

Spike Studio 3.5_

by Eli Meir

Introduction

Introduction.....	2
Authors Address and Phone Number.....	3

Orientation

General Orientation to Spike Studio.....	6
Figure 1: Screen Layout.....	8
Changing Parameters (or Changing Variables).....	9
Changing the Mouse Function.....	9
Figure 2: The Icon Menu.....	10
A Little Terminology.....	11

Getting Started

Getting Started.....	14
----------------------	----

Digitizing

Digitizing.....	20
Setting Information about the Board.....	21
Setting the Sampling Rate.....	21
Selecting the Channels to Collect From.....	22
Setting the Board Gain.....	23
Triggered Acquisitions.....	24
Single Trigger Acquisitions.....	25
Multiple Triggering.....	27
Triggering with a Delay.....	29
Digitizing Data.....	30
NI: Digitizing Data.....	30
SB: Digitizing Data.....	31
GW: Digitizing Data.....	32
Size of Digitized Data.....	34
Amount of Data to Collect.....	35
NI: and SB: Demultiplexing.....	36
NI: and SB: Stopping Demultiplexing.....	37
Final Data.....	38

Waveform Output (D/A)

Outputting Waveforms (D/A).....	40
Outputting a Waveform.....	41
Analog Waveforms.....	42
Digital Waveforms.....	43
Making Waveforms for Output.....	44
Outputting Waveforms While Triggering.....	45
Initialization of Output Channels.....	45

Display

Displaying Data.....	48
Opening a Digitized Data File.....	48
Opening a Text Data File.....	49
General Screen Organization.....	50
Changing the Active Window.....	51
Channel Number Parameter.....	52
Drawing and Redrawing Data.....	53
Forcing Traces to Redraw.....	53
Positioning of Windows on the Screen.....	54
Tiling Windows.....	54
Time From Start Parameter.....	55
Seconds per Window Parameter.....	55
✕ Zooming in on Data.....	56
Overlapping Screens.....	56
Clipping the Data.....	57
Display Modification Functions.....	58
Rectifying Data.....	58
Inverting Data.....	58
Correcting Offsets in the Data.....	59
Changing the Offset.....	59
Changing the Gain.....	60
Effect of Display on Analyses.....	60
Drawing Algorithm.....	61
Outputting Pictures.....	62
A Note on High Resolution Pictures.....	62
Copying a Picture to the Clipboard.....	63
Saving a Picture.....	63
Printing.....	63
Page Setup.....	64
Page Layout.....	65
Dividing up the Page.....	65
Putting Pictures on the Page.....	66
Outputting the Layout.....	67
Saving and Loading Layouts.....	67

Event Analysis

Event Analysis.....	70
Spikes.....	70
Automated Event Analysis.....	71
↳ Lower Threshold.....	71
↳ Upper Threshold.....	71
Minimum Event Duration.....	72
Minimum Interevent Duration.....	72
Minimum Spike Number.....	72
A Very Simple Filter.....	73
Calculating Events.....	73
Midevents.....	73
Event Finding Algorithm.....	74
An Alternate Event Finding Algorithm.....	75
User Modifications of Calculated Events.....	76
↔ Adding Events.....	77
⊗ Deleting Events.....	77
↖ Undeleting Events.....	78
→ ← Merging Events.....	78
Getting Rid of User Modifications of Events.....	79
Number of Events.....	79
Comparing Between Two Traces.....	80
Calculating Information About Events.....	81
# Counting Spikes in an Event.....	81
~ Spiking Frequency.....	81
┌ Average Instantaneous Spike Frequency.....	82
↑ Event Peak-to-Peak Voltage Difference.....	82
∫ Event Integral.....	83

Saving Analyses of Events.....	84
Save Options.....	84
Saving.....	85
Saving to Clipboard.....	86
Saving Instantaneous Frequencies for One Event.....	86
Loading Events.....	87
Recalling Parameters Used to Calculate an Event File.....	87
Storing Events.....	88
Markers.....	89
Storing Markers.....	89
Latencies and Number of Spikes After the Markers.....	90

Digitized Data Analysis

Digitized Data Analysis.....	92
?Finding Time and Voltage in the Data.....	92
Data Arithmetic.....	93
Setting Up a Data Storage Area.....	94
Sampling Rate of a Data Storage Area.....	94
Display of Stored Data.....	95
Choosing the Input Trace for the Storage Area.....	95
+Adding Data.....	96
-Subtracting Data.....	96
Adding and Subtracting Data From Markers.....	97
Scaling Stored Data.....	97
Other Storage Area Functions.....	98
Averaging Data With Markers.....	98
Saving the Stored Data.....	98
Erasing the Data Storage Area.....	99

Graphs

Graphs

.....
102

Making a New Graph

.....
102

Setting the Characteristics of a Graph

.....
103

Setting Up a Histogram

.....
105

Setting Up a Raster Plot

.....
107

Changing a Graph Without Erasing Data

.....
108

Adding Data to the Graphs

.....
109

Automatic Graph Updating During Data Collection

.....
110

Saving the Data in a Graph

.....
111

Saving Data in a Graph to the Clipboard

.....
112

Clearing a Graph

.....
112

Changing the Size of a Graph Window

.....
112

Saving, Copying, and Printing a Graph

.....
113

Removing a Graph

.....
113

Other Functions

Other Functions

.....
116

Changing Memory Allocation

.....
116

Setting Memory Usage

.....
117

Filtering Event Files

.....
118

Cutting Data

.....
118

Artificial Events

.....
119

Saving the Configuration

.....
120

Saving Digitized Data

.....
121

Total Seconds in File

.....
122

Sampling Rate Info Item

.....
122

Clear

.....
122

Inactive Menu Items

.....
122

Quitting the Program

.....
122

About Spike Studio

.....
123

Acknowledgments

.....
123

Appendix A: Application Notes

Contents

.....
126

Application Note 1: Post Stimulus Time Histograms

.....
127

Application Note 2: GW: and NI: Triggered Data Acquisition

.....
132

Application Note 3: Data Averaging

.....
137

Introduction

Introduction

Spike Studio was written for several neurobiology laboratories to digitize and analyze physiological signals on the Macintosh family of computers. Spike Studio will digitize multiple channels of data, allows flexible viewing of the digitized data on the screen, and has nice features for making publication-quality figures from this data.

The power of Spike Studio is its event discrimination algorithm, which will automatically pick out all the events in a stretch of data. An event can be a spike, a burst of spikes, a stimulator pulse, or anything else with an onset and an offset. Several types of analyses can then be performed on the events including counting spikes, finding spiking frequencies, finding the peak-to-peak heights, and integrating. Raster plots and PST histograms can be made either on-the-fly or post-processing. In addition, Spike Studio includes data addition, subtraction, and averaging functions. Thus Spike Studio facilitates the analysis of many of the types of data normally encountered in neurophysiological work.

Spike Studio has also been successfully used in undergraduate and graduate neurobiology laboratory courses.

This manual is intended to explain how to use each feature in Spike Studio and exactly how it functions. It also contains small introductions to each group of features, and two sections at the beginning which are intended to help new users of the program get oriented. Aside from these two sections, I have tried to design this manual more as a reference than as a tutorial. The manual does not include a lot of background information or give examples of how to use Spike Studio to analyze specific types of data, although with a little creative use of the programs features it is possible to analyze data in many different ways. Some examples of how to use the program to perform various tasks are included in a series of application notes in an appendix. The manual also assumes a general familiarity with neurophysiological data, as well as a familiarity with the Macintosh user interface. If you are not familiar with one or both of these, particularly with how to use a Macintosh, you should find references on them before reading this manual.

Spike Studio works with National Instruments and GW Instruments digitizing boards, as well as with sound digitizers (including the microphone ports built into the newer Macintoshes). Since there are some differences between these different brands of digitizing board, there are some parts of the manual which will only apply to owners of one brand. These parts will be marked off with either an **NI:** a **GW:**, or an **SB:** (for sound board), and will be indented relative to the rest of the text.

This manual does not discuss the Spike Studio source code. Source code is available if you have a good reason for wanting it, however it is probably much faster and easier to ask the author to make changes.

Spike Studio is a continually evolving program. As more people use it, more features are added to the program, and new versions are available to all owners. If you find that there is some feature which would be useful for you to have in the program, I may be able to add it for you in the next update. I am also always ready to fix any bugs which are found. If a major bug is found (which hopefully will be rare) I will send a new version of the program to all registered user. When minor bugs are found I will include the fix in the next update.

Previous to this version, Spike Studio has been a commercial program, which was sold to laboratories across the U.S., in Europe and in the Middle East. I am now releasing this version (v. 3.5) as pseudo-freeware. What this means is that you are free to use the program as much as you want, without any obligation to pay me. However, if you like the program and wish to encourage me to keep supporting it, and if you want to receive notice of updates, bugs, etc., then I would appreciate getting paid. The price of the program is quite reasonable, perhaps even cheap compared to other commercial data acquisition programs, and I think the benefits you will receive will be worth it.

Enjoy, and I hope Spike Studio helps you in your data analysis. For more information, help, bug reports, etc., write or call:

Eli Meir
149 Grandview Ct.
Ithaca, NY 14850
607-273-1804 (leave a message)
email: meir@zoology.washington.edu

P.S. A note on distribution. Spike Studio v3.5 and the accompanying manual are freely redistributable, as long as no one charges for or makes any money off of the distribution. The program must be distributed for \$0, not even charging for costs such as shipping and handling, etc.. Also, the read_me file must be sent along with the program.

Macintosh is a registered trademark of Apple Computer Inc.. Canvas is a registered trademark of Deneba Software Inc.. Spike Studio is a trademark of Eli Meir.

Orientation

General Orientation to Spike Studio

Figure 1 on the following page shows an outline of a typical screen in the Spike Studio program. This section will explain generally what each of the parts of that screen are, where in the manual to find information about them, and also a little of the terminology that is used in this manual. Hopefully this will help orient you to the manual. If you have never used Spike Studio before, you may also want to go through the Getting Started section which will take you through the steps you would use in a typical session with Spike Studio.

If you look at the picture of the screen on the next page, you will see that Spike Studio uses several windows. On top there are two windows displaying data. These are called the data traces. The upper of these is currently the active window (as seen by the fact that its title bar is not ghosted), and thus it is the one which can currently be manipulated and analyzed. Just below the trace itself and within the same window is a thin strip where the onsets and offset of events are displayed. You can see that several spikes have been picked out on the traces displayed. In the lower part of the screen are three more windows. The left window contains the parameters (also called the variables) which help control the display of the traces and the event discriminator. To the right of the parameters in the same window are three useful pieces of information. On the right is a window containing the icon menu, which enables you to pick out different functions for the mouse to have. At the very bottom of the screen is a message display window. When the program needs to give you a message it will display it here.

At any time, one of the parameters and one of the icons in the icon menu will be highlighted. The highlighted parameter is the one that can currently be changed. The highlighted icon indicates the current function of the mouse. The mouse pointer will also change shape to indicate its current function whenever the mouse is positioned inside a data trace window.

If you use the graphing features of Spike Studio another window will appear for each graph which is displayed (see Graphs chapter, p. 102). Similarly, if you use the data storage area of Spike Studio to do data addition or averaging, a window will appear to display the results of these operations (see Digitized Data Analysis chapter, p. 92).

All windows in Spike Studio can be moved around and resized just like any other Macintosh windows. Some of the windows will not show the resizing icon that normally appears in the bottom-righthand corner, but they can be resized nonetheless.

The chapter which follows this one is a series of step-by-step instructions that take you through the entire process of digitizing and analyzing some data for events. If you have never used Spike Studio before, it is a good place to start. After that, the manual is divided into several chapters which parallel to a certain extent the organization of the screen.

The Digitizing chapter (p. 20) discusses data digitization (collection). It explains the Spike Studio commands which are associated with digitizing data, and also talks about the characteristics of the data and data storage.

The Waveform Output chapter (p. 40) talks about the waveform output features of Spike Studio. It explains how to construct a waveform for output, and how to set up Spike Studio to output that waveform on either an analog or a digital output port.

The Display chapter (p. 48) covers the data display features of Spike Studio, explaining how to load data into the program, how to select between the traces, and how to modify the appearance of each trace. All of the items in the Display menu are explained in this chapter. In addition, this chapter discusses how to save and print pictures from Spike Studio. In reference to figure 1, the display section talks about the two traces at the top of the screen, one of the icons in the icon menu and three of the parameters - Time From Start, Seconds Per Screen, and Channel Number.

The Event Analysis chapter (p. 70) describes how to do event analysis in Spike Studio. This chapter explains most of the commands in the icon menu. It also explains the two horizontal lines running through each trace, the spike onsets and offsets (shown just below each trace), and four of the parameters - Lower Threshold, Upper Threshold, Minimum Event Duration, and Minimum Interevent Duration. The information item Number of Events is explained here, as are all of the items in the Analysis menu.

The Digitized Data Analysis chapter (p. 92) discusses adding, subtracting, and averaging digitized data, as well as other functions associated with analyzing the digitized data itself (as opposed to analyzing events). It explains three of the icons in the icon menu - the question mark and the plus and minus signs. It also explains all items in the Data menu.

The Graphs chapter (p. 102) explains the graphing functions in Spike Studio. Currently you can make histograms and raster plots. All items in the Graph menu are explained here.

The Other Functions chapter (p. 116) covers some odds and ends that have been put into the program because they're useful, but didn't really fit anywhere else. These include most of the items in the Extras menu and the information items Total Seconds in File and Sampling Rate. There are also several little tidbits of information here that might be interesting. The sections on allocating memory might be particularly useful.

Finally, there is an appendix (p. 127) which contains examples of how to use Spike Studio for various applications.

Changing Parameters (or Changing Variables)

There are seven parameters shown in the Parameters window. Four of these - 'Lower Threshold', 'Upper Threshold', 'Minimum Event Duration', and 'Minimum Interevent Duration' - help control spike and event discrimination and will be discussed in the Event Analysis section of this manual (p. 70). The other three - 'Time From Start', 'Seconds per Window', and 'Channel Number' - help control the screen display and are discussed in the Display section of this manual.

The parameters shown will always be those of the currently active trace window (the last trace which you were working on). There are two ways to change a parameter. The first way is to use the mouse to click on one of the parameters. This will highlight it. You then enter the value you would like the parameter to have, and press return when done. The parameter will now be set to the value you entered. As you type, you will see the number you are typing appearing next to the parameter. A decimal point will only appear once the number after the decimal has been typed. If you make a mistake in typing, press any key that is not a number to cancel the change in the value.

You can also use the arrow keys to change the variables. Use the up and down arrow keys on the keyboard to highlight the variable you want to change, and then use the left and right arrow keys to make the number bigger or smaller. The left key will make the number smaller, and the right key will make the number larger, usually by the smallest increment possible or reasonable. If you hold down the left or right arrow key, the numbers will increase or decrease rapidly, allowing you to change them by large amounts. In general, you will want to type in a number if you want to change the variable by a large amount, and use the arrow keys if you want to change it incrementally.

After you change a parameter, the window of the currently active trace will automatically be brought forward and made active.

Changing the Mouse Function

The mouse has many different functions in Spike Studio, and you can easily switch between these functions. To change functions quickly, use the icon menu window on the lower right-hand part of the screen, which includes an icon for almost every mouse function in the program. Figure 2-2 on the following page shows the function each icon is associated with and the page number where that function is described. The highlighted icon indicates the current mouse function. To switch mouse functions, click on the appropriate icon, and it will become highlighted. The mouse pointer will also change shape depending on its current function; many of these shapes are similar to the icon for that function. You can also change the mouse function by selecting the menu item associated with the new function.

A Little Terminology

Following is a brief explanation of some of the terminology used in the manual. You may want to refer back to this if you read a word later on and can't quite figure out what it refers to.

Channel/Trace	Each of the inputs on the digitizing board is a channel. Thus a channel of data is the data collected from one of these board inputs in one digitization. It is the equivalent of one of the data files which are stored on the disk. A trace is essentially the same as a channel of data, but it is used more often to describe what is on the screen. Thus each trace window on the screen contains the trace representing one channel of data (p. 48).
Configuration	The configuration is a place where initialization information is stored for the program. In many programs this would be known as a configuration file, but in Spike Studio it is stored as a part of the program itself so there is no extra file. Thus it is referred to simply as the configuration, not the configuration file. For more information, see Saving the Configuration (p. 120).
Data	This refers to any type of data in the program. This can include digitized data, event onsets and offsets, or any of the calculations which can be done on events. Which exactly it refers to depends on context. However, it is most commonly used to refer to digitized data. The 'Data' menu includes items which manipulate digitized data (p. 92).
Digitized Data	Digitized data refers to the numbers which come out of the digitizing board (the "raw" numbers, before any type of processing such as event discrimination has been done on them) (p. 38).
Data Storage Area	This is an area of memory set aside to perform operations on digitized data, such as averaging data (p. 93).
Event	Anything with an onset and an offset. Usually a piece of data that has been picked out as a spike or burst (p. 70).
Spikes	A piece of data which goes above and then below some voltage level. See the definition of spikes in the Event Analysis section (p. 70).
Stored Event	The program has a special storage area for events that is not associated with any trace. The events there are called stored events, and are used primarily in making histograms (p. 88).

Marker	Markers just mark off a set of times, to be used for certain operations such as data averaging (p. 89).
Trace	See Channel/Trace above.
Active Trace	See Currently Active Trace below.
Currently Active Trace	When there is more than one trace window open, only one of them will be active at any time. The active one is the one whose parameters are displayed and which you can work with (p. 51).
Mouse Function	At any time when there is data loaded into the program, the mouse will perform a certain function if it is clicked somewhere on the active trace. The function performed will be the current mouse function (see Changing the Mouse Function above).
NI:	NI stands for National Instruments, a company which makes digitizing boards that Spike Studio knows how to use. A bold NI: means that the following section of the manual only applies to users of National Instruments boards.
GW:	GW stands for GW Instruments, a company which makes digitizing boards that Spike Studio knows how to use. A bold GW: means that the following section of the manual only applies to users of GW Instruments boards.
SB:	SB stands for a Macintosh sound board, which can be either the built-in microphone input on newer Macs, or a third-party sound board. Spike Studio knows how to collect data from these inputs. A bold SB: means that the following section of the manual only applies to people using a sound board to collect data.
Parameters/Variables	The parameters are the numbers on the lower left of the screen which control parts of the screen display and the event discrimination algorithm. They are also called the variables. The parameter which is highlighted can currently be changed (see Changing Parameters above).

