ED eNews

The eNewsletter from Cambridge Electronic Design

#70 ~ July 2012

Contents

News – Updates & Meetings Scripts Spotlight - Finding minimum and maximum values Spike2 – Using loops in sequence files Spike2 Script – Cursor updates Signal - Overdrawn data 3D Signal Script – Generate random pulses Did you know...? - Set codes for overdrawn WaveMarks Recent Questions - Lock Y-axes on overdrawn channels

News

US Training Days: Thursday 11th and Friday 12th October

Our US training days will take place at the DoubleTree by Hilton, 300 Canal Street, New Orleans just prior to the Society for Neuroscience meeting. These days are designed for both existing and prospective users and are made up of multiple threads to cover all aspects of sampling and analysis with Spike2 and Signal.

Attendance is limited so please register early to reserve your place on the appropriate course.

Reduce mains noise contamination with the CED 4001-16 Line frequency pulse unit



The CED 4001-16 Line frequency pulse unit generates mains-frequency-locked pulses and can be used to reduce the magnitude of mains noise ('hum') that can sometimes be seen in recorded waveforms. This unit allows you to record the mains pulses as an event channel, alongside the waveform signal. Hum removal can then be performed off-line using the Spike2 scripts HumRemove.s2s or HumRemoveExpress.s2s which generate an average of the mains noise using the recorded mains pulse as a trigger. This average is then subtracted from the raw waveform to reduce the mains noise.

Estab	lished	119	70
)	. 8	1	
F	5	1	
- 2-	- [+-	
	3		
-5-	- [1	
	E	-	
to top	5	2	
1	2	2	
	2	-	
~~	- [
	3	-	
5		-	
	5	-	3
2	5	-	Ł
A	- ?		3
	2		ł
- E	- [-	1
15	5	-	
5	5		I.
1	8	÷	Ł
	5		Ł
+	- 1	-	
	5	1	1
5	6	-	Ł
	8		
	5		
4			1
3	3		1
5	- [-	1
)	5	1	1
5	- 5	-	ł
1	1	1	ţ
5	5	-	F
1	-7		
	5		1
2	3	+	1
		to top	to top

CED 2502 Skin conductance unit



Taking advantage of the latest components, the redesigned CED 2502 uses the constant-voltage principle and has manually switched conductance ranges of 10, 25 and 100 micromhos. Suitable electrodes for GSR studies are supplied with the unit.

Future meetings and events

International Motoneuron Meeting 2012

Crowne Plaza Hotel Coogee, Sydney, Australia July 23rd – 26th 2012

CED US Training Days 2012

Doubletree by Hilton, New Orleans, LA, USA October 11th - 12th 2012

Society for Neuroscience 2012

Ernest N Morial Convention Center, New Orleans, LA, USA October 13th - 17th 2012

Latest versions of Spike2 and Signal

Updates for Spike2 and Signal are available from the <u>CED downloads</u> page, or by clicking on the links in the table below. Demonstration versions of the latest software are also available.

Spike2 downloads	Released	Signal downloads	Released
Spike2 version 7.10b	07/12	Signal version 5.07	06/12
Spike2 version 6.16	03/11	Signal version 4.09	08/11
Spike2 demo	05/12	Signal demo	06/12

Scripts Spotlight

I often find that when writing scripts for analysis a common requirement is to locate the maximum and minimum values in a channel. It is of course possible to do this using active cursor searches, but setting up active cursors can take many lines of code within a script. Luckily for the script writer, there is a single function, MinMax(), that can be used to return these simple values

The MinMax() script function returns the minimum and maximum values from a channel in a data file or result view, or the minimum and maximum interval from an event or marker channel drawn as dots or lines. MinMax() can also make use of additional arguments to return the X-axis position of the minimum and maximum values and set the channel draw mode to use for the measurements.

Func MinMax(chan%, start, finish, &min, &max,{,&minP{,&maxP{, mode% {,binsz {,trig%|edge%{, as%}}}});



The attached script code example contains a user-defined function called <u>UseMinMax%()</u>. This utilises the MinMax() function to assign the minimum and maximum values on a selected channel between time 0.0 and the end of the file to two variables named minVal and maxVal. It also returns the positions of the minimum and maximum values and sets two cursors to mark these times before displaying the measured values in a message box.



