evaluation copies but do declare this on the screen. One that we have found to be quite useful can be found at  www.hyperionics.com and is called HyperCam.

**User group**

It will take a short time to set up an archive message board on our web site. As soon as this is done I will use this newsletter to announce that it is available. This seems to be the most sensible route to go rather than direct emailing both for security and for minimising spam.