Getting Started

Getting Started

Before getting into specifics, this section presents a simple run-through of the program from starting data collection to saving the analyzed data. Those of you who have previously used the program may want to skip this section. We'll start by collecting some data, then we'll find events within that data, and finally we'll save those events to a file.

After installing all the digitizing hardware, the first step is to connect the digitizing board to whatever unit contains the undigitized data (e.g., tape recorder, VCR, amplifier, etc.).

Next run Spike Studio by double clicking on its icon. If this is the first time the program has been run, then the first thing to appear will be a dialog box where you must indicate which type of board you have, and the expansion slot of the computer where the digitizing board is installed (if it isn't a sound board). You should also select the voltage range which your board is set to collect over.

When you finish, a box with a copyright message will appear, which you can get rid of by clicking in it. Then three windows will appear at the bottom of the screen, along with the Spike Studio menus. So that you don't have to input your board type each time the program starts, go to the 'Extras' menu and select 'Save Configuration'.

Before collecting data, we need to tell Spike Studio what channels to collect from and what the sampling rate should be. First, select the 'Channels to Collect' function from the 'File' menu. A dialog box with all possible channels will appear, with some checked and some unchecked. Check the channels you want data collected from and uncheck those which you don't want data collected from.

Finally, select the 'Sampling Rate' function in the 'File' menu. Here you can type in the sampling rate which you want the digitizing to occur at (how many thousands of samples should be taken each second). Type in the sampling rate you want and click 'OK'. (SB: Instead of being able to type in any sampling rate you want, you will be presented with a list of choices from which you must pick one). A good starting sampling rate is 5 kHz.

GW: and SB: Now you are ready to digitize some data. Select the 'Get Data' command in the 'File' menu. A dialog box will appear asking how many seconds of data you want to collect and what the board gain should be (only on GW). For now, leave the board gain alone and set the time to collect to 30 seconds (GW: this dialog will ask only for a board gain if you have a FIFO board).

Next a dialog box will ask you where you want to save the data. This is the actual digitized data, so it will take up a lot of room. You can figure on about 10 to 80 Kbytes of data per second (this number is explained in the Digitizing section, p. 34), so it must be saved to a hard disk with a significant amount of room on it. Thus when deciding where to save the data, be careful to select an appropriate drive. The filename you give can be anything as long as it doesn't end in a number.

When you push the save button on the dialog box, a message will appear telling you to press return when you are ready to start digitizing. At this point, start playing the data. Then press the return key to start the data acquisition. The computer will now freeze up until data collection is finished.

NI: Now you are ready to digitize some data. Select the 'Get Data' command in the 'File' menu. A dialog box will appear, asking where you want to save the data. This is the actual digitized data, so it will take up a lot of room. You can figure on about 10 to 80 Kbytes of data per second (this number is explained in the Digitizing section, p. 34), and you will actually need double this amount of room, so it must be saved to a hard disk with a significant amount of room on it. Thus when deciding where to save the data, be careful to select an appropriate drive.

When you push the save button on the dialog box, a message will appear telling you to press return when you are ready to start digitizing. At this point, start playing the data. Then press the return key to start the data acquisition. The program will keep digitizing data until you push a key on the keyboard or the mouse button. For now, wait until about 30 seconds of data has been digitized and then push a key. A standard Macintosh save dialog box will appear asking what you want to name the data files. The filename you give can be anything as long as it doesn't end with a number. A new message will now appear saying that the data is being demultiplexed. For now, don't press any keys and just wait until this message goes away.

Note: If you get error messages while trying to collect data here and you can't easily figure out what the problem is, you can still try out the program by using the 'Open' function in the 'File' menu to load in previously collected data.

After the data collection is finished, a new window will appear on the screen displaying the first seconds of the first channel of data. The gain and offset of the data are automatically adjusted to fit the data in the window. Beneath the data is a series of variables which you will use to analyze the data. The highlighted variable is the one which you can currently change. Right now, the top one, 'Lower Threshold', should be highlighted. To change the lower threshold, push either the right or left arrow key on the keyboard. This will raise or lower the value of 'Lower Threshold', respectively. To select another variable, push either the up or down arrow key. These will cycle you through all the different variables. The highlighted variable can always be changed using the right or left arrows. The highlighted variable can also be changed by typing in a new number on the keyboard, and then pressing return (p. 9).

For now, concentrate on the top four variables - 'Lower Threshold', 'Upper Threshold', 'Minimum Event Duration', and 'Minimum Interevent Duration'. These are the variables you will use to pick out onsets and offsets of events. An event is a series of spikes which meet the criteria defined below. For each event, the program will generate an onset at the beginning of the first spike in the event, and an offset at the end of the last spike in the event. If you are analyzing the spikes themselves, you can set the parameters so that an event is just one spike.

The thresholds are the horizontal lines you see running across the data on the screen (p. 71). Only data between these lines will be considered as activity when looking for onsets and offsets. All the data below the lower line will be ignored. Any spike which goes above the upper line will be thrown out. Thus you should raise or lower the lower line until it is right above any noise, and adjust the upper line until it includes all the spikes you want to include.

The 'Minimum Event Duration' is the minimum length of an event, in seconds, in order for the event to be counted (p. 72). No onset or offset will be generated for an event which does not last this long. Thus, 'Minimum Event Duration' should be set a little lower than the smallest event which you want to have counted (if you are counting spikes, set it very small using the arrow keys).

The 'Minimum Interevent Duration' is the minimum distance, in seconds, between any two spikes for those two spikes to be counted as separate events (p. 72). Thus any spikes that are within the 'Minimum Interevent Duration' of each other will be combined together and counted as part of the same event (set this very low when picking out individual spikes).

Once you have those four variables set approximately, press 'e' on the keyboard. This tells the program to calculate all the events in the data on the screen (you can also use the 'Events on Screen' command in the 'Analysis' menu) (p. 73). In a few seconds, the program will display a series of green and red lines (all black lines if you have a monochrome monitor). The lines should be approximately underneath the events, with a green line marking the start of each event, and a red line marking the end of each event.

If the lines do not match up with the events to your satisfaction, play with the four variables until you achieve a better fit. In general, the process of finding the events is an interactive process, where you will change the variables a number of times until a good fit is achieved. With a little practice, this process will become fairly quick, and you will find that these four variables allow a great deal of precision in selecting where the onsets and offsets occur.

Sometime, however, there will be an odd event which doesn't get its onset or offset in the proper place. In this case, there are add and delete functions (p. 76). Move the mouse to the icons in the Icon Menu window and click on the one which says 'Del'. The cursor should change to a crosshair. Now move the mouse pointer so that it is within the two lines of the event which you wish to change. Press the mouse button. The onset and offset lines now become ghosted (faint), and a red message appears at the bottom of the screen announcing that an event was deleted. Now go back to the icons and click on the one that says 'Add'. The cursor changes to a leftward pointing arrow. Place the arrow tip where you want the onset to be, and click the button. The arrow now points rightward. Move to where you want the offset to be and again click the button. Now new green and red lines should be displayed marking the onset and offset of the event you just added.

The rest of the variables on the screen allow you to move forward and backward in the digitized data ('Time From Start' - p. 55), display varying amounts of data on the screen ('Seconds per Window' - p. 55), and switch which channel of data you are looking at ('Channel Number' - p. 52).

If you want to find out information about any of the events, such as their heights, number of spikes, or frequency of spiking within them, then either go to the 'Analysis' menu and select 'Event Heights', 'Spike Frequencies', or 'Number of Spikes?', or select the appropriate function from the icons on the screen, and then click within the event you want information about (p. 81). The information will appear at the bottom of the screen.

Once you have satisfied yourself that the computer is finding all the events correctly, go to the 'File' menu and select 'Save' (p. 84). This will bring up a dialog box which asks if you would like to recalculate events or not. For now, just say yes. Then the standard Macintosh save dialog box will appear, asking where to save the analyzed data. This file will be a text file containing all the onsets and offsets of the events (p. 85). You can then use another program to further analyze the onsets and offsets, or look at the numbers in a word processor.

To analyze the next channel of data, increment the 'Channel Number' variable with the right arrow key. The next channel of data will automatically be loaded in, and you can analyze it the same way you analyzed the first one.

When you are done analyzing all the channels, you can either go on to analyze another stretch of data by starting again from the 'Get Data' command, or leave the program using the 'Quit' command.

There are many more functions which can be performed with Spike Studio, but now that you've gone through these functions you have a good idea of the basics of using Spike Studio, and it should be fairly easy to use the manual to figure out the rest of the program. You may also want to look at the application notes in Appendix B for more examples of using the program. Good Luck.

Digitizing

Digitizing

This section will cover digitizing data with Spike Studio. There are several aspects to digitizing data. First of all, you must set up the parameters of the data taking. These include the digitizing board type, the computer slot where the digitizing board is installed, the sampling rate to use, the channels from which to collect data and whether to trigger the data acquisition. Next is the actual data collection, for which the user must decide how many seconds of data to collect and where to save the data. Finally for National Instruments or sound boards the data must be demultiplexed. There are several menu items which control digitizing, and they are all in the 'File' menu. Most of the parameters can be set once and then saved as part of the configuration (see Saving the Configuration, p. 120).

You can also output data as you are digitizing. This is described in the next chapter, Digital Output.

GW: and SB: With either GW Instruments digitizing boards or sound inputs, the limiting factor on digitizing data is the amount of memory in your computer. You may have to trade off the length of time you digitize against the number of channels collected from and the sampling rate. Spike Studio also supports GW's FIFO board, which will enable you to avoid these limitations. In theory, you could also get around this with the sound input, and if someone needs that I can try to implement it.

NI: Spike Studio currently assumes that you have both an A/D board and a DMA board. Spike Studio will not work well without a DMA board. This could be changed if someone is using just an A/D board without a DMA board.

SB: On some models of Macintosh, the sound board appears to do some funny automatic gain adjustments which make this input unsuitable for scientific work. Other Macintosh models appear to give reasonable signals. So proceed with care.

Setting Information about the Board

You must tell the program the brand of your digitizing board, the computer slot where the board is installed, and the voltage range the board is set to collect at. If you are not using Spike Studio to collect data, you can also indicate that you have no digitizing board.

If you have a National Instruments board, you must also indicate whether or not you have a multiplexer board; if you have a GW Instruments board you should indicate whether you have a FIFO board.

You can determine the board's slot number by looking on the back of the computer, or by using desk accessories. The voltage range is set using dip switches or jumpers on the board.

Once you have this information, use the 'Set Board' function in the 'File' menu to tell Spike Studio what they are. When you select this function, a dialog box will appear with selections for each type of board, a place to indicate the slot number, and two choices for voltage ranges. Just select your board type, enter the correct slot number, select the correct voltage range, and press OK. You should only need to do this once if you then save the configuration (see Saving the Configuration, p. 120).

Note that there may be voltage ranges on the board which are not given as options in the program. Inform me, and I can add your voltage range as an option.

Note that if you are using a sound input board, the voltage range will be ignored, and voltage values given by the program will not be accurate. This may be fixed in a future version of the program (depending on demand, so feel free to write and complain).

Setting the Sampling Rate

The sampling rate can be changed using the 'Sampling Rate' function in the 'File' menu. A dialog box will ask what sampling rate you would like. Enter the rate you want to sample at, in kilohertz / channel. If you normally use one sampling rate, you may want to save this rate in the configuration (see Saving the Configuration, p. 120).

SB: Some sound boards (including the built in Macintosh input) have a limited number of sampling rates available. In this case, a dialog box will appear with a list of the possible sampling rates. Check the box next to the rate you want.

NI: and GW: The maximum speed of the digitizing boards is around 90 - 100 kHz, so a sampling rate is available only if that sampling rate times the number of channels of data being taken is less than 90kHz. Higher sampling rates obviously take more memory and disk space per second of data, so if you are running out of disk space or are not able to take as long a time of data as you would like due to memory constraints you may want to reduce the sampling rate.

Selecting the Channels to Collect From

The 'Channels to Collect' command in the 'File' menu allows you to change set which A/D board channels data will be collected from. When you select this function, a dialog will appear containing all the available channels, with check boxes next to each channel. To change the channels data is being collected from, simply check the ones you want, and uncheck the ones you don't want. If the sampling rate is too high, you may not be allowed to collect from all the channels (see Setting the Sampling Rate, p. 21). If you normally collect from the same channels, you will want to save these channels in the configuration (see Saving the Configuration, p. 120).

NI: National Instruments boards require that the number of channels that data is collected from be a power of two. Thus, only 1, 2, 4, 8, or 16 (for single-ended setups) channels are allowed. These channels do not have to be consecutive. Only the total number matters.

On those systems with a multiplexer board, the channels you are selecting are not channels on the multiplexer board, they are still the channels on the A/D board. Thus each channel you select is actually 4 multiplexed channels. This means when you select channel 1 in the dialog box, you will really be collecting from channels 1 through 4 on the multiplexer board. Channel 2 will give you channels 5 through 8, channel 3 will give you channels 9 through 12, and so on up to channel 8 which gives channels 29 through 32.

Setting the Board Gain

NI: It is possible to change the board gain on National Instruments boards but no one has requested this feature yet, so it is not implemented. The gain is always set to 1.

GW: The digitizing board has a gain circuit in its hardware that can be set to 1, 10 or 100 times gain. You can set the board gain each time you select the 'Get Data' command by selecting the appropriate check box. The board gain will be set even if you then Cancel the data taking, so that you can use 'Get Data' to set the board gain even if you don't want to collect data. You will want to save the board gain which you use most commonly in the configuration (see Saving the Configuration, p. 120).

The board gain is different than the software gain that is contained in Spike Studio (see Changing the Gain, p. 60) in that the board gain actually increases the voltage values coming into the digitizer. The advantage of appropriately setting the board gain is that the resolution of the board increases. If you have the board set to -10 to 10 volts the resolution is about 5mV. This increases to a resolution of 0.5mV if the board gain is increased to 10, and 50 μ V if the board gain is set to 100. While a high gain increases the resolution, however, it decreases the voltage range over which data can be collected. Initially your board will be set to either take voltages between -5 and 5 volts or -10 and 10 volts. If you set the board gain to 10, these ranges drop to -0.5 to 0.5 volt, or -1 to 1 volt.

You should also be aware that increasing the board gain will slow down the maximum speed at which the board can digitize. Spike Studio does not take this into account, so if you are digitizing at a very high rate or on a large number of channels and get an error in the digitizing, you may want to try reducing the board gain (if the gain is set to 1, then this is not a problem).

Triggered Acquisitions

Instead of starting each data acquisition by hand, you can have data acquisitions started by a trigger. This trigger can either be a threshold value on a certain channel, or it can be an external signal on the trigger input of the board, and the trigger can be a low-to-high or a high-to-low transition. If you have more than one trigger in a row and would like to collect data after each of these triggers, you can do this with a single command using the multiple triggering feature. If you normally use triggering, you may want to save these triggering parameters in the configuration (see Saving the Configuration, p. 120). For examples of doing triggered acquisitions, see Application Note 2 in Appendix B, p. 132.

SB: It is possible to have a sound input device respond to triggers, and this may be implemented if someone requests it. However, currently the sound input cannot be triggered.

Single Trigger Acquisitions

To perform a data acquisition where a single trigger starts the acquisition, select the 'Set Trigger' function in the 'File' menu. A dialog box will appear in which you specify all the parameters for the triggering. At the top of the dialog box you can turn the triggering on and off. When 'Trigger the Data Acquisition' is checked, the program will wait for a trigger before starting to collect data, otherwise it won't. Below that are the parameters which control the triggering.

GW: There are two sets of radio buttons controlling the triggering. The buttons on the left allow you to choose between triggering from the trigger input on the board (select 'External Triggering') or triggering off of one of the A/D channels (select 'Trigger from Data'). On the right you can choose whether the data acquisition should be triggered on a 'Low to High' transition or on a 'High to Low' transition. If you are triggering off of one of the A/D channels, you must also input which channel to trigger off of (the 'Channel' field) and the threshold voltage at which to trigger (the 'Trigger Level' field). The board will then start digitizing when the data on the trigger 'Channel' crosses the 'Trigger Level' in either the 'Low to High' or 'High to Low' directions. With external triggering, 'Channel' and 'Trigger Level' have no effect.

The 'Multiple Triggering' button is explained in the next section, and the 'Trigger Delay' button is explained below in the Triggering with Delay section.

When using external triggering, once you start the data acquisition you cannot interrupt it. The only way to escape is to restart the computer. However when triggering from a channel you can interrupt the acquisition while the program is waiting for the trigger by moving the mouse (this may take a few seconds). This is provided so that if you set one of the triggering parameters incorrectly (e.g., setting the threshold too high), and the data acquisition will never be triggered, you can escape back to the program. Note that there is a danger with this feature that you might accidentally move the mouse and interrupt the acquisition.

NI: Three buttons on the left allow you to choose the location of the trigger. You can trigger from the external trigger input on the board (select 'External Triggering'), one of the A/D channels (select 'Trigger from Data'), or you can only collect data when the gating input on the board is enabled (select 'External Gating'). On the right you can choose whether the data acquisition should be triggered on a 'Low to High' transition or on a 'High to Low' transition. If you are triggering off of one of the A/D channels, you must also enter which channel to trigger from (the 'Channel' field) and the threshold voltage at which to trigger (the 'Trigger Level' field). The board will then start digitizing when the data on the trigger 'Channel' crosses the 'Trigger Level' in either the 'Low to High' or 'High to Low' directions. With external triggering or external gating, 'Channel' and 'Trigger Level' have no effect.

You may have both internal triggering and external triggering active at the same time. In this case, the program first waits for an external trigger, and then waits for an internal trigger before collecting data. Similarly, you can have both internal triggering and external gating active at the same time. In this case, the program only collects data after seeing an internal trigger which occurred while the gate was active, and after the trigger only collects data while the gate is active.

With internal triggering, you can also collect data just preceding the trigger. This is known as pre-triggering. Set the 'Pre Trigger Amount' field to the amount of data you would like to collect before the trigger.

The 'Multiple Triggering' button is explained in the next section.

When using triggering, once you start the data acquisition you cannot interrupt it. The only way to escape is to restart the computer. However, if Spike Studio waits for 30 seconds after you start a triggered data acquisition without seeing a trigger, it will automatically stop the data acquisition and return with an error message. Note that this may be a problem with Multiple Triggering, since you will not be able to have more than 30 seconds between triggers. If this causes problems for anyone, tell me and I'll change it.