- Q. I am trying to use the graphical editor to generate a series of pulse trains from digital output 0. My problem is that, when I try to run the sequence, I get an error because there are too many instructions required to output all of my pulses. If I save the sequence as a text file I can see that there is a long list of DIGOUT commands, one for each of my pulses. Is there a more efficient way to produce repeating pulse trains within the sequence?
- A. The graphical editor is a quick and easy way to generate outputs and is usually suitable for most users requirements but, for more complex output tasks, the text editor offers greater flexibility and control. In this case, allowing the user to utilise loops to repeat a series of sequence instructions.

In this example, rather than using the DIGOUT and DELAY instruction multiple times to generate the required number of pulses, it is far more efficient to take the instructions that generate a single pulse duration with interpulse interval, and then use a loop that simply repeats these the required number of times utilising the MOV and DBNZ instructions.

S Polse trains			0 0 0	3	Pulse	train 3	qoo				
No errora			18 II	No	w curr	ent					¥ 11 == 4 ?
	SET	1,1,0	; Get rate & scaling OK					-	SET	1,1,0	:Get rate & scaling CK
	Vae	V45 LoopCal	· Define variable for section loops						Van	V1. counter	Wartable for loss counter
	VAR	V46.RampC=0	I Define variable for rann lonna						VAR	V2. repeater100	Wariable for no. repeats
	Van	V47 Delayten	. Define variable for delay loops								the second second second
	VAR	Vill Delauled	and another one	11.5	- 0	iono i	rn.	12	HALT		(Teletal eastion, no submit
	VAD	V40 Celaviel	and another one	11.5	τ.						the second second second
	VAR	V50.Delavi=0	and another one	1 3	FI 0	001	EAL	12	1007	counter, repeats	This sets the counter
	VAR	V51.Delav5=0	and another one		Т					101000000000000000000000000000000000000	
		Contraction of the second			0	002	00:-		DIGOUT	farmer 21	
EOr	DIGOUT	E			0	003			DELAY	#40.0991-1	
	DELAY	#10.9921-1			8	004			DIGOUT	1	
	HATT		. The of this semistre section			005			DETAY	*10.0891-7	
	10004		· ···· ··· ···· ·····			0.04			DBNZZ	counter DO	This charge the count and loops back
FAI	DIGOUT	1			1.1				Province.	an an an a star a star	Lines success and evening and anothe success
	DETAY	a (0.008)-1			1.0	007			TIME	#0	hark to initial halt the seconder
	DIGOUT	1									
	DELBY	# #0.0993-1									
	DIGOUT	10		1.5	_	_		_	-		
	DELAY	= (0.099) -1		_							
	DIGOUT	1									
	DELAY	# (0.099)-1									
	DIGOUT	[
	DELAY	#(0.099)-1									
	DIGOUT	[
	DELAY	# (0.099)-1									
	DIGOUT	10									
	DELAY	# (0.059) -1									
	DIGOUT	1									
	DELAY	#(0.099)-1									
	DIGOUT	1									
	DELAY	# (0.0991-1									
	DECOURT	F									
	DELAY	# (0.0991-1									
	DIGOUT	F									
	DELAY	##0.0991-3									
	DIGOTT	1									
	DELEV	+(0.059)-1									
	DTOATT	1 01									
	DETAY	# 40 04ht -1									
	at case	a for one of a 1									
*											

Comparison of sequences generated using the graphical editor (left) and using a loop structure in the text editor (right) to generate a pulse train with 100 pulses

The MOV instruction sets up a counter using a variable value. This should be set to the number of repeats you require. The DBNZ instruction (which stands for Decrement and Branch if Not Zero) subtracts 1 from the value of the counter and, if the result is not 0, branches back to our first DIGOUT instruction to repeat the pulse output. Once the counter reaches 0, the DBNZ instruction

くろうろうろうろうろうろうろうろうろうろうろう Back to top 2

allows the sequence to progress to the next instruction, in this case, to JUMP to the initial HALT instruction and end that sequence section.