A note on external gating. I have never managed to try this to see if it actually works.

Multiple Triggering

Multiple triggering enables one to collect a set amount of data after each of a certain number of consecutive triggers, without having to collect all the data in-between the triggers. It is designed for situations where a preparation is stimulated multiple times with rest periods between the stimulations, and only a brief time after each trigger is important. It is also nice to use this feature along with the graphs, for on-the-fly graphing of collected data (see chapter on Graphs, p. 102).

Multiple triggering is controlled as part of the normal triggering operations, and when using multiple triggering all of the parameters discussed in Single Trigger Acquisitions must be set. There are two additional parameters which need to be set for multiple triggering - the number of triggers after which to take data and whether to store the data in one file per channel or in multiple files per channel. These are set in a multiple triggering dialog box, along with the switch to turn multiple triggering on and off.

To get to the multiple triggering dialog box, select the 'Set Trigger' item in the 'File' menu. A dialog box will come up where you can set the triggering parameters (see Single Trigger Acquisitions above). Click on the 'Multiple Triggering' button in this dialog box. A second dialog box will appear which contains the multiple triggering controls.

Multiple triggering is turned on when the 'Enable Multiple Triggering' check box at the top is checked. Note that multiple triggering will only work if triggering is enabled. Thus if the 'Enable Multiple Triggering' box is checked but the 'Trigger the Data Acquisition' box (see Single Trigger Acquisitions above) is not, then no multiple triggering will occur.

Data will be collected after each trigger seen up, for the number of triggers specified in the 'Number of Triggers' field of the dialog box. One caveat: if a trigger occurs while data is still being collected from the last trigger, the second trigger will be ignored. There is also a little down time after each data collection while the data is written to disk, and triggers occurring during this time period will be ignored. Saving data does not take a long time, but if the time between the end of one data collection and the next trigger is too small you may run into problems with triggers being skipped, so if you are pushing the limits, beware.

There are two possible ways to save the data after it is collected. One way is to save a single data file for each channel, just as in any other data collection, and put the pieces of data collected after each trigger into the data file consecutively. Alternatively, multiple data files can be stored for each channel, with each data file holding the data collected after one trigger. If 'One file per channel' is set, the data after all the triggers will be stored in one file per channel. No delimiters will be put between the data from one trigger and that from the next. For this reason, you may want to have one of the channels recording the trigger itself. If 'One file per trigger' is set then a separate file will be saved for each trigger on each channel.

When saving separate files for each trigger ('One file per trigger'), the normal system of file naming cannot be used because there is more than one file per channel. Therefore every channel is given its own filename, and numbers are put after these filenames to indicate the trigger number. A Macintosh save dialog box will come up as it normally does after selecting 'Get Data', and you should choose the folder you want to save the data in here, but you can ignore the filename field. You choose the filenames in a second dialog box which will appear after 'Get Data' is selected. The dialog box will contain a number of text fields labeled with the numbers of the channels you are collecting data from. In each field, type in the filename you would like used for that channel. If you are collecting on less than eight channels, there will be extra, unlabeled text fields which you can ignore. If you are collecting on more than eight channels, then you should save the data in 'One file per channel', or else weird things may happen.

As an example of this filenaming, if you are collecting data on channels 2, 4, and 7 and you want to collect after 3 trigger, then the filenaming dialog box will appear with three text fields labeled as 'Channel 2', 'Channel 4', and 'Channel 7', next to which you type the name you would like to give to each channel. Let's say you decide to call them 'stimulus', 'response', and 'trigger'. Nine files will then be created, named 'stimulus1', 'response1', 'trigger1', 'stimulus2', 'response2', etc., with the 1's storing the data collected after the first trigger seen, the 2's storing the data after the second trigger seen, and so on. Note that the filenames you give should not have a number at the end, since this will lead to confusion as to what the trigger number of that file is. When one of these files is opened by the program, the 'Channel Number' parameter will actually give the trigger number of that file instead of the channel number, and will switch between trigger numbers instead of channel numbers (see Changing Parameters, p. 9).

The length of data to take after each trigger is set in the 'Get Data' function. When you select the 'Get Data' function, one of the dialog boxes which appears will ask how much data to collect. Enter the amount of data you want to collect after each trigger (in seconds). This amount of data will be saved after each trigger.

If you would like more help with multiple trigger acquisitions, you may want to look at Application Note 2 in Appendix B.

NI: If you have both external and internal triggering set, then when you start a data acquisition Spike Studio will first look for an external trigger. After it sees the external trigger, Spike Studio will collect data after each internal trigger it sees. No data will be collected after the external trigger and the program will only look for one initial external trigger - this trigger just serves to alert the program to start paying attention to internal triggers.

Similarly, if both internal triggers and external gating are set, when the gate is enabled then the program will look for internal triggers and when its disabled then the program will not look for triggers. No data will be collected without seeing an internal trigger, even if the gate is enabled. If the gate becomes disabled in the middle of collecting data after an internal trigger, then the data collection for that trigger will be finished when the gate becomes active again. Be aware that, as mentioned above, I have never actually seen the gating work.

Triggering with a Delay

NI: While the triggering with delay ability as described below is certainly possible with National Instruments digitizing boards, it is currently unimplemented. If you need this ability, I can write it in for you.

Sometimes you may want to delay data collection for a certain length of time after a trigger condition is met. For instance, if you trigger a preparation but do not expect to see any effects until 30 seconds after the trigger, you may want to start collecting data 30 seconds after the trigger. Spike Studio allows you to set a delay between triggering and the start of data collection of up to around 600 seconds.

Triggering with a delay is controlled as part of the normal triggering operations, and when using delay triggering all of the parameters discussed in Single Trigger Acquisitions must be set. Delay triggering can also be used with Multiple Triggering (discussed above), and in this case there will be a delay after each trigger. To set a delay after triggering, select the 'Set Trigger' item in the 'File' menu. A dialog box will come up where you can set the triggering parameters (see Single Trigger Acquisitions above). Click on the 'Trigger Delay' button in this dialog box. A second dialog box will appear which contains the delay triggering controls.

Triggering with a delay is turned on when the 'Delay after trigger' check box at the top is checked. Below this is a place where you can type in the number of milliseconds you would like to delay for. You must delay for a minimum of 20 milliseconds, and can delay for up to 600000 milliseconds (600 seconds).

Note that there is some slop in the length of the delay, and it is probably only accurate to within 30 to 40 milliseconds, so that you probably should not use this for small delays or when you need a high degree of precision. In those cases, collect data starting from the trigger time, and then use the Cut Beginning function to cut off the part you don't need (p. 118).

Digitizing Data

The section contains the commands you use to digitize data, and explains the format, size, and naming conventions of digitized data files. Much of this is board specific, so you will see many of the functions marked off as only applying to a certain board.

NI: Digitizing Data

The only command in the program which actually starts data digitization is the 'Get Data' command in the 'File' menu. Connect the raw data source (e.g., tape recorder, VCR, live prep, etc.) to the digitizing board, then select 'Get Data'. A dialog box will come up asking where you want to store the digitized data. Be sure the data will fit on the disk you select, (see Size of Digitized Data below), otherwise there will be an error during the data collection (and your disk may become unhappy). The filename you enter here will be the name of the file which will contain the undemultiplexed data (see Demultiplexing below).

Next a message will appear saying to press return when you want to start collecting data. Start the undigitized data playback a little before the place where you want to start digitizing. When you reach the data you want to digitize, press return. Data is now being digitized. The program will keep digitizing data until you tell it to stop by pushing a key (see Amount of Data to Take below). Then it will spend some time demultiplexing the digitized data (see Demultiplexing below), first asking you for the base filename for the demultiplexed traces. The filename you enter will be the base name for the files which hold the data traces. The data from the first channel will be put into a file named <filename>1, the data from the second channel will be in a file called <filename>2, etc.. The only limitation on this filename is that it should not end in a number (see Final Data below). Note that if you use the same filename as you did for a previous acquisition, the previous files will be overwritten and no warning messages will be given.

Finally when the demultiplexing is through the first channel will be automatically loaded.

SB: Digitizing Data

The only command in the program which actually starts data digitization is the 'Get Data' command in the 'File' menu. Connect the raw data source (e.g., tape recorder, VCR, live prep, etc.) to the digitizing board, then select 'Get Data'. A dialog box will come up asking where you want to store the digitized data. Be sure the data will fit on the disk you select, (see Size of Digitized Data below), otherwise there will be an error during the data collection (and your disk may become unhappy). The filename you enter here will be the name of the file which will contain the undemultiplexed data (see Demultiplexing below).

Next you have to decide is how much data you want to collect. A dialog box will appear asking how many seconds of data to collect (see Amount of Data to Collect below). Don't worry about being exact in where you start and end the acquisition, since you can trim the data files afterwards (see Cutting Data, p. 118). This is the last point at which you can stop the data collection. If you press OK, there is no way (short of restarting the computer) not to digitize data for that length of time.

Now a dialog will appear saying to press return when you want to start collecting data. Start the undigitized data playback a little before the place where you want to start digitizing. When you reach the data you want to digitize, press return. Data is now being digitized, for the length of time you specified above.

After data collection is finished, the program will spend some time demultiplexing the digitized data (see Demultiplexing below), first asking you for the base filename for the demultiplexed traces. The filename you enter here will be the base name for the files which hold the data traces. The data from the first channel will be put into a file named <filename>1, the data from the second channel will be in a file called <filename>2, etc.. The only limitation on this filename is that it should not end in a number (see Final Data below). Note that if you use the same filename as you did on a previous acquisition, the previous files will be overwritten and no warning messages will be given.

Finally when the demultiplexing is through the first channel will be automatically loaded.

GW: Digitizing Data

The only command in the program which actually starts data digitization is the 'Get Data' command in the 'File' menu. Connect the raw data source (e.g., tape recorder, VCR, live prep, etc.) to the digitizing board, then select 'Get Data'. The first thing you have to decide is how much data you want to collect, and the gain setting you want to use. (If you have a FIFO board, you won't need to decide on a length of time). A dialog box will appear on the screen asking how many seconds of data to collect (see Amount of Data to Collect below), and whether the board gain should be at 1, 10, or 100 (see Setting the Board Gain above). Don't worry about being exact in where you start and end the acquisition, since you can trim the data files afterwards (see Cutting Data, p. 118).

Next a dialog box will ask where you want to store the digitized data. Be sure the data will fit on the disk you select (see Size of Data below). The filename you type in here will be the base name for the files which hold the data traces. The data from the first channel will be put into a file named <filename>1, the data from the second channel will be in a file called <filename>2, etc.. The only limitation on this filename is that it should not end in a number (see Final Data below). Note that if you use the same filename as you did on a previous acquisition, the previous files will be overwritten and no warning messages will be given. This save dialog box is the last point at which you can stop the data collection. If you press OK, there is no way (short of restarting the computer) of not digitizing that length of time of data.

If you are not doing a triggered acquisition, a message will appear telling you to press return when you want to start collecting data. If the undigitized data is recorded, start the playback a little before the place where you want to start digitizing. When you reach the data you want to digitize, press return. Data is now being digitized. While data is being digitized, the computer will appear to freeze up. When data collection is through, the first channel which you collected will automatically be loaded.

If you are doing a triggered acquisition, a message will appear saying that the computer is waiting for a trigger. The computer will also beep at this point to let you know it is ready. The message will not go away when the acquisition is triggered, so you won't know whether the trigger was seen until the data appears. During a multiply triggered acquisition, the program will display on the screen the number of triggers it has collected, so you will have an idea of whether or not the acquisition is working. If you would like to interrupt a multiply triggered acquisition, press any key on the keyboard. For examples of triggered acquisitions, see Application Note 2 in Appendix B, p. 132.

The data acquisition process is different if you have a FIFO board. Instead of the computer locking up, it will continuously collect data and save it to disk until a key is pressed on the keyboard (except with multiple triggering, which works as described above). Pressing a key or pressing the mouse button will end the data collection.

Size of Digitized Data

You can use the following formula to calculate how much room a data acquisition will take in bytes, given the sampling rate in Hz, the number of channels which you are collecting on, and the number of seconds of data which you collect:

Sampling rate * Number of Seconds * Number of Channels * 2 bytes

The factor of two is included because each sample is 12 bits, or 1 1/2 bytes long, and so takes up 2 bytes of disk space. Each individual trace will take:

Sampling rate * Number of Seconds * 2 bytes

This is also the amount of memory which is required to hold the whole trace in memory at once.

As an example, suppose you collected 60 seconds of data on 4 channels at a sampling rate of 5 kHz. The amount of space it would take on your hard disk to store all that data would be:

$5000 \text{ Hz} * 60 \text{ seconds} * 4 \text{ channels} * 2 = 2,400,000$
bytes = 2.4 megabytes

GW: The program will make sure that there is enough memory to store all the digitized data you want to collect, so you must only worry about having enough disk space.

SB: Some boards, such as the Macintosh sound input on older Macintoshes, only use 1 byte per sample. This will be converted to 2 bytes per sample during the demultiplexing step.

NI: Since National Instruments boards initially save the data in a multiplexed form, you will need to have at least double the disk space given by the above formula because of the space required for demultiplexing (see Demultiplexing below). However, the above formula produces the final amount of disk space used to hold the data. Remember that with a multiplexer board, the number of channels is actually 4 times the number of channels checked in the 'Channels to Collect' dialog (see Selecting the Channels to Digitize From above).

Amount of Data to Collect

With a National Instruments board or with a GW Instruments FIFO board you do not need to decide how much data you want to collect before starting digitization. Instead, once you start collecting data you may stop the data collection process any time you want to by pressing any key on the keyboard. The number of seconds of data that have been taken so far will be displayed on the screen. There is a few second delay between when you push a key and when the data collection stops, but if you get too much data you can cut it off later (see Cutting Data, p. 118). When acquiring data, you must be careful not to collect more data than will fit on your disk, and you must make sure there is enough room on the disk for demultiplexing if you have a National Instruments board, otherwise you may experience some problems.

With a GW Instruments board but no FIFO board, a dialog box will appear on the screen after you give the 'Get Data' command asking how many seconds of data to take. Type in the correct number. The data is stored in memory, and only saved to disk after the collecting is done, so the amount of data you will be able to collect is limited by how much memory you have. You can get an idea of how much memory is available with the 'Set Memory' command in the 'Extras menu (see Setting Memory Usage, p. 117).

In general, it is better to take a little too much data rather than too little, since later on you will be able to chop off the extra. If you are running under multifinder and there is not enough memory to collect as much data as you would like, you might try increasing the amount of memory allocated to Spike Studio (see Changing Memory Allocation, p. 116).

NI: and SB: Demultiplexing

When a National Instruments digitizing board or a sound board collect data, they save it in a multiplexed form on the disk (i.e., one sample from channel one is saved, then a sample from channel two, then one from channel three, etc., until it gets to the last channel, and then it goes back to channel one). To be useful, the data from each channel must be put into its own separate file. This is called demultiplexing.

SB: On many sound boards, the data must be converted from 1 byte / sample to 2 bytes / sample for Spike Studio to be able to use it. This step is also carried out in the demultiplexing.

Demultiplexing will start as soon as you stop collecting data. A standard Macintosh save file dialog box appears, requesting a filename for the demultiplexed data. The name you enter here will become the base name for the data files. The data from channel one will be stored in a file called <filename>1, that from channel two will be in a file called <filename>2, etc.. The only limitation on this filename is that it should not end in a number (see Final Data below). After you input the filename, the data will be demultiplexed, and while this is occurring, the screen will display the total number of seconds collected and the number of seconds that have been demultiplexed so far.

During demultiplexing there is both a multiplexed and an unmultiplexed version of the data so you must have twice as much disk space available to store both versions. After a multiplexed data file has been completely demultiplexed, the original multiplexed file is automatically deleted.

NI: and SB: Stopping Demultiplexing

Sometimes you may take a long stretch of data and then realize that you made a mistake. Alternatively, you may be acquiring data in real time, and not want to take the time to demultiplex in between data collections. In such cases there are a few options for delaying the demultiplexing. If you have just collected data and you want to stop the demultiplexing as it is happening, just press any key on the keyboard. This will stop the demultiplexing within a few seconds. The original multiplexed data file will not be erased, so you may go back and demultiplex the data at a later time using the 'Demultiplex' command in the 'Extras' menu. The demultiplexed files will contain the seconds of data which were demultiplexed before you pressed a key. Thus if you have taken way too much data, you can stop the demultiplexing after the relevant data has been demultiplexed.

Alternatively, if you know in advance that you don't want to demultiplex the data right after you collect it, you can check the 'No Demultiplex' option in the 'Extras' menu. This turns off the automatic demultiplexing, and you can demultiplex the data at a later time using the 'Demultiplex' command in the 'Extras' menu. Note that multiplexed data will be stored in a file with the name you specified in the 'Get Data' save dialog box. This file name defaults to "Temp", so if you collect more data before demultiplexing, you should remember to change the filename for the new data, otherwise you will erase the old data. Once a multiplexed data file is fully demultiplexed, the multiplexed data file will automatically be erased.

WARNING: Be careful when using the 'Demultiplex' command that the number of channels is set to the same number as when you originally collected the data, otherwise the demultiplexed data **WILL BE GARBAGE.**

Final Data

The final, usable, demultiplexed data will be stored in a series of trace files named <filename>1, <filename>2, <filename>3, etc.. <filename> is the name you entered when you were collecting the data (or just before demultiplexing on NI and SB systems) as the name for the traces to be stored under. <filename>1 holds the digitized data from the channel 1, <filename>2 from the channel 2 and so on. For instance if you collected data on channels 1, 3, and 4, and typed in the filename "foo" for the trace names, then you would end up with three files called "foo1", "foo3", and "foo4". These are the files which contain the data to be analyzed by the program. Spike Studio will recognize that the files belong together because they have the same <filename>, and will recognize which channel each file came from by the number at the end. Thus the one restriction on the <filename> is that it should not end in a number.

For the technically inclined, the actual format of the data in these files is as a series of binary numbers between -2048 and 2048. Each number is one voltage value as read off of the digitizing board, with 12 bits of resolution (thus the odd range of values). Each value takes 2 bytes of memory or disk space. There is nothing else in the trace files except these digitized values (although the resource fork of the file contains additional data - see Opening a Digitized Data File, p. 48).