The DELAY instruction can take a value in s(), ms() and us() and convert these into the correct number of sequencer steps for the required delay. You can also subtract additional sequencer steps to take into account the time required to execute the DELAY instruction itself, as each sequencer instruction takes one (or more) steps. In the example above, the second DELAY instruction subtracts 2 additional sequencer steps from the delay time to account for the time required to execute the following JUMP instruction as well.

Scripts: Spike2

- Q. Are I am attempting to write a script that finds the first intersection between a waveform channel and a horizontal threshold cursor, but have to re-run the script each time I re-position the horizontal cursor. Is there a way to get this to update whenever I move the threshold?
- A. The script, <u>FindIntersect.s2s</u>, uses an Idle routine to automatically update a cursor position when any movement of the horizontal 'threshold' cursor is detected. This example will interactively update Cursor(0) to always show the first intersection point in the viewed time range.



- Q. I'm putting together a poster presentation and would like to generate a 'waterfall' image of multiple data traces. Can I do this in Signal?
- A. Signal can overdraw frames and subsets of frames from a data file in the current view to allow visual comparison of recorded data. The Overdraw settings option available from the View menu (or using the Ctrl+Shift+D keyboard shortcut) opens a dialog that sets the frames to include in the list and the manner in which overdrawn frames are displayed.

🔽 Overdraw fra	mes 🔽	3D drawing			
Frames All frames 👻					
Frame subset	All frames 💌				
Colour usage	Fade to backg	round 👻			
Maximum overdrawn frames		0			
Maximum time ra	0 🄶 s				
X axis space for	0 🌲 %				
Y axis space for	40 🊔 🎗				
Perspective X si	65 🊔 🎗				
Perspective Y si	50 🊔 %				
3D Z axis is scaled by the irame		Index 🔻			
🔽 Update displa	ay to show change	s			
		Cancel			

Overdraw settings dialog

Overdrawn data can be displayed on the same time axis as the current frame or using a 3D 'waterfall' display mode. When the 3D drawing checkbox is selected the user can adjust the X and Y settings used to display the overdrawn data.

Show me how!

Click on the link above to play a video tutorial describing the 3D Overdraw settings in Signal

Back to top

Back to top

Scripts: Signal

- Q. I am planning an experiment and would really like to be able to apply a pulse train where the pulses have a random interval. Is this possible in Signal?
- A. The script, <u>SetRandPulses.sgs</u>, is an example script that adds trains of output pulses with exponentially distributed intervals to the pulse editor. It generates a toolbar to open a new data file, using the current sampling configuration, and to start and stop sampling. At the end of each sweep of data the script generates a new set of pulses which are then output from digital output 0 when sampling continues. These pulse output times are also marked in the digital marker channel for reference.

Did you know ...?

You can change codes for spikes drawn in triggered display mode by using the Ctrl+Alt keyboard shortcut and clicking and dragging a line across the data. Any spike that crosses the line can then be set to any code.



Selecting spikes with Ctrl+Alt and drag in triggered display mode

Recent Questions

- Q. I often drag and drop my channels on top of each other for visual comparison of amplitudes and latencies, but when I re-scale the data in this display, only the first channel updates. Is there any way to force the other overdrawn channels to the same scale?
- A. When channels are 'stacked' by clicking and dragging the channel number to the left of the Y-axis and dropping it on another channel in the same view, the y-axis display relates to the first channel in the group. This is always the 'top-most' channel number displayed on the left of the shared Y-axis. Double-clicking on a channel number promotes it to the top of the group.

You can set all of the 'grouped' channels to share the same y-axis by opening the Y-range dialog and selecting the Lock axes checkbox. The Group offset field can be used to set a vertical display offset to apply to each locked channel, giving an evenly spaced display of each channel in the group.

Channel 1 EMG (Waveform)		-	Ontimise
Top 12 24556.29			Show All
Bottom 300.8632	Volt		Draw
Lock axes Group offset	0		Holp
Large tick spacing	1		пер
Tick subdivisions	4	*	Cancel
Logarithmic Auto units	Off	•	Close

Y range dialog showing Lock axes and Group offset options



User group

Try the <u>CED Forums</u> 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.

.