Waveform Output (D/A)

Outputting Waveforms (D/A)

In addition to digitizing data, Spike Studio can output waveforms through the digital and digital to analog ports of your digitizing board. Spike Studio can output waveforms at the same time that it is collecting data, so you can use the output to control experimental equipment while you are digitizing. You could also use Spike Studio as a general waveform output device, although there are probably other programs more suited to this function. All the setup for outputting waveforms is done through the 'Set D/A' function. The output is also affected by the sampling rate. The actual output is carried out each time you use the 'Get Data' function.

SB: Although it should be possible to use the sound output port of the Macintosh to output waveforms, this is currently not implemented in Spike Studio. Nobody has requested this capability, so I have no plans to add it in.

NI: All of the output functionality described below has been implemented for National Instruments boards. However, none of it has been tested. This means that the first person to actually use it will have some problems which we will have to work out together. The problems will not be hard for me to solve, and after they are fixed the output functions will work as advertised.

Outputting a Waveform

Spike Studio can output both analog and digital waveforms. Both digitizing boards which Spike Studio supports will let you output up to 2 analog and 1 digital waveform simultaneously.

To set up the output, use the ‘Set D/A’ function in the ‘File’ menu. A dialog box will appear where you can turn the output on and off, select which traces to output on which output channels, and indicate whether you also want to input data as you output data. You can use any trace in Spike Studio as an output waveform, and Making Waveforms for Output (below) describes how to go about making a trace with the waveform you want to output. This requires some care, so read that section.

To turn on or off waveform output, check or uncheck the ‘Turn on D/A’ check box. If the ‘Turn on D/A’ box is checked, then each time you ‘Get Data’, the waveforms you indicate below this will be output.

If you want to collect data as you output waveforms, then check the ‘Input Channel’ box. If you just want to output waveforms, without any input, then uncheck this box. If this box is checked, you must also select the channels to collect in the normal way (through ‘Channels to Collect’, p. 22).

Below this you can turn on or off the three possible output channels. The first two are D/A channels, channels which output an analog waveform based on the values given to them. The third is a digital output channel, which outputs an 8 bit digital signal based on the values given to it. Next to each of these is a pop-up menu, which allows you to select a trace to be output on that channel from all the currently open traces. Each of these output channels which are checked will output the values in the trace indicated to its right each time the 'Get Data' command is performed.

The traces are output at the same sampling rate as data is collected. This is true regardless of the sampling rate of the output trace. For instance, if you're using a trace for output that was collected at 5 kHz, but have now set the sampling rate for data collection (using 'Sampling Rate' in the 'File' menu, p. 21) to 1kHz, then the data in the trace will be output at 1000 samples per second, or 5 times slower than it actually is. See Making Waveforms for Output below for more information on this.

To actually output the waveform, select the 'Get Data' function from the 'File' menu. If you are collecting data as you are outputting a waveform, then you will get the normal sequence of dialog boxes for data collection, as described in Digitizing Data, p. 30. Otherwise, you will only get a subset of the data collection prompts which apply to data output.

GW: Whether or not you are collecting data as you output waveforms, a dialog box will appear asking how long you want to collect data for. This will also be the length of time for which the data from the trace is output. Note that the trace must have enough data in it to last this amount of time, or else you will receive an error message saying that there is not enough data in the trace. If you are just outputting data, the program will then wait for you to press <return> or <enter> on the keyboard before starting to output the waveform.

NI: Whether or not you are collecting data as you are outputting, you will start the output by pushing either <return> or <enter> on the keyboard, and stop it by pressing any key on the keyboard. If the time between starting and stopping the output is longer than the amount of data in the trace, then when the end of the trace is reached, it will be repeated from the beginning. Spike Studio will keep repeating the trace every time the its end is reached, until you press a key to stop the output.

Analog Waveforms

When you output a waveform on one of the analog output channels (D/A 1 or D/A 2) then the voltages seen on the output channel is the same as the voltage which you see on the trace in Spike Studio (you can use the Time/Voltage tool, p. 92).

However, no inversion, rectification, or correcting of offsets is done when outputting the trace. If you have done any of the above and want to output the trace as it appears on the screen, then you can use the 'Save Digitized Data' command to save it out modified as it is on the screen (p. 121), and then 'Open' the new trace file.

Digital Waveforms

When we talk about digital output, we have to switch from talking about voltages to talking about bits and bytes. The digital output ports that Spike Studio uses are 8 bit ports, meaning they have eight wires coming out of them, each of which can be on (+5 volts) or off (0 volts). When you output on one of these ports, you pass it a number between 0 and 255 which tells it which of the bits to turn on, and which to turn off. A 0 means turn all the bits off, and a 255 means turn them all on. For numbers in-between, you must convert the decimal number into a binary number, which then tells you which bits will be on and which will be off.

The binary numbers present a couple problems when trying to output from Spike Studio on a digital port. The first problem is that Spike Studio stores each data value in 2 bytes, or 16 bits. Thus the digital output port can only use half of each data point in a trace. Spike Studio truncates each data point to 8 bits before outputting it on a digital port, using the low byte. In practical terms, what this means is that when you make a trace to output on the digital port, you should only have numbers in it between 0 and 255, and none higher, since anything larger than 255 will be cut to a number less than 255.

The second, and more serious problem is that there is no direct way within Spike Studio to see the data values as numbers between 0 and 255. Instead, Spike Studio only shows you voltages. So you must have a way of converting between a voltage and the digital number representing that voltage. The way to do this is to multiply the voltages by a fudge factor which represents the voltage scale being used internally in Spike Studio. The fudge factor depends on the voltage range of the board, as set by the 'Set Board' function. If you collected the trace with a 10 volt interval, then the fudge factor is 409.6. If you are collected with a 20 volt interval, then the fudge factor is 204.8 (these fudge factors will be difference if you are using the sound board as the input device). When you read a voltage out of Spike Studio (using the Time/Voltage tool (p. 92), for instance), then multiply by the appropriate fudge factor to get the data value which will be sent to the digital port.

For those of you who are technically minded, these fudge factors come from the fact that the digitizing boards are collecting 12 bit data, and these 12 bit values are spread over either 10 or 20 volts. 12 bits can contain numbers from 0 to 4095, so there are 4096 possible values for each data point. $10/4096 = 1/409.6$, and $20/4096 = 1/204.8$, and to get from digital values to voltages, Spike Studio multiplies by these factors. To go the other way, you must take the inverse of these factors, and multiply by the voltage to get the digital value.

Making Waveforms for Output

The easiest way to make waveforms for output is to use a spreadsheet to construct the waveform, save it out as text, and load it in using the ‘Open Text’ function (see Opening a Text Data File, p. 49). Make the waveform as a column of numbers, with each number being one sample. There are two things you have to be careful of when constructing the waveform. The first is the scale of the numbers, and the second is the sampling rate.

If you are outputting waveforms on an analog output channel, then each number should just be the voltage which you want to see on the output channel. Spike Studio will take care of converting the numbers appropriately. However, if you are outputting on the digital output channel, then you must construct the voltages in such a way that Spike Studio will convert them into the appropriate digital numbers. See Digital Waveforms above for an explanation of how this works. For instance, if your board is set to collect data between -5 and 5 volts, then the appropriate fudge factor is 409.6 (see Digital Waveforms, above). Thus, if you want to turn on all the bits in the output port, meaning you want to send it a value of 255, then you need to solve the following equation:

$$x * 409.6 = 255$$

This gives a voltage of 0.623, so this is the number you put in your spreadsheet.

The sampling rate is a little more tricky. Since Spike Studio does not pay attention to the sampling rate of the trace which it is outputting (see *Outputting a Waveform* above), you have to construct your text file with the appropriate number of samples per second. For instance, if you want to output a square wave with a frequency of 10 Hz, and you will be outputting it at 2 kHz, each cycle of the square wave must be $2000 \text{ samples} / 10 = 200$ samples long. Another way of looking at this is that at 2 kHz, each sample lasts $1/2000 = 0.5$ milliseconds. So to get a cycle of the square wave to last 100 milliseconds, you must have $100 / 0.5 = 200$ samples for that cycle.

One problem with this method is that if you use the same trace and output it at different sampling rates, then the frequencies of events in the output waveform will change. If you took our square wave from above and output it with a sampling rate of 4 kHz, it would change from a 10 Hz square wave to a 20 Hz square wave. This means that if you use a variety of sampling rates, you will need a corresponding variety of output traces. This is a little bit of a pain, and perhaps could be changed in the future.

One advantage to doing things in this way is that whenever you collect some data, you can immediately output that data, and the output waveform will look exactly like the input waveform which you just collected. This can be a quite powerful capability in some applications.

Outputting Waveforms While Triggering

You can output waveforms while using Spike Studio's triggering capabilities. In this case, the output will not start until the trigger is seen. If you are using multiple triggering, then the output will restart from the beginning of the output traces after each trigger, and there will be no output at times when there would be no data collection, as described in the section Multiple Triggering (p. 27). Note that you can perform triggered output without any input using an external trigger, but to trigger off of one of the input channels, you probably have to be collecting data from it (I haven't tested this to see for sure).

Initialization of Output Channels

If you are using either a National Instruments or GW Instruments board, and have saved this fact in the Configuration, then each time you start up Spike Studio, the output channels of the board will be initialized to 0. The analog output channels are initialized to 0 volts and the digital output channel is initialized to 0 on all 8 bits.

Display

Displaying Data

Spike Studio includes a very flexible display for the digitized data traces. You can display up to 16 data traces at once, and for each you can rectify it, invert it, correct offsets, change the gain, and change the temporal resolution. You can also print the data as it appears on the screen, or export it to a drawing program, using Spike Studio as a chart-recorder substitute. In addition, Spike Studio has a nice page layout feature which allows you to print multiple traces and graphs neatly on a page. The menu items which perform all of this are in the 'Display' menu, except for the printing and export functions which are contained in the 'File' and 'Edit' menus. In addition, some parts of the display can be controlled from the keyboard with the parameters (see Changing Parameters, p. 9) or with the mouse (see Zooming in on Data, p. 56).

None of the display functions affects the actual data stored on the disk, so you do not have to be afraid of experimenting with different ways of viewing the data. The way the data look on the screen will, however, effect the way they are analyzed (see Effect of Display on Analysis, p. 60).

Opening a Digitized Data File

There are a number of ways to get a digitized data file into the program. After using the 'Get Data' command, the first channel of the data which you just collected will be loaded into the program automatically. Alternatively, you can use the 'Open' command in the 'File' menu to load any file of digitized data into the program. A standard Macintosh file requester will come up asking what file you want to load in. The program does not check whether the file is actually a digitized data file, so if you have garbage on the screen check to see if you loaded in the right file. A third way to load in data is given below under Channel Number.

Every time you reload a data file which you have previously looked at, Spike Studio sets the display to look like it did the last time you were examining that data file. The program remembers whether the data were rectified or inverted and whether they were being clipped, as well as the gain and offset (see below for descriptions of all these). It also remembers the sampling rate (shown in the Sampling Rate info item, p. 122) and which channels were collected from when that data was digitized.

Opening a Text Data File

When Spike Studio collects data, it always saves it as a digitized data file which can be opened as described above. Other programs which collect data from digitizers will generally save the data in a similar form (see Final Data, p. 38). However, you may occasionally run into data which is not in a form that Spike Studio can read, and yet which you want to analyze using Spike Studio. In this case, if you can save the data from the other program in a text format, then there is a function in Spike Studio called 'Open Text' which can read in the text formatted data and convert it into a normal Spike Studio data file. A text format is a format where the numbers are spelled out with the characters "0"- "9" (i.e. something that you could look at in a word-processor). Usually programs which can save in text format will have an option in their save dialog boxes where you can select 'Text'.

This ability to read data in a text format is particularly useful if you are using the D/A capabilities of Spike Studio to output waveforms (p. 44), and want to design a waveform in another program such as a spreadsheet. In this case, you can make a column of numbers in the spreadsheet which describe the waveform, save this out in text format, and then use the 'Open Text' function to read it into Spike Studio.

To read in a text data file, select 'Open Text' from the 'File' menu. A normal Macintosh file requester will appear, where you should select the text file you want to read in. This text file may contain numbers in the form #, #.#, #e#, #E#, #.#e#, or #.#E#, with "-" signs as appropriate, where # can represent any string of numbers ("0" through "9"). Each number should be separated from the next number by a non-numeric symbol, or else by being on a separate line. This covers most text formats, and so Spike Studio can read the text files saved out of most other programs.

Once you have selected the text file, Spike Studio will ask you for a filename where it should save the data in a Spike Studio file. The text file will then be converted to a Spike Studio file, and saved in the filename you indicated. The Spike Studio file will then be opened, and can be treated as any other trace file.

When converting from the text file to the Spike Studio file, the program must assume a sampling rate and board gain to be able to correctly interpret the file afterwards. These will be taken from the current sampling rate and board gain, set with the Sampling Rate (p. 21), Set Board (p. 21), and Get Data (GW only, p. 23) commands. Thus, before you use 'Open Text', you should set the sampling rate and board gain appropriately (setting the sampling rate is not important for output waveforms).

For more information on using these files for outputting waveforms, see the Outputting Waveforms chapter (p. 40).

General Screen Organization

There are always three windows open in Spike Studio, one each for the parameters, the icon menu, and messages. These will usually be towards the bottom of the screen. In addition, there can be windows which hold traces, graphs, or the data storage area.

You can have up to 16 traces open at once within Spike Studio. Each trace which is opened is given its own window, and has its own set of parameters, events, and display modifications such as rectifying. Thus each trace is essentially independent of other traces in the program. The exceptions to this independence are the start time and time scale, which are kept the same across all traces. Thus, if you are working on one trace and set it to display 5 seconds of data, starting at 3 seconds into the file (see Time From Start and Seconds per Window below), then all other traces which are open will be moved so that they are displaying data starting at 3 seconds into the file, at a horizontal resolution corresponding to the 5 second display of the first trace. This means that it is always possible to make direct timing comparisons between traces on the screen.

In addition to the 16 traces, you can also have up to 10 graphs and a data storage area open, each within its own window. The 'Windows' menu has a list of all the open windows and lets you pick between them, as described in Changing the Active Window below.

For those of you who are technically minded, note that Spike Studio takes into account sampling rates in the time scales of the traces, so if two open traces were collected at different sampling rates, there will be more data points per inch displayed for one of the traces than on the other. As an example, say there are two trace windows open, the first window 5 inches across and containing a trace which was collected at 2 kHz, and the second window 10 inches long, containing a trace that was collected at 1kHz. If you select the first window and set 'Time from Start' to 0 and 'Seconds per Window' to 1, then the first window will display the first second of its trace, which contains 2000 samples. The second window will display the first 2 seconds of its trace, since its twice as wide as the first window, but this two seconds still only contains 2000 samples, since the sampling rate was half that of the first trace.

See Figure 1 (p. 8) and the chapter Orientation (p. 6) for more information.

Changing the Active Window

Either a trace, graph, or data storage area will always be the currently active window, the window that is front most on the screen and has a title bar that is not ghosted. This front most window is the one that you can work on at that point in time, and the functions available in the menus will reflect what type of window is currently active. Most Macintosh programs work in a similar way, so this should feel familiar (e.g. in a word-processor, you can only work on one document at a time, but can switch between many open documents).

To switch between windows, you can either click in the window you want to make active, or you can select the window from the 'Windows' menu. Either of these will then bring that window to the front of the screen, and make its title bar solid instead of ghosted. The available menu items will also change, as will the parameters if you switched to a trace window. If you ever perform a function without it having any effect in the appropriate window, one of the first things you might try is to make sure that the window is active. The 'Windows' menu has a check mark next to the window which is active.

Note that even when a non-trace window is the active window, the program still retains an 'active trace'. If a non-trace window is active and you select an event analysis function from the icon menu, change one of the parameters, or perform another function which should affect the active trace, then the last trace window which was active will be made active again, and the action you did will affect that trace.

Channel Number Parameter

The 'Channel Number' is the number of the channel from which the currently displayed data were collected. You can change which channel you are looking at by changing this parameter on the screen (see Changing Parameters, p. 9). This is equivalent to closing that window and then using the 'Open' command to open the file containing the next or previous channel of data. Only the channels which you collected together will be cycled through. In other words, if you collected channels 1, 3, and 4, then if you are looking at channel 1 and increment the 'Channel Number', it will be incremented to channel 3. If you want to change by more than one channel, say from channel one to channel four, press the arrow key more than once in rapid succession, or type in the number of the new channel. The program will then skip over the intervening channels and load in the channel you want. Whenever you switch channels, any analysis which you did on the old channel will be thrown away without any warning, so don't switch channels until after you have saved your analysis. If you change the channel by accident using the arrow keys, immediately push the other arrow key to get back to the original channel. If done fast enough, this will prevent a new channel of digitized data from being loaded in, and the analysis of the old channel will remain intact.

If the data currently loaded in are not stored in a file named <filename><number>, then the 'Channel Number' parameter will be replaced with a message saying "No Channel Number", and you will not be able to switch channels using this parameter. You then must use the 'Open' command to switch channels (see Opening a Digitized Data File above). Also beware that if you used Multiple Triggering with separate files saved for each trigger (p. 27), then the channel number parameter will cycle through triggers, not through channels.

Drawing and Redrawing Data

Whenever you do something which should change the display, Spike Studio will normally automatically redraw any parts of the display which might have changed appearance. However, there are times when you might not want everything redrawn all the time. The primary reason you might not want things redrawn is that it can take a while, and if you are making a lot of changes you may sit around quite a bit waiting for things to redraw. Because of this, Spike Studio includes a command which disable parts of the automatic redrawing, and another command which will force a redraw.

Whenever you change either the ‘Time From Start’ or ‘Seconds per Window’ parameters (see below), this changes the display of all open traces. If you have several traces open, but are only working on one at a time, you may want to only have the trace you’re working on redisplay as you move back and forth in it. You can do this by unchecking the ‘Always Redraw All’ item in the ‘Display’ menu. When ‘Always Redraw All’ is checked, then all open traces will be redrawn each time ‘Time From Start’ or ‘Seconds per Window’ is changed. When it is off, then only the active trace will be redrawn. You can then force the other traces to redraw as described below. There is a big note of caution about not redrawing all the traces. Even if ‘Always Redraw All’ is unchecked, so the traces are not being redrawn, any operations which affect that trace will assume that its at the current ‘Time From Start’ and has the current time scale. This means that *the display will not correspond to the data* which will be used in operations on that trace. So when ‘Always Redraw All’ is off, make sure you redraw other traces before you perform an operation on one of them.

Forcing Traces to Redraw

Sometimes you will want to force all the traces to redraw. The most usual reason for this will be because ‘Always Redraw All’ is unchecked, but there might be other situations where you would like everything redrawn. You can force all the traces to redraw by selecting the ‘Redraw All’ command in the ‘Display’ menu. All traces will then be redrawn according to the current drawing parameters. You can also make just the active trace window redraw by pressing ‘r’ on the keyboard.

Positioning of Windows on the Screen

You can move and resize all the windows in Spike Studio just as you would any window in a Macintosh program. You can also have the program automatically arrange the windows for you, as described below in Tiling Windows. In addition, if you have arranged windows in the program in a way that pleases you, this arrangement can be saved in the configuration so that each time you use the program this arrangement of windows will be used. See Saving the Configuration (p. 120) for more information on this. Even without saving window positions to the configuration, Spike Studio remembers a little about where you have positioned your trace windows as the program is running. When you close a trace window and then open another trace, the new trace window will be in exactly the same position as the trace window you just closed.

Tiling Windows

If you have a lot of windows on the screen and its getting kind of messy, you can tidy the windows up using the 'Tile Windows' function. This function will move the Parameters, Icon Menu, and Info windows to the bottom of the screen, and then arrange all the other windows into a number of columns which you specify, so that all windows are the same size and none are overlapping. To tile the windows, select the 'Tile Windows' function from the 'Display' menu. A dialog box will ask how many columns you want the windows put into. Type in the number of columns you want and click 'OK'. The windows will now rearrange themselves into that many columns.

Time From Start Parameter

The 'Time From Start' parameter governs what part of the trace you are looking at. The number is how many seconds into the data the left side of the window is, with 0 seconds being the beginning of the data in that file. This variable is one of the parameters on the screen (see Changing Parameters, p. 9). When you use the arrow keys to change the 'Time From Start', you move backwards and forwards by one screenful for each push of the arrow. If you have digitized a very long stretch of data, longer than will fit into memory at once, there may occasionally be a delay when you switch what position of the data you are looking at while the computer loads a new page of data into memory.

Note that, as described above in General Screen Organization, when you change the 'Time From Start' for the trace you are working on, it is also changed for all other open traces, so that all open traces always start at the same time.

Seconds per Window Parameter

The 'Seconds per Window' parameter is the number of seconds of data displayed across the width of the active trace window. It is similar in function to the sweep rate on an oscilloscope. The lowest setting (which depends on the sampling rate of the active trace) plots one sample on each column of points on the screen. Each time you increase 'Seconds per Window' by one notch (using the right arrow key - see Changing Parameters, p. 9), one more sample is plotted on each column on the screen, and a vertical line is drawn between the lowest and highest samples on each column (see Drawing Algorithm below for more information on this). If you set this too high, you may notice that trace is not drawn out to the edge of the screen, even though there are that many seconds in the data file. This is because there is not enough memory used for the trace to hold all the data (see Setting Memory Usage, p. 117).

When you change 'Seconds per Window' for the trace you are working on, it is also changed for all other open traces. The 'Seconds per Window' of each trace is set so their time scales are the same on the screen, meaning that the number of seconds of data displayed per inch is the same in each trace window. This means that you can visually compare time related things between different traces on the screen, even if they were collected at different sampling rates.

✕ Zooming in on Data

There may be times when you would like to more closely examine one section of a trace being displayed. You can do this with the 'Time From Start' and 'Seconds per Window' variables. This can be a little clumsy, however. The Set Position function is nicer, because it allows you to use the mouse to zoom in on a portion of the data.

Select the 'Set Position' function from the 'Display' menu or the icon in the icon menu. The mouse pointer will turn into an 'X' (as in x marks the spot). If you then click the mouse at some point in the trace, that point will be moved to the left edge of the window. If you click and hold the button down, you can highlight an area of the trace, and this area will be expanded to cover the whole window as closely as possible. Note that data cannot be expanded past one data value per horizontal position in the window.

When you use 'Set Position', it is exactly like changing 'Time From Start' and 'Seconds per Window', and all open traces will be affected as described above.

Overlapping Screens

There may be times when you are moving through a trace one window-full at a time, and would like to have the end of the last window-full of data show up as the beginning of the next window-full. For instance, if many of your events tended to fall between windows of data, this would enable you to see the whole event on at least one window-full. To get this overlapping of window-fulls of data, use the 'Set Overlap' command in the 'Display' menu. When you select this, a dialog box will ask how many seconds of overlap you would like. When you switch from one window full of data to the next, this number of seconds of data from the end of the last window-full will be placed at the beginning of the next window-full. Similarly when you move backwards through the data, the first seconds of the initial window-full will show up as the last seconds of the window-full before it.

Clipping the Data

In order to make sure that the trace fits in its display area, the data is normally clipped to that area. If a data value falls above or below the assigned area, the value is displayed at the limit of the area and a message appears saying that some parts of the data have been clipped. In a trace window, this means that data will never intrude into the area at the bottom of the window reserved for events. In outputting pictures, this means that the trace will not extend beyond the size of the window on the screen, and using the Page Layout function (p. 65) no data will extend beyond its assigned rectangle. Usually this clipping is helpful, because it prevents the display from getting cluttered with stray lines, but sometimes you may want the whole trace displayed without any clipping, especially if you are exporting the trace as a picture (see Outputting Picture below). In this case, you can check the 'No Clipping' item in the 'Display' menu. When 'No Clipping' is checked, no clipping of data will be performed on that trace. 'No Clipping' will slow down the display somewhat. If you normally want 'No Clipping' checked, you can save this in the configuration (see Saving the Configuration, p. 120).

Display Modification Functions

The following functions give you further control over the way data is displayed. They are all contained in the 'Display' menu. For the items which are either on or off, you can save their state in the configuration so that they are always set properly when the program starts up (see Saving the Configuration, p. 120).

Rectifying Data

Rectifying means taking all the digitized data below some baseline and flipping them up, so that the data below and above the baseline are superimposed on each other. For instance, if the baseline is at 0 volts, than any negative voltages will be changed to positive voltages when they are displayed. A value of -2 volts will be changed to 2 volts, etc.. Values above the baseline are left the same as they were.

To turn rectification on or off, select 'Rectify' in the 'Display' menu, which will check or uncheck it. The 'Auto Offset' or 'Offset' functions set the baseline around which rectification occurs (see Correcting Offsets in the Data, p. 59) Data is rectified both on the screen and when doing event analyses. Note that data cannot be both rectified and inverted. 'Rectify' only changes the active trace, and each open trace can be independently rectified or unrectified.

Inverting Data

Inverting the data simply takes all the digitized data below a baseline and flips it above, and takes all the data above and flips it below. Thus, if the baseline were at 0 volts, a -2 volts sample would become a 2 volts sample, and a 1 volt sample would become a -1 volt sample.

To invert the data, select 'Invert' from the 'Display' menu. 'Invert' will now become checked and the data will become inverted. To turn inversion off, select 'Invert' again. Data are inverted both on the screen and when doing event analyses. Note that data cannot be both rectified and inverted, so uncheck 'Rectify' before checking 'Invert'. 'Invert' only changes the active trace, and each open trace can be independently inverted or uninverted.

Correcting Offsets in the Data

When digitizing data, there may be an offset in the data. That is, the digitized data may not be centered around 0 volts, but instead may be shifted up or down. This can make viewing and analyzing the data difficult. To correct for this, there is an 'Auto Offset' function. 'Auto Offset' goes through the first 5 seconds of data and finds the average voltage. This average voltage is then subtracted from every sample, so that the data is shifted to be centered around 0 volts. To turn 'Auto Offset' on or off, select 'Auto Offset' in the 'Display' menu, which will check or uncheck it. When 'Auto Offset' is turned off, the offset will initially be set at 0 volts, so nothing will be subtracted from the data. Regardless of whether 'Auto Offset' is on or off, you can change the offset manually (see Changing the Offset below). 'Auto Offset' can be used with either inverted or rectified data. Changing the status of 'Auto Offset' also changes the gain. The offset is used when doing event analysis.

Note that each open trace has its own offset, and 'Auto Offset' can be used or unused independently by each open trace.

Changing the Offset

There may be times when you would like to correct for offsets in the data, but the 'Auto Offset' function is not correcting them properly. In this case, you can use the 'Change Offset' function in the 'Display' menu. This will bring up a dialog box which enables you to change the offset. The number of volts which you type in here will be subtracted from every data value before it is drawn, so a more positive number here will shift the data down and a more negative number will shift the data up. Note that this also affects the voltage reading that the program will give, since the data value associated with 0 volts will be changed (although voltages stay the same relative to each other). The offset is used when doing event analysis. Note that each open trace has its own offset.

Changing the Gain

The gain is analogous to amplification on an oscilloscope. It makes the digitized data look taller or shorter on the screen. The data values are either multiplied or divided by an integer. When a new data file is loaded in, the gain is automatically set based on the first five seconds of data, so that the data just fits in its window. When a data trace is loaded, a message will inform you of the gain setting.

If you want to change the gain, you can use the 'Change Gain' function in the 'Display' menu. This will bring up a dialog box which lets you change the gain. The heights of all the data points will be multiplied by this number before being displayed, so if you decrease the number, the trace on the screen will shrink, and if you increase the number, the trace will expand. The number is rounded somewhat, so the gain will be close to, but not necessarily exactly, the number you typed in. The gain does not affect any of the analyses. Each open trace has its own gain.

Effect of Display on Analyses

When data is being used for any of the event analyses, the data is rectified, inverted, and offset, if any of these are selected before the analysis is performed. Analyses which are done for one window-full of data use the data currently displayed on the screen, which is set by 'Time From Start' and 'Seconds per Window'. The gain does not effect any analysis, nor does clipping. None of the digitized data analyses (p. 92) are affected by anything in the display, except for the Time/Voltage function.

Drawing Algorithm

This section gives a detailed discussion of how Spike Studio draws data on the screen. You do not need to read or understand it to be able to make full use of the program.

The screen is divided into columns of pixels (the dots on the screen). Each window has a column a certain number of pixels tall, and there are a certain number of columns across each window. The number of samples drawn in the window is determined by 'Seconds per Window'. At its lowest level, one sample is drawn on each column in the window, but as 'Seconds per Window' is raised, there are more samples that are supposed to be displayed than there are columns, so more than one sample must be drawn on each column. The program does this by converting 'Seconds per Window' to a number called the spacer. The spacer is the number of samples which must be drawn on each column so that 'Seconds per Window' of data can fit in the window.

Spike Studio starts at the sample specified by 'Time From Start', and gathers all the samples up to 'Time From Start' plus spacer. That is, it gets all the samples which must fit on the first column of the window. From each of these samples, Spike Studio subtracts the offset. Then, if the data is being rectified, Spike Studio takes the absolute value of the sample. If the data is being inverted, Spike Studio multiplies the sample by -1. Next, for all of the samples being displayed on the first column, Spike Studio selects the largest and smallest voltages. It divides or multiplies these by the gain. Then it clips them (if clipping is selected) so that they fit within the window's height and don't overwrite the event display area at the bottom of the window. Finally it draws a vertical line in the first column in the window between the two points specified by those samples. This whole routine is then repeated for the other columns in the window.

Occasionally the 'Seconds per Window' of a trace may be set so that there is less than one sample per column of data. In this case, Spike Studio puts the samples in the appropriate columns, and draws lines between them.

Outputting Pictures

There are several ways to output pictures of traces and graphs from Spike Studio. You can copy pictures to the clipboard, save a picture to disk, or print a picture. You can also lay out a number of traces and graphs on a page, and copy, save or print this layed out page. Some of these methods output low resolution pictures, equivalent to the resolution you see on the screen. Others output high resolution pictures, at the resolution of your printer. In combination with the display functions, this makes Spike Studio as flexible as many chart-recorders and a useful tool for displaying and labeling data.

All the commands below will work on traces, graphs, and the stored data trace. For all of them except the Page Layout window, the commands work on whatever window is active. So you should first select the window with the picture you want to output, by clicking in it or by selecting it from the 'Windows' menu (see Changing the Active Window above), and then select the appropriate command below.

When exporting pictures of traces, you may want to turn off the clipping of data so that all the data is faithfully represented (see Clipping the Data above). Once the export is finished you can turn clipping back on.

Note that Spike Studio does not know how to deal with pictures, so once you output a picture, you cannot read it back into Spike Studio. Therefore, always save bursts and stored data traces using Spike Studio's 'Save' command in addition to outputting pictures. If you want to export the trace data as numbers, see Saving Digitized Data (p. 121).

A Note on High Resolution Pictures

As of now, I have not figured out how to make a high resolution PICT file that is properly read by a drawing program. The drawing program I have used in testing is Canvas, and the only way I have found to have it print a Spike Studio picture in high resolution is a little cumbersome. When Canvas reads in a high resolution picture from Spike Studio, the picture will be much larger than one page. A Page Layout picture (see below), will be several pages wide and several pages high. To work with this picture, you should shrink it to 25% on the screen (in canvas, you do this with one of the icons on the left of the picture, in other drawing programs this option may be in the menus). Then, when you print the picture, again shrink it to 25% of its size. In this way, you will get a one page, high resolution printout.

There are no problems like this when Spike Studio prints out high resolution pictures directly.

Copying a Picture to the Clipboard

You can copy a picture of a trace, graph, or the stored data trace from Spike Studio to the clipboard, from which you could then paste the picture into another program (such as a drawing program). The copies are at low resolution.

To copy a picture to the clipboard, first activate the window with the trace or graph you want to copy, by clicking in it to make it the front most window or by choosing it in the 'Windows' menu. If you are copying a trace, use the 'Time From Start' and 'Seconds Per Window' parameters to get just the part of the trace you want copied displayed in the window, and use the 'Set Gain' and 'Set Offset' menu items in the 'Display' menu to get the height and vertical position of the trace the way you want it. You may also want to turn off clipping, by checking the 'No Clipping' item in the 'Display' menu. If you are copying a graph or the stored data trace, then make the window exactly the size that you want the picture to be when you copy it. Then select either the 'Cut' or 'Copy' commands from the 'Edit' menu ('Cut' and 'Copy' are equivalent in Spike Studio). The program will wait a second while it does the copying, and then a picture of the trace or graph will be in the clipboard. You can then switch to another program, such as a drawing program, and Paste the picture in.

If you are copying a trace, a picture of the traces bursts will also be copied along with the trace.

Saving a Picture

Follow the instructions for Copying a Picture to the Clipboard above, but use the 'Save Picture' command in the 'File' menu instead of the 'Copy' command. When you select 'Save Picture', a standard Macintosh save dialog box will appear, where you should specify the filename where you want the picture saved. The picture will then be saved into that file at high resolution.

Printing

See Copying a Picture to the Clipboard above, but use the 'Print' command in the 'File' menu instead of the 'Copy' command. After selecting 'Print', a standard Macintosh printing dialog box will appear, and when you say 'OK' then the picture will be printed at high resolution.

Page Setup

The 'Page Setup' menu item in the 'File' menu will bring up the standard Page Setup dialog box. The only real use for this dialog box in Spike Studio is turn the page sideways when printing. This will allow a longer section of the trace to be printed. I'm not sure how, if at all, any of the other 'Page Setup' items will affect the output. Currently the program ignores them, but the printer may not.

Page Layout

Page Layout is a feature of Spike Studio which makes it easier to lay out a number of traces and graphs on a single page. The Page Layout command gives you an area on the screen which represents one printed page. You can subdivide this area into a grid of rectangles, and within each rectangle you can put a trace, a graph, a stored data trace, a time scale bar, or nothing.

When you use Page Layout to output pictures of several traces on a page, all the traces on that page will be lined up in time. This means that they will all start at the same time and use the same time scale, and so are directly comparable to one another. This is handy if you are making a figure of several traces which were recorded concurrently (e.g. the traces from multiple electrodes in a single prep). The stored data trace will be output with exactly the same time scale as the other traces, but will start at its time 0. With each trace, you can choose whether to also have the traces bursts and/or a voltage scale bar be output along with the trace, and if you want a voltage scale bar then you can specify its size. Graphs will be sized to fit within the rectangle.

You can then either copy, save, or print the picture which you constructed using Page Layout.

To use the Page Layout command, select 'Page Layout' from the 'File' menu. A new window will appear, with a bunch of buttons down the right side, and a rectangle on the left with some dashed lines running through it vertically and horizontally. These are all explained below.

Dividing up the Page

The dashed lines on the page divide up it up into several rectangles, each of which can contain a picture of something. You can add more horizontal dividers by clicking on the dashed line just above the page, holding the mouse down, and dragging the dashed line down onto the position on the page where you want a new divider. Similarly, you can get a new vertical divider by clicking on and dragging the dashed line from the right of the page into the position where you want a new vertical divider. You can move a divider by moving the pointer over it until the pointer changes to a double arrow, then clicking and dragging the divider to the position where you want it. If you want to get rid of a divider, all you have to do is move it off the top or right of the page.

Putting Pictures on the Page

To specify what you want in a given rectangle on the page, click on the rectangle. A dialog box will appear, where you can select what picture you would like in that rectangle, and specify some parameters if appropriate. First you must choose what type of picture you would like. You specify this with the radio buttons at the top of the dialog box, and the choices include Trace, Graph, Stored Data Trace, Time Scale, and Nothing. If you do not have a Trace, Graph, or Stored Data Trace open in the program at that time, then the corresponding radio button will be missing.

If you select Trace or Graph, then you must specify which of the available traces or graphs you would like in this rectangle. You do this in the pop-up menu that's just below the radio buttons. You will see the name of one of the traces or graphs. Click and hold down the mouse button on this name, and a menu will pop up with a list of all the traces or graphs currently open. Select the one you want to have put into this rectangle on the page.

For Traces you must specify whether you would like the bursts also put into the picture, whether you would like a voltage scale bar, and what size the voltage scale bar should be. Check or uncheck the appropriate check boxes, and if you want a scale bar, type in the size of the bar you want in volts, in the text field provided. Note that the gain and offset are copied from the gain and offset used in the trace window, so use the 'Set Gain' and 'Set Offset' commands to change them.

For a Time Scale, you will need to specify what length of time you want the bar to go. You specify this in the same place that you would give the voltage scale if you had selected a trace.

When you have everything set up the way you want, click OK, and you will see the name of whatever you just selected appear in that rectangle on the page.

If you want a trace or a graph to span more than one rectangle, then select the same trace or graph in consecutive rectangles. Any time the same trace or graph is selected in a consecutive rectangles horizontally or vertically then the picture which is output spans all those consecutive rectangles. If the same trace or graph is selected in consecutive rectangles both horizontally and vertically, then the picture will span a larger rectangle with the aggregate dimensions of all the consecutive rectangles.

Outputting the Layout

You can Copy, Save, or Print the layout using the appropriate buttons on the left of the Page Layout window. These commands work the same as described above, except that the resolution is determined by the 'High-Res' check box at the top of all the buttons. If 'High-Res' is checked, then the output will be at the resolution of the printer (300 or 600 dpi for most laser printers). If 'High-Res' is unchecked, then the output will be at the resolution of the screen (usually around 72 dpi). High resolution pictures on a nice laser printer are publication quality, but low resolution pictures will be dealt with much faster by a printer or drawing program. If you are saving or copying high resolution pictures, see A Note on High Resolution Pictures above.

Saving and Loading Layouts

It could be that you will set up a page layout, and then will want to reuse that same page layout several times. You can save a page layout using the 'Save Layout' button on the left of the Page Layout window, and then reload it using the 'Load Layout' button just above it. Save Layout will save a file containing the positions of all the dividers currently on the page. What is inside of each of the rectangles which the dividers create will not be saved.

Event Analysis

Event Analysis

Spike Studio includes two algorithms for picking out events from digitized data. An event can be a spike, a burst of spikes, an EPSP, a square wave from a stimulator, or any other signal which has a definitive onset and offset. Event analysis is in some sense the core of Spike Studio, and most of the analysis functions revolve around the events picked out. Currently, Spike Studio includes six types of analysis functions for events: finding onsets and offsets, counting spikes in a given part of the data, finding spiking frequencies, finding instantaneous frequencies, finding the peak-to-peak height of an event, and finding the integrated voltage within an event.

Note that when doing analyses, the data is analyzed just as it appears on the screen (see Effect of Display on Analyses, p. 60) . If rectification or inversion are selected, the data will be rectified or inverted when it is analyzed, and any offset will be subtracted from the data when it is analyzed. Note also that all commands below work on the currently active trace (the trace who's window is front most - p. 51), and that each trace has its own set of events, and its own parameters.

Spikes

Throughout this section of the manual you will be reading about spikes. A spike refers to a place in the data where the voltage goes up and then down. More precisely, a spike is an area where the data rises above the 'Lower Threshold' and then goes back below the 'Lower Threshold', without ever rising above the 'Upper Threshold'.

Automated Event Analysis

An event is a spike or a set of spikes which are close enough together that they make a continuous stream of electrical activity. This is more precisely defined using four parameters; Lower Threshold, Upper Threshold, Minimum Event Duration, and Minimum Interevent Duration (see Changing Parameters for information on how to change these). You can also set a minimum number of spikes in an event, and a maximum duration for a spike to be included in an event. There is a second event discrimination algorithm included in Spike Studio which does not use Minimum Event Duration or Minimum Interevent Duration, and its described below under An Alternate Event Finding Algorithm.

Lower Threshold

The 'Lower Threshold' is the voltage above which data is defined as significant. Any data below the lower threshold will not be considered significant during the analysis. As the program goes through the digitized data searching for events, it considers only the samples above the lower threshold. The threshold line is displayed visually as a horizontal line across the digitized data. You will probably raise or lower this line until it is just above any noise.

In addition to setting the lower threshold by changing the 'Lower Threshold' parameter, you can also drag the threshold line with the mouse, using the threshold positioning function. Select 'Position Threshold' from the 'Analysis' menu or click on the icon (pictured at the top of this section) in the icon menu. Then click on the lower threshold line, and while holding the mouse button down, drag the line to where you would like it and then release the mouse button.

↖ **Upper Threshold**

The 'Upper Threshold' is the voltage above which spikes are thrown out. If a spike goes above the upper threshold, it will not be counted as a spike during the analysis. As the program goes through the digitized data searching for events, if the voltage first goes above the lower threshold, then above the upper threshold, and then back below both thresholds, that entire piece of data will be treated as if it was below the lower threshold. The lower and upper thresholds can be used together to make a window discriminator.

In addition to setting the upper threshold by changing the 'Upper Threshold' parameter, you can also drag the threshold line around with the mouse by using the threshold positioning function (see Lower Threshold above).

Minimum Event Duration

The 'Minimum Event Duration' is the number of seconds which an event has to last before it is considered an event. Any activity which lasts for less time than the 'Minimum Event Duration' will not be counted as an event. Note that 'Minimum Event Duration' can be set down to 0.000 (using the arrow keys). When the number on the screen is 0.000, the minimum event duration is set to one sample or a couple samples (there are a few low values that will read 0). Any one isolated sample which goes above the threshold will then be counted as an event (this is useful for quick spikes). You will probably set 'Minimum Event Duration' to be a little less than the shortest event you want counted (set it low to find spikes).

Minimum Interevent Duration

The 'Minimum Interevent Duration' is the number of seconds by which two spikes must be separated in order to be considered two separate events. Thus two samples between the thresholds which are within 'Minimum Interevent Duration' of each other will be combined and considered one event. When the number on the screen is 0.000, the minimum interevent duration is set to one sample or a couple samples (there are a few low values that will read 0), and any one isolated sample that falls below the threshold will then separate two events. You will probably set Minimum Interevent Duration to be a little less than the shortest length of time between events in your data (set it low to find spikes).

Minimum Spike Number

The Minimum Spike Number is the minimum number of spikes which must be found inside an event for that event to be counted. The default is a one spike minimum, so that initially no events are thrown out for having too few spikes. This number can be changed using the 'Minimum Spike #' item in the 'Analysis' menu. When that item is selected, you can enter the minimum number of spikes you would like contained in an event. If an event contains fewer than the Minimum Spike Number of spikes, then it is thrown out.

Spike Studio includes a very simple filter for filtering out very slow, long noise in the digitized data. To activate it, select the 'Filter Spikes' function in the 'Analysis' menu. This will bring up a dialog box with a number and a check box. The check box turns the filter on and off, and is set to off when the program is started up. The number is the amount of time a spike has to last for it to be thrown out. Any spike which lasts longer than this amount of time will not be counted as a spike when doing an event analysis.

This function was written to get rid of slow, large shifts of the offset in the data. It does not work very well, and so is not integrated well into the program. It will not do anything when counting spikes or finding frequencies, for instance. We have not found it useful. Still, it's left in the program in case someone finds a use for it. A future version of the program may contain a standard filter to cut off low and high frequency waves in the data.

Calculating Events

To have the program calculate the events, there are two options. The 'Events on Screen' and 'All Events' commands in the 'Analysis' menu will calculate the onsets and offsets of the events in the data currently being displayed on the screen and all the events in this channel of digitized data, respectively. Pressing 'e' on the keyboard (for events) will also calculate the events on the screen.

The analysis control variables actually give a great deal of flexibility in picking out events of activity. Usually some trial and error is involved, setting the variables, calculating the events on the screen, then adjusting one or more variables and recalculating, until the proper settings are found. With a little practice, you will learn the effect of each variable and easily determine which variable should be changed in a given situation. In many cases, the Upper Threshold, Minimum Spike Number, and Simple Filter can all be ignored, leaving just three variables which must be adjusted during event calculation.

Midevents

In some situations, you may want to look at the middle of each event instead of looking at the onsets and offsets. If you check 'Midevents' in the 'Analysis' menu, then Spike Studio will save the middle point of each event instead of the onset and offset. The middle of the event is found by counting all the spikes in the event, and picking the middle of the middle spike. Only one line will appear for each event on the screen.

Note that when switching between onset/offset and midevents, all previous event data is thrown out. When using midevents, none of the other analysis functions (such as frequency or height) can be used. If you commonly use midevents, you may want to save it as part of the configuration (see Saving the Configuration).

Event Finding Algorithm

What follows is a detailed explanation of the algorithm used to define the onsets and offsets of the events (it is not necessary to understand this in order to use the program):

Spike Studio starts at the beginning of the digitized data and goes through samples until it finds one above the 'Lower Threshold'. This becomes the onset of the first event. Spike Studio then continues through the data until a sample drops below the 'Lower Threshold'. This becomes the offset of the first event. If between the onset and the offset, any sample was above the 'Upper Threshold' then this onset and offset are thrown out and Spike Studio looks for the next sample above the 'Lower Threshold' to become the onset.

Once it has a beginning onset and offset, Spike Studio continues examining the data. If it gets the number of samples away from the offset which is equal to the 'Minimum Interevent Duration', then this onset and offset become an event. If, however, it finds another sample over the 'Lower Threshold' before it gets 'Minimum Interevent Duration' away from the offset, then Spike Studio keeps looking through the samples until another sample drops below the 'Lower Threshold'. As long as no samples within this spike went over the 'Upper Threshold' then the sample which dropped below the 'Lower Threshold' becomes the new offset. If a sample in that spike did go over the 'Upper Threshold' then the old offset is retained. If the 'Simple Filter' is active, then each stretch of data which goes above and then below the 'Lower Threshold' is also checked to see if it is longer than the filter length, and if it is then that whole stretch is assumed to be below the 'Lower Threshold'.

This whole process of looking through the samples and resetting the offset of the event whenever samples go above and then below the 'Lower Threshold' but not above the 'Upper Threshold' keeps happening until no samples within 'Minimum Interevent Duration' of the offset go above the 'Lower Threshold'. At this point the onset and the offset of the first event are set. If the time between the onset and offset is more than 'Minimum Event Duration', and the number of times that the data went above and then below the 'Lower Threshold' but not above the 'Upper Threshold' is equal to or more than the 'Minimum Spike Number', then the event is stored and the search for the next event starts from the offset of this event. If the time between the onset and the offset is shorter than 'Minimum Event Duration' or the number of spikes in the event is less than 'Minimum Spike Number' then the event is thrown out and the search starts again at the offset. Spike Studio keeps looking for events until the end of the data piece which is being analyzed.

If the end of the data is reached while in the middle of an event, the end of the data becomes the final offset, and the event is counted if it is longer than the 'Minimum Event Duration'. No event is counted if it is within 'Minimum Interevent Duration' of the beginning of the data being used. Both of these actions are motivated by the assumption that onsets are more important than offsets.

When the program is finding midevents, the same event finding algorithm is used, but after an event is found, the number of spikes in that event are counted, the middle spike is determined, and the middle of the middle spike is saved as the event.

An Alternate Event Finding Algorithm

There is a second event finding algorithm in Spike Studio which is much simpler than that described above, and may be useful in certain situations. Its called the All in One event finding algorithm, and it makes one event which starts at the beginning of the first spike in the trace and ends at the end of the last spike in the trace. Thus, all spikes in the trace are put into one event. This could be useful if, for instance, you are looking at the response to a stimulus and all spikes in the trace are the response to on stimuli, so you want to analyze them together.

The All in One algorithm pays attention to the lower and upper thresholds, minimum number of spikes, the offset, rectification and inversion just as described above for the normal event finding algorithm. However, it does not pay attention to the Minimum Event Duration or the Minimum Interevent Duration. Instead, the algorithm starts at the beginning of the trace and looks for the beginning of the first spike which goes above the lower threshold but stays below the upper threshold. The beginning of this spike becomes the onset of the event. It then goes through the data looking for the last spike in the trace which goes above the lower threshold but stays below the upper threshold. The end of this spike becomes the offset of the burst.

To use the All in One algorithm, select the ‘All in One Event’ command from the ‘Analysis’ menu. An event will be generated as described above. When you save this event, make sure you say no when asked if you want to recalculate events, since the recalculation will use the normal event algorithm.

User Modifications of Calculated Events

While the algorithm which picks out events is quite flexible, and will pick out most events correctly, there will be some events which will not be picked out correctly. For this reason, Spike Studio includes several functions which allow you to modify the automatically calculated events. These include functions to add an event, to delete an event, to undelete a deleted event, to merge two events, and to remove all the user modifications of events. These functions are also useful for doing analysis on pieces of data which would not be picked out by the event discrimination algorithm. The event modification functions can be selected from the 'Analysis' menu or from the icon menu on the lower right of the screen. Shown next to the title of each function is the picture of the icon in the icon menu which will select this function.

Adding Events

If Spike Studio misses an event, you can add the event in by hand using the 'Add Event' function in the 'Analysis' menu. This function can also be selected by pushing 'a' on the keyboard. When you select 'Add Event', the cursor will become a leftward pointing arrow. When you press and hold down the mouse button, a line will appear among the event onset and offset lines. This line will mark the onset of the event you want to add. You can move the line to the place where you want the new event's onset to be by holding the mouse button down and moving the arrow until it points to the proper location. When you release the button, the onset is fixed, and the arrow will now point rightward. When you push and hold down the mouse button again a new line will appear for the offset. Again while holding the mouse button down, move the arrow to point to where you want the offset to be, and release the button. A new event is now defined, with the onset and offset you just specified, and you once again have a leftward pointing arrow so you are ready to add another event.

If you are using midevents, then you will place only one line at the center of the event, and that will define the new event. When adding midevents, the pointer will always be a leftwards pointing arrow.

Deleting Events

Spike Studio may pick out a event where there shouldn't be one. You can remove this event by using the 'Delete Event' function in the 'Analysis' menu. This function can also be selected by pushing 'd' on the keyboard. When you select this function, the pointer will change to a crosshair. Position the crosshair over the event you want to delete, and click the mouse button. The event will become ghosted (faint), the number inside the event will disappear, and a message will tell you that an event was deleted. If the event was a user added event, then the event will disappear entirely. In either case, the event will no longer be counted and will not be saved when you save the events. If there is both a user added event and a computer event at the same location, 'Delete Events' will delete the user added event first.

🖱️ Undeleting Events

Sometimes you may delete an event and later change your mind. If the event you deleted was a user event, you must put the event back again using the 'Add Event' function, since the original is completely gone. If, however, the event was a computer generated event, it will appear on the screen as a ghosted event, and you can reinstate it using the 'Undelete Event' function. Select the 'Undelete Event' function from the 'Analysis' menu or press 'u' on the keyboard. The pointer will become the normal Macintosh pointer. Point to the ghosted event you want to reinstate, and click the button. The event will fully reappear, a number will be placed inside of it, and a message will tell you that an event was undeleted. This event will again be counted, and will be saved when you save events.

→←Merging Events

If Spike Studio picks out two events where you feel there should only be one, you can merge the two events using the 'Merge Events' function in the 'Analysis' menu. This function can also be selected by pushing 'm' on the keyboard. This function is the equivalent of deleting the two events, and then adding a new event with the onset exactly at the position of the onset of the first event and the offset exactly at the position of the offset of the second event. When you select 'Merge Events', the pointer will change to two arrows pointing inward. Position the pointer between the two events you want to have merged, and click the mouse button. The two events will be deleted, and a new one will be added which has the same onset as the first one and the same offset as the second one.

Since the onset and offset is set based on the events that are being merged, merging can be more exact than deleting and then adding in a new event by hand. It is also more convenient since only one operation needs to be performed instead of four. Note that this will not work correctly if the onset of the first event or the offset of the second event are not the correct onset or offset for the new event. Note also that the old events are deleted, meaning that if one of them was a user-added event it will disappear entirely.

Getting Rid of User Modifications of Events

Whenever you change channels using the Channel Number parameter, or load in a new data file, all the added and deleted events are thrown away along with the calculated events. Occasionally you may decide that you would like to get rid of all your additions to or deletions from the data you're currently working on. There are two functions that will accomplish this for you. The 'Undelete All Events' function gets rid of all event deletions which you made. It is equivalent to going through all the data using the 'Undelete Events' function on each deleted event. All computer calculated events are then counted as events. This includes any events deleted as part of a merge operation. Remember that once a user added event is deleted, it is permanently gone, so no user added (and then deleted) events will be undeleted by this function.

The 'Delete User Events' function removes all user added events, including any events formed as a result of a merge operation.

When using the 'Undelete All Events' function, be careful about undeleting an event in the same place as you added an event. If this happens, there will be two events in the same spot, and both will be counted as events and saved. In a case like this, you should delete the user added event.

Number of Events

'Number of Events' is a piece of information which is displayed under the icon menu on the screen. It gives an indication of how many events are currently stored in the computers memory. 'Number of Events' equals the number of computer calculated events plus the number of user-added events minus the number of computer calculated events which have been deleted. After calculating 'All Events' this number should be accurate. At other times, however, it just gives an indication of how many events there are, since if you delete a computer event on one screen, then go to another screen and calculate events for the new screen, the computer calculated event on the previous screen is thrown away, but the deletion of that event is still stored.

Comparing Between Two Traces

There are many situations where the data in one trace needs to be analyzed relative to another trace. Spike Studio gives you several options for making these types of comparisons. First, the 'Time From Start' and time scales of all traces will always be the same, so that you can visually compare between traces by just lining one trace window up underneath another. The Page Layout function will allow you to do this more precisely in a print-out (p. 65). Another way of comparing between channels is to use markers (see Markers, p. 89), which let you store the onset times of events in one trace and use these onset times in an analysis of another trace. Creative combinations of markers with Artificial Events might also be productive. Stored events (p. 88) will let you use the events calculated on one trace to make histograms of frequencies and other calculations from another trace. Finally, while there is no direct way to copy events from one trace to another, you can save the events from one trace into a file and then use the Load Events function to load the events back into another file (p. 87). There are of course a lot of post-processing methods for comparing two channels. These might be areas of the program which could be expanded if someone had specific ideas or requests.

Calculating Information About Events

Once the events are calculated, you may want to find out more information about particular events. Five types of analyses can be performed: the number of spikes in an event, the frequency of spiking in the event, the average instantaneous frequency between spikes in an event, the peak-to-peak voltage difference of an event, and the integral of the voltages in the event can all be found. These functions can all be selected either from the 'Analysis' menu, or from the icon menu on the lower right portion of the screen. Shown next to the title of each function is the picture of the icon which selects this function in the icon menu.

Any of these pieces of information can also be saved for all events. See Saving Analyses of Events to find out how to do this.

#Counting Spikes in an Event

To count the number of spikes in an event, first define the onset and offset of the event, either by having the program calculate events or by using the 'Add Event' function. Then select the 'Number of Spikes?' function in the 'Analysis' menu or the icon in the icon menu. The pointer will become a number sign (#). Position the number sign within the event you want to analyze, and click the mouse button. A message will tell you how many spikes are between the onset and offset of that event, where a spike is defined as any section of the data where samples went above and then back below the lower threshold. Note that this number will not include spikes which went above the upper threshold.

^Spiking Frequency

The spiking frequency is the number of spikes in an event divided by the length of that event. To find the spiking frequency in an event, first define the onset and offset of the event, either by having the program calculate events or by using the 'Add Events' function. Then select the 'Spike Frequencies' function in the 'Analysis' menu or the icon in the icon menu. The pointer will become a sine wave. Position the sine wave within the event you want to analyze and click the mouse button. A message will tell you the average spike frequency.

Average Instantaneous Spike Frequency

The instantaneous frequency between two spikes is the reciprocal of the period of time between the two spikes. In other words, if all the spikes were separated by the same amount of time as these two, this would be the frequency at which spiking was occurring. Spike Studio calculates instantaneous frequency by subtracting the time of onset of the first spike from the time of onset of the second spike, and then dividing 1 by this time difference. Spike Studio will not use spikes which go above the upper threshold in calculating frequencies.

To find the average instantaneous spiking frequency in an event, first define the onset and offset of the event, either by having the program calculate events or by using the 'Add Event' function. Then select the 'Inst Spike Freq.' function in the 'Analysis' menu or the icon in the icon menu. The pointer will become a square wave. Position the square wave within the event you want to analyze and click the mouse button. A message will tell you the average instantaneous spike frequency. To find all the instantaneous spike frequencies within an event, see Saving Spike Frequencies for One Event below.

Event Peak-to-Peak Voltage Difference

To find the voltage difference spanned by an event, first define the onset and offset of the event, either by having the program calculate events or by using the 'Add Event' function. Then select the 'Event Height' function in the 'Analysis' menu or the icon in the icon menu. The pointer will become two arrows, one pointing up and one pointing down. Position the arrows within the event you want to analyze, and click the mouse button. A message will tell you peak-to-peak voltage.

The voltage difference is calculated by going through all the samples between the onset and the offset of the event and finding the sample which has the lowest voltage and the one with the highest voltage. The lowest voltage is then subtracted from the highest, and this difference is the peak-to-peak voltage. If the data has been rectified, the lowest voltage in the event is assumed to be 0. Hence for rectified data, the peak-to-peak voltage is really the maximum voltage above the baseline, where the baseline is defined by the offset.

Note that the 'Event Height' function will not work well for single unrectified spikes, since the event will only include the part that is higher than the lower threshold, but will not include the part of the spike that goes below the lower threshold. Thus the minimum voltage in the spike will be wrong, and the only useful information given by this function will be the relative heights of the spikes. The user can get around this problem by using the 'Add Event' function to add in events by hand that include all of the spike, or by rectifying the data, or possibly by using the 'Artificial Events' function.

Note also that 'Event Height' ignores the upper threshold.

5.1 Event Integral

To find the integrated area under an event, first define the onset and offset of the event, either by having the program calculate events or by using the 'Add Event' function. Then select the 'Event Integral' function in the 'Analysis' menu or the icon in the icon menu. The pointer will become an integral symbol. Position the symbol within the event you want to analyze, and click the mouse button. A message will tell you the integrated area.

The event integral is calculated by going through all the samples between the onset and the offset of the event and summing their voltages. This produces a number which has units of [volts * (sampling-rate-period seconds)]. Note that the offset, rectification, and inversion of the data will all affect the integrand, as will the sampling rate. These will not change the relative integral values within one file, but they will change the absolute values and make it difficult to compare files.

Therefore, when you intend to calculate integrals, you may want to always use the same offset value, use the same sampling rate, and either always or never rectify or invert the data.

Note that 'Event Integral' ignores the upper threshold.

Saving Analyses of Events

Once you have all the data analysis variables set correctly and have made any event additions or deletions necessary, you will want to save information from the events to a file. It is possible to save all of the following information for each event: the onset and offset; the number of spikes; the spiking frequency; the average instantaneous frequency; the instantaneous frequency between each two spikes; the instantaneous frequency between each two spikes, with a time stamp for each spike; the peak-to-peak height; and the integral.

The procedure for saving any of these is the same, except when saving all instantaneous frequencies for one event. First select which types of analysis you would like to save using the 'Save Options' function. Then save them using the 'Save' function.

Save Options

Use the 'Save Options' function in the 'File' menu to pick which types of analyses you would like to save. When you select 'Save Options', a dialog box will appear with a list of all the analysis types which can be saved. Check the boxes next to the ones you want saved, and uncheck the boxes next to the ones you don't want saved. If you always save the same information, you can have this stored in the configuration (see Saving the Configuration).

If you are using midevents, only the time of each event can be saved. None of the other analyses makes sense with regard to midevents.

Saving

To save some analyses of the events, use the 'Save' function in the 'File' menu. When you select this command, a dialog box will ask whether you want to recalculate events. If you say no, only the events which are currently in the computer will be used. This means that if you have just calculated events for the screen, only the events on that screen will be used, along with any user added events in other parts of the data. If you say yes, (that events should be recalculated), the program will calculate events for all the data in that file, as if you had selected the 'All Events' function before saving the analyses.

Next the standard Macintosh saving dialog box will request a name and location for the file containing the first type of analysis you want to save. The analysis will be saved in the file you specify. Another standard Macintosh save dialog box will come up for the second type of analysis which you want to save, and so on. If you decide not to save a certain type of analysis for this file, simply click cancel in the save dialog for that type of analysis.

The saved files will be text files, each containing a column of numbers. The accuracy of the numbers will be related to the sampling rate (for time values) or the board gain (for voltage values). For sampling rates less than 1kHz, time values will be saved to the nearest millisecond, for sampling rates between 1 and 10kHz they will be saved to the nearest 100 microseconds, and for rates above 10kHz the accuracy will be to the nearest 10 microseconds. Voltage values follow a similar scheme, based on the voltage range of the board and the board gain. Frequencies are always saved with two numbers after the decimal place. All these files can then be read into a graphing or analysis program (I have a custom analysis program available for anyone who is interested). In addition, the files can be read into a word processor.

For onset-offset files, two columns of numbers are stored. The first column contains the onsets and the second column contains the corresponding offsets (see Filtering Event Files to separate onsets from offsets). All onsets and offsets are in seconds relative to the beginning of the data file. For number of spikes, spike frequency, average instantaneous spike frequency, event heights, and event integrals, the first number in the column is the value for the first event, the next number is the value for the second event, etc.. For spike frequencies with time stamps, the first number is the onset time of the first spike, the second number is the instantaneous frequency between the first and second spikes, the third number is the onset time of the second spike, the fourth number is the instantaneous frequency between the second and third spikes, etc., with the onsets and frequencies arranged in two separate columns. Note that no instantaneous frequency is calculated between the last spike of one event and the first spike of the next event.

If you are using midevents, the only analysis that can be saved is the time of each event (check Events in Save Options). In this case, the time of the middle of each event, relative to the beginning of the data file, is saved as a column of numbers in a text file.

Saving to Clipboard

In some instances it may be more convenient to save data to the clipboard instead of to a file. If you save the data to the clipboard, you can 'Paste' it directly into another program, saving yourself a few steps. In order to save data in the clipboard, use the 'Copy Data' command in the 'Edit' menu. When you select this command, a dialog box will ask whether you want to save data from the traces or from a graph. Select 'Traces'. Data will now be saved as text in the clipboard, in exactly the same format as if it had been written to a file.

Only one type of data can be saved to the clipboard at a time. The type of data is determined with the 'Save Options' menu item (see Save Options above). The first option set will be the one that is saved to the clipboard. For instance, if Number of Spikes, Spiking Frequency, and Event Height are all selected, then Number of Spikes will be saved to the clipboard.

Note that if you happen to have a file in the folder you're using named 'FooBazooKalamazoo' it will be erased when you use the 'Copy Data' function.

Saving Instantaneous Frequencies for One Event

There may be times when you want to know the instantaneous frequencies between spikes within a single stretch of data or a single event. In this case, you can use the 'Save Freq. - Event' function in the 'File' menu. First define the onset and offset of the event, either by having the program calculate events or by using the 'Add Event' function. Then select the 'Save Freq. - Event' function. The pointer will become a picture of a disk with a square wave in it. Position the pointer within the event for which you want to save instantaneous frequencies and click the mouse button. A standard Macintosh save dialog box will request the name of the file to save the frequencies in. The instantaneous frequencies between each two spikes in the event are then saved sequentially in that file, in the same format as is discussed above in the Saving section. No time stamps will be saved with the frequencies.

Loading Events

Sometimes you may wish to reload previously saved events to check where they appear on the trace. You can do this using the 'Load Events' function in the 'File' menu. When the file requester comes up, select the event file that you had previously saved for this trace, and the events in that file will be reloaded. The events are stored in the program just like calculated events. If Midevents is checked when the events are loaded in, they will be assumed to be midevents instead of onsets and offsets, otherwise they will be assumed to be onsets and offsets, so make sure Midevents is checked or unchecked correctly (see Midevents above). Note that some events may be off by one sample since a little rounding of the event times may happen as the onsets and offsets are saved.

Recalling Parameters Used to Calculate an Event File

When you load in an event file which was made by Spike Studio, the program will try to set up everything exactly as it was when you originally calculated the events. It will set the threshold levels and the minimum event and interevent durations as they were set when the events were saved, and will also rectify, invert, or offset the data to make it look exactly the same as when you did the original calculation. This is useful if you want to go back and look at the parameters you used to calculate the events in a certain file. (For the programmers among you, this info is stored in the resource fork of the file).

Storing Events

In order to use some of the graphing functions, you need to store a series of events. For instance, when making histograms, these events are used as the bins for the histograms (see Graphs). In Spike Studio lingo, these are known as stored events. There are two ways to make stored events. One way is by copying the events from the screen, and the other is by making a series of regularly spaced events.

To make stored events from regular events, first define the onsets and offsets of the events, either by having the program calculate events, by using the 'Add Event' function, or by using the 'Artificial Events' function. Then select the 'Store Events' function from the 'Extras' menu. A dialog box will appear, in which you should check the 'From Screen' box. Click OK, and the events from the screen are then stored. Note that 'From Screen' does not mean only the events which are currently visible on the screen, but rather all events which are currently calculated for this trace.

Alternatively, you can store a series of regularly spaced events. Select the 'Store Events' function from the 'Extras' menu. In the dialog box, uncheck the 'From Screen' box. Then enter the time at which you want your events to start, the time at which they should end, and the length of each event, into the appropriate boxes on the right. The program will store a series of events starting at the 'Starting' time, each 'Length' seconds long, until no more will fit before the 'Ending' time. The events will be consecutive - the onset of each event will be at the offset of the previous event.

Markers

Spike Studio includes a method of analyzing data on one channel in relation to data on another. This is accomplished through the use of markers. A marker is simply defined as a time in the data, similar to an event onset. You can define a set of markers on one channel of data, and then load in another channel and use these markers when analyzing the data from the new channel. Spike Studio includes three types of marker analyses: counting the number of spikes after each marker, adding together the data after each marker, and averaging the data after each marker. You can also use markers with the Artificial Events function (see Artificial Events) to give them more versatility. Data addition and averaging will be discussed in the Digitized Data Analysis section of this manual.

Markers are especially useful if you record stimulations of your preparation on one channel, and want to know the response of your preparation after each stimulation, as recorded on another channel. In this case, you can go to the channel which is recording the stimulation, set a marker point at each stimulation, and then switch to the response channel and do a marker analysis.

Storing Markers

A set of markers is defined by using the 'Store Markers' command in the the 'Extras' menu. Load in the channel of data with which you want to define the marker points. Then make a set of events whose onsets are at the positions where you want the markers to be (or whose midpoints are at the marker positions if you are using midevents). You can make these events using any of the event calculating procedures, and by using the 'Add Event' and 'Delete Event' functions. Now, select the 'Store Markers' command. The time of each event onset will be stored as a marker, and a message will appear telling you the number of markers which were stored.

Latencies and Number of Spikes After the Markers

You can save the time from a marker to the first spike, and the number of spikes after each marker using the 'Marker Spike Count' function in the 'Analysis' menu. First store the appropriate markers (see Storing Markers) and load the trace you want to analyze. Then select 'Marker Spike Count'. A dialog box will ask how much time to examine after each marker. This is the period of time in which spikes will be counted after each marker. Next, a save dialog box will ask where you want to save the latencies. Spike Studio will save a column of numbers into this file whose value are the time from each marker point until the next data point which rises above the 'Lower Threshold'. In other words, it is the time from the marker to the first spike after the marker. Spike Studio will only look the number of milliseconds you have specified for the first spike, and if there are no spikes within that period a -1 will be written to the file for that marker. If you do not want to save latencies, press Cancel.

Then another save dialog box will ask where you want to save the number of spikes. Spike Studio will then go to each marker point and count the number of spikes contained in the length of time you specified after each marker. These spike counts will also be saved as a column of numbers in a text file, with the first number being the number of spikes after the first marker, the second number the spikes after the second marker, etc.. The numbers in both these files will be saved in a format identical to that used in other save functions (see Saving, p. 85).

Digitized Data Analysis

Digitized Data Analysis

In addition to its event analysis features, Spike Studio has analysis functions which are used on the digitized data itself. These include a simple function to just find the time and voltage at a certain point in the data displayed on the screen, a few functions to add, subtract, and scale data, and a function to average data.

?Finding Time and Voltage in the Data

You can find the time a particular event occurred or the voltage at a particular place in the data by using the 'Time/Voltage' function in the 'Analysis' menu. When you select this function from the 'Analysis' menu, or click on the icon in the icon menu, the cursor becomes a question mark. Move the cursor near the position for which you want the time or voltage, then press and hold down the mouse button. A vertical line will appear, which extends up through the trace, and a display will appear in the lower right corner that looks like:

T: 10.023 V: 0.003|0.020

The first number is the time at the position of the vertical line. The second and third numbers are the minimum and maximum voltages of the data which fall on that vertical line, relative to the offset. The time given is the initial time of that column of data (see Drawing Algorithm, p. 61) (if you don't know what this means, don't worry about it).

Data Arithmetic

Spike Studio has functions for adding, subtracting, dividing, and averaging digitized data. These are accomplished by setting up a 'data storage area'. Data can be added to or subtracted from this storage area, and the data in the storage area can be scaled by a certain number. Using markers, an average of a number of sections of data can be put into the storage area. The data in this storage area can then be saved to a file, and opened like any other data file. All of the items which perform these functions are in the 'Data' menu, and the 'Add Data' and 'Subtract Data' items also have icons in the icon menu which appear when you set up a storage area.

All of these functions work with a certain length of data. While you specify this length in milliseconds, it is actually stored in the program as a number of samples. Thus, do not try to add data collected at one sampling rate to data collected at another sampling rate. Similarly, you should not add data collected with one board gain to data collected with another. (Although you might go through a two-step process where you first scale the data from one file before adding data from the second).

Since all of these functions work with the digitized data directly, the way the data is displayed on the screen will not affect these functions. This means, for instance, that offsets will not be corrected.

Setting Up a Data Storage Area

To set up a data storage area, you use the 'Setup Data Storage' function in the 'Data' menu. When this function is selected, a dialog box will ask what length of time should be stored in this area. Enter the length of data (in milliseconds) that you want to store. A section of memory is now set aside to store additions, subtractions or averages of data, and all values within this area will be set to zero. This chunk of memory is called the 'data storage area'. A window will also appear where the data in the data storage area is shown. This is called the 'stored data trace'. Whenever you want a fresh data storage area, use the 'Setup Data Storage' function to clear an area of memory for data storage, but be aware that the old data storage area (if there is one) will be wiped out and no warning will be given.

Once an area is set up for data storage, the 'Add Data', 'Subtract Data', 'Add Marked Data', 'Subtract Marked Data', and 'Scale Data' functions all become active, and icons will appear in the icon menu for the 'Add Data' and 'Subtract Data' functions. In addition, the mouse function automatically changes to 'Add Data'.

If you try to make a data storage area that is larger than the amount of available memory, the 'Setup Data Storage' function will give an error message and no data storage area is set up. In this case, you might try closing some traces or graphs, or lowering the amount of memory allocated to each trace (see Setting Memory Usage, p. 117).

Note for those of you who are technically minded. To determine the amount of memory which is needed for a given amount of time, see *Size of Digitized Data*, p. 34, but multiply the result by two, since 4 bytes are set aside for each value in the storage area.

Sampling Rate of a Data Storage Area

The data storage area has a sampling rate associated with it, and will only accept data from traces which were collected at this sampling rate. The sampling rate is copied from the currently active trace at the time that the data storage area was made (see *Setting up a Data Storage Area* above). If you want to change the sampling rate, select a trace window whose trace was collected at the sampling rate you want, making it the active trace, and then select the 'Setup Data Storage' function from the 'Data' menu, as described above. This will clear the data storage area, and set its sampling rate to that of the active trace.

Spike Studio is currently not smart enough to handle adding data together which was collected at different sampling rates. Normally this won't be a problem, since you will normally use the data storage area for performing operations on parts of the data from within a single trace, or to compare between two or more traces which were collected at the same time, and therefore at the same sampling rate. If you do try to use data for a data storage operation from a trace with a different sampling rate, you will get an error message.

Display of Stored Data

When you make a data storage area, a window is automatically brought up to show the data which is stored in the storage area. This window is called the ‘Data Storage Window’, and it displays the stored data as a trace, just like all the other traces in Spike Studio are displayed. However, you have much less control over this trace than you do over other Spike Studio trace. The gain, offset, and time scale are all set automatically to make sure that all the data will fit into the window provided. The only way you can change the display of the stored data trace is to resize the Data Storage Window, using the resizing box in the bottom righthand corner. Alternatively, if you want to have the more control over the display of this trace, save it out (see Saving the Stored Data below) and then load it back in with the ‘Open’ command (p. 48). You can now treat that data just as you would treat any other data trace.

Choosing the Input Trace for the Storage Area

When using three of the functions which put data into the data storage area, ‘Average Data’, ‘Add Marked Data’, and, ‘Subtract Marked Data’, you must specify which of the available data traces the data will come from. Each time you make a data storage area using the ‘Setup Data Storage’ function, the input trace will be set to the currently active trace (the front most trace - p. 51). If this is not the trace you want to use for input to the data storage area, or if you want to input to the area from more than one trace, then you can switch input traces using the ‘Input Trace’ function in the ‘Data’ menu. When you select this function, a dialog box will appear with a pop-up menu containing all the trace currently open in the program. Select the one you want to work with, and this will become the new input trace to the data storage area. Note that you will only be able to select traces which were collected at the same sampling rate as the initial input trace (see Sampling Rate of a Data Storage Area above).

The functions ‘Add Data’ and ‘Subtract Data’ always add or subtract data from the active trace (the trace where you are clicking the mouse pointer to add or subtract, see below), regardless of which trace is the data storage input trace.

+Adding Data

To add data to the data storage area, use the 'Add Data' function. When this function is selected, either from the 'Data' menu or from the icon menu, the mouse will become a plus sign. Move the mouse to the position in the data where you would like data added from and push down the mouse button. A line will appear, which should be moved exactly to the beginning of the piece of data you would like added. Then release the mouse button. The piece of data starting from where you clicked and as long as the stored data area will be added to the stored data.

Since this method is fairly inexact, and data addition is something which you usually want to do precisely, there is a trick which you might find useful. Use the 'Time/Voltage' function to find the precise spot where you want data addition to start (see Finding Time and Voltage in the Data). Then go to the 'Time From Start' parameter and set the start of the screen as this time (see Time From Start Parameter). Now when you go to do the addition, just move the pointer all the way over to the left edge of the screen and click.

As an alternative to this function, see Adding and Subtracting Data From Markers below.

—Subtracting Data

To subtract data from the data storage area, use the 'Subtract Data' function. When this function is selected, either from the 'Data' menu or from the icon menu, the mouse will change to a minus sign. Move the mouse to the position where you would like the data subtraction to start from and push down the mouse button. A line will appear, which should be moved exactly to the beginning of the piece of data you would like subtracted. Then release the mouse button. The piece of data starting from where you clicked and as long as the stored data area will be subtracted from the stored data. See the hint above under Adding Data to make the subtraction position exact, or see Adding and Subtracting Data From Markers below.

Adding and Subtracting Data From Markers

You can add or subtract from a series of positions within your data using markers. This can be useful when averaging data together (although see *Averaging Data With Markers* below). To add or subtract data after markers, first setup a data storage area (see *Setting Up a Data Storage Area* above). The size of this storage area will determine the length of data which is added or subtracted after each marker. Next, make a set of markers (see *Storing Markers*). Also select the trace which you want the data to come from (see *Choosing the Input Trace for the Data Storage Area* above). Now, select either the 'Add Marked Data' or 'Subtract Marked Data' function. The data starting at each of the markers will be added or subtracted to the data storage area, and the length of data added or subtracted after each marker will be equal to the length of the data storage area. If the data storage area already contained data, this data will be added to or subtracted from the previous data.

Scaling Stored Data

You can multiply or divide the stored data by a constant using the 'Scale Data' function in the 'Data' menu. When you select this function, a dialog box will ask whether you want to multiply or divide the data, and by what number. Check either multiply or divide, enter the number, and click OK. The stored data will then be multiplied or divided by this number. Note that 'Scale Data' can only be used to scale stored data, not to scale normal traces.

Other Storage Area Functions

In addition to the data arithmetic functions described above, you can also use the data storage area to do data averaging. Also described in this section are functions to save the data in the data storage area, and how to get rid of the area when you're done with it.

Averaging Data With Markers

Sometimes you may put your preparation under the same set of conditions a number of times, and want to average its responses. Spike Studio allows you to average digitized data from different points in a digitized data file by using markers. You can then analyze the averaged data. You could do this using the Add and Scale Data functions discussed above, but using markers to perform the data averaging is more exact and faster for some types of data. For a discussion of markers, see Markers in the Event Analysis section. For examples of how to use the data averaging, see Application Note 3, in Appendix B.

To average the digitized data after a set of markers, you can use the 'Average Data' function in the 'Data' menu. First setup a data storage area (see Setting Up a Data Storage Area above). The size of this storage area will determine the length of data which is averaged. Next, make a set of markers (see Storing Markers, p. 89). Then, select the 'Average Data' function. The data starting at each of the markers will be averaged together, and the length of data averaged after each marker will be equal to the length of the data storage area. The averaged data will replace whatever was previously in the data storage area.

The marker averaging produces a simple average. The voltage of the first sample after the first marker is added to the voltage of the first sample after each other marker and then this summed voltage is divided by the number of markers. This value is the first voltage of the averaged data, and all the other samples are averaged similarly.

Saving the Stored Data

Once you have added, subtracted, scaled, and averaged to your satisfaction, use the 'Save' function in the 'File' menu to save your creation. First make the data storage window active by clicking in it or selecting it from the 'Windows' menu to bring it in front of all the other windows. Then select 'Save', which will bring up the usual Macintosh save dialog box, asking you for a filename and then saving the stored data into the file which you specify.

Note that when you save the stored data, it is converted from 4 bytes per value back to 2 bytes per value. A value from the digitizer board is 1 1/2 bytes, and 16 times the maximum 1 1/2 byte value can be stored in 2 bytes. More than this will give an undefined number. Thus, if you have added or subtracted more than 16 sections of data, you should probably divide the data by an appropriate amount before you save it. This is not a problem when averaging.

Erasing the Data Storage Area

If you want a new data storage area, use the 'Setup Data Storage' function, which will erase the old area and set up a new area (see Setting up a Data Storage Area). However, if you are through with data storage areas, you can click on the close box of the data storage window to completely remove it. This will free the memory, inactivate all the storage area functions, and remove the 'Add Data' and 'Subtract Data' icons from the icon menu.

Graphs

Graphs

Spike Studio includes several functions for creating graphs from the analyzed data. Currently two types of graphs are available: histograms and raster plots. Graphs can be made either while the data is being collected or while it is being processed. The data in a graph can then be saved into a file, the graph can be printed or a picture of the graph can be saved.

For making PST histograms, you might want to look at Application Note 1 in Appendix B.

Making a New Graph

To set up a new graph, select the ‘New Graph’ function in the ‘Graphs’ menu. This will initialize a new graph, ask you for a name for the graph, and then let you set it up as described in Setting the Characteristics of a Graph (below). You can have up to 10 separate graphs in the program at once. Each will be displayed in its own window, and can be saved or printed independently from the rest.

Making Graphs takes memory, and if you are running low on memory you may find that you cannot make a new graph. In this case, you can increase the memory available for graphs using the ‘Set Memory’ function (see Setting Memory Usage, p. 117) or increase the total memory available to Spike Studio (see Changing Memory Allocation, p. 116).

Note that to make histograms, you must have some stored events which indicate how many bins to use and where the bins are. If there are no stored events then you will not be allowed to make a new histogram graph (see both Storing Events, p. 88, and Appendix B, p. 127).

Setting the Characteristics of a Graph

Whenever you make a new graph, a dialog box will appear in which you can set up the graph. Subsequently, you can always re-setup a graph by using the 'Setup Graph' function in the 'Graphs' menu. If there is more than one graph when you select the 'Setup Graph' function than a dialog box will ask which graph you would like to setup.

There are quite a few things you can choose when setting up graphs. Some of these are common to all types of graphs and some are specific to each particular type of graph. The general things are all contained in the dialog box which appears after you select 'Setup Graph' or 'New Graph'. In this dialog box is a button called 'Settings' which brings up a dialog box specific to the type of graph (see the Setting Up section, below, for each graph type).

The general settings are as follows:

Graph Type: You must select which type of graph you would like. Currently you can choose between 'Histograms' and 'Raster Plots'.

Algorithm: To the right of the graph type you should pick what type of algorithm you would like to use in calculating information for the graph. There are two options, ‘Normal’ and ‘Threshold’. ‘Normal’ is the regular event calculation algorithm (described in the Event Analysis section). With the ‘Normal’ algorithm, the events calculated will each be considered a single unit for calculation purposes. For instance, a frequency histogram would be a histogram of the frequency of events within the time covered by that bin, not a histogram of spikes within each calculated event, and a raster plot would show each event, not each spike within each event.

‘Threshold’ is a much simpler algorithm which uses a single threshold to pick out events. You set the threshold to use in the text input box just to the right of the ‘Threshold’ radio button. With ‘Threshold’, anytime the data goes above and then below the threshold which you set, this stretch is counted as an event. All the other parameters are ignored with this algorithm, and data is not inverted, rectified, or offset. Thus a frequency histogram with the ‘Threshold’ algorithm would be a histogram of the spiking frequency within each bin and a raster plot would show each spike within that stretch of data.

The advantage of the ‘Threshold’ algorithm is that it is significantly faster than the ‘Normal’ algorithm, so if you are trying to make graphs during data collection it may be advantageous to use ‘Threshold’. It is also simpler to use ‘Threshold’ if you are only interested in spikes.

Note that if you want to use the ‘Lower Threshold’ or ‘Upper Threshold’ parameters to set the threshold in a graph, you must add the offset being used (see Setting the Offset) to the threshold parameter in order to get the value for the graph threshold.

Channels: The ‘Channels’ button allows you to specify which channels of data will contribute to the graph. When you click on this button, a dialog box appears with all the possible channels displayed. Data from any channel which you check will contribute to the graph, and data from any channel without a check will not contribute to the graph.

On-The-Fly Graphing: At the bottom of the dialog box is an item which allows you to make graphs as data is being collected. If 'Update graph upon collection' is checked, then the graph will be updated each time data is collected (see Automatic Graph Updating During Data Collection). Graphs which are being updated upon data collection will not be updated when you select the Add to Graph function (see Adding Data to the Graphs below).

Clearing: Finally, there is a 'Clear' button which has the same function as the 'Clear Data' menu item (see Clearing a Graph). Note that even if you don't press the clear button, changing some functions will automatically clear the graph. Changing the type of graph, or changing the channels which go into the graph will automatically clear the graph (even if you change them back to their original settings).

Setting Up a Histogram

Spike Studio can make histograms of average number of events, average event frequencies, average instantaneous event frequencies, average peak-to-peak heights of each bin, and average integrals of each bin. Each histogram can have several channels of data displayed side-by-side, distinguished by color, and all histogram types can be included in one graph.

To set up a histogram, first make a new graph, set up the general characteristics of the graph as described above, making the graph type 'Histograms'. Then click on the 'Settings' button, from the 'Setup Graph' dialog box. A new dialog box will appear, in which you can set things specific to histograms.

On the left you can select the types of analyses you would like to create histograms of in this graph. Histograms can be made of any analysis included in Spike Studio (see Calculating Information About Events). Check the analyses you would like included in this graph and uncheck the ones which you would not like included.

To the right of each type of analysis, you can set the scale which will be used in the histogram of that analysis. You can enter values for the bottom of the histogram in the 'Minimum' box for that analysis, and for the top of the histogram in the 'Maximum' box. If 'Variable' is not checked for that analysis than the 'Minimum' and 'Maximum' will determine that scale. If 'Variable' is checked, the initial scale will be from 'Minimum' to 'Maximum', and Spike Studio adjust the scale as data is added to the graph so that if a value is seen which is outside the range set by 'Minimum' and 'Maximum', the 'Minimum' or 'Maximum' is set to that value. This usually provides a scale which reasonably fits the data.

In the window for the graph, a separate histogram will be made for each type of data you select. These histograms will appear one on top of the other in the window. The minimum and maximum values for each histogram will be shown to its left. If there is more than one channel in the graph, each channel will receive its own color and the channels will be drawn from lowest to highest channel number in each bin.

The only part of the histogram which is not set from within the ‘Settings’ dialog are the bins. The bins are set using the ‘Store Events’ (see Storing Events). Each ‘Stored Event’ becomes one bin. The histogram remembers the number of bins it has, but not the size or location of its bins. Instead, the size and location of the bins in all histograms is determined by the current set of stored events. The current set of stored events is the set of events which were stored the last time that the ‘Store Events’ function was used. This means that each time ‘Store Events’ is used, the sizes and/or locations of the bins in all histograms are changed to the new set of stored events. This can be used to advantage sometimes (see Application Notes 1, Example 3), but also requires some caution. If you ever change the number of stored events, as opposed to just changing their sizes and locations, you should do a ‘Clear Graph’ on all histogram graphs to reset the number of bins within the graphs. Otherwise, if there are more stored events than there are bins in the histogram, the extra stored events will be ignored, and if there are more bins than stored events, the extra bins will not be updated and the data in them will no longer be comparable to the data in the other bins. Note also that if there are stored events defined past the end of a stretch of data, those bins will be ignored when adding that data to the histogram.

Finally, if you are making histograms of more than one type of analysis, you have a choice of putting all the histograms in one graph (thus having them all stacked one on top of the other in the same window, which may look a bit squashed) or making a separate graph for each analysis. This is basically a stylistic decision when you are making histograms from already digitized data, but if you are making the histograms on-the-fly then it will be significantly faster to put all the analyses into one graph, so that the data only needs to be processed once.

Setting Up a Raster Plot

Spike Studio can make raster plots which show events from up to 100 data traces, and can show event onsets, event offsets, or both for each trace. You can select from 8 different symbol types when plotting the events, and the size of each symbol is adjustable. In addition, you can change the spacing between rows of data.

Up to 8 different channels can be included in each raster plot. To select which channels are included, see Setting the Characteristics of a Graph above. There may be multiple traces included in the graph for each channel. For instance, if you were stimulating a prep 25 times and wanted to see the spikes on channels 1 and 3 after each stimulation, you would include channels 1 and 3 in the graph, and at the end of your data collection the graph would show 25 traces from channel 1 and 25 traces from channel 3.

To set up a raster plot, first make a new graph, setting up the general characteristics of the graph as described above, and selecting the graph type 'Raster Plot'. Then, from the 'Setup Graph' dialog box, click on the 'Settings' button. A new dialog box will appear, in which you can set up things specific to raster plots.

Most of the dialog box is devoted to the appearance of the data. For each channel which is included in the graph, there will be a line of items which you can use to change the appearance of its traces. There are six columns of items, with their titles at the top. If the 'Onset' box is checked for a channel, than traces from that channel will have their onsets plotted using the symbol given in the first 'Symbol' column, with a radius given in the first 'Size' column. Similarly, if the 'Offset' box is checked, then traces from that channel will have their offsets plotted using the symbol given in the second 'Symbol' column, with a radius given in the second 'Size' column. To change a symbol, click on the symbol you want to change and it will switch to the next possible symbol type.

The 'Row Spacing' item at the top right lets you set how far apart the traces will be drawn. The traces will be separated by the number of pixels given in 'Row Spacing'.

The time scale of the raster plot will go from 0 to 'Max Time'. If the 'Max Time Variable' box is checked, then 'Max Time' will be updated each time data is added to the graph so that all the data is within 'Max Time'.

'Max Num Events' is the maximum number of events which can be stored for each trace. You should set this number to a little more than the maximum possible number of events that you might get on any particular trace, but not too much more since the higher this number is, the more memory it will take to make the graph. If there are more than 'Max Num Events' events from one of the data traces, the extra events will not appear in the graph and no error message will be given.

You can have up to 100 traces in a graph, but memory is not allocated for all 100 traces when the graph is created. Instead, enough memory for one trace is allocated each time a new trace is added. This is done to use the minimum amount of memory necessary, but it can have disadvantages. For one thing, you may run out of memory before all the traces have been added to your graph. If you are adding traces from already collected data, then you can usually use the ‘Set Memory’ function in the ‘Extras’ menu to free more memory. If you are adding on-the-fly, however, then once you run out of memory subsequently collected traces will not be added to the raster plot. Furthermore, allocating memory takes time, and you do not want to waste this time if you are graphing on-the-fly. For this reason, you can set aside memory for a certain number of traces when you make the graph. You do this by setting ‘Initial Num Traces’ to the number of traces you would like to pre-allocate memory for. If you are making a raster plot on-the-fly, you should probably set ‘Initial Num Traces’ to the total number of traces you expect to collect, or a little more.

Changing a Graph Without Erasing Data

You can change the settings of a graph without clearing the data by selecting ‘Setup Graph’ in the ‘Graphs’ menu and clicking on the ‘Settings’ dialog box. If there is more than one graph open, you will be asked to select which graph you want change. The main reason to do this is to change the scales of the graph or the way a graph is displayed.

With histograms, you can safely change the scales without any data being erased. If you change the types of histograms that are being displayed, the change will occur without erasing anything, but depending on what was being the previous display and what you ask it to display, some of the analyses might contain no data. Thus if you change the analyses to display it is safest to clear the graph and start fresh.

With raster plots, you can safely change the time scale, the spacing between rows of data, the symbols and symbol sizes, and whether onsets or offsets are displayed for each channel, without any loss of data. However, changing the maximum number of events or the initial number of traces will erase all the previous data.

Adding Data to the Graphs

There are two ways to add data to a graph. One way is to automatically add data during collection, which is discussed in the next section. The other way is to add data from one of the open traces, discussed here.

To add data to the graphs from one of the traces, first make that trace the active trace by either clicking in its window, or by selecting it from the 'Windows' menu. Then select the 'Add To Graphs' item in the 'Graph' menu. The data in the active trace will be sent to each graph, to be processed and added to the data in that graph (as long as the graph is using data from the channel number of that trace - see Setting the Characteristics of a Graph, above). Graphs which are getting updated upon data collection will not be updated with the 'Add To Graphs' command. Files whose names do not specify a channel number will be considered as channel 1 (see Final Data, p. 38, and Channel Number Parameter, p. 52).

If a graph is using the Normal algorithm for calculating events from the data (not the Threshold algorithm), it does not recalculate events when the 'Add To Graphs' command is given. Instead, the graph will just use the events which are already calculated. It uses all calculated events for that whole trace of data, including events which occur earlier or later than the data currently showing on the screen.

If a graph is using the Threshold algorithm, the trace is used starting at the left of the screen. Thus with histograms, for instance, if you define a series of bins which cover a screenful of data, you could ‘Add To Graphs’ from a screen which starts with the beginning of a certain trial in your experiment, then move the trace over so that the screen starts with the beginning of the next trial and ‘Add To Graphs’ again, and so on, building up a histogram. You could also load in a series of different traces and add each one to the graph to make a histogram of several trials. Similarly, with raster plots you could get all the spikes on one screenful of data as one trace, then all the spikes on another screenful of data as a second trace.

Note: For histogram bins, if you make your bins from the events on the screen (‘Store Events’ with ‘From Screen’ checked) you should make your events at the beginning of a trace (with ‘Time From Start’ equal to 0), unless you specifically want the bins starting some ways past the beginning of the screen of data.

Another Note: Notice that the data used to update graphs is not from the same position when using Normal or when using Threshold algorithms. The Normal algorithm uses all events, *from the beginning of the data to the end*. The Threshold algorithm uses *only the data on the screen*.

Automatic Graph Updating During Data Collection

One of the most powerful features of Spike Studio's graphs is that they can be updated as data is being collected, giving you with real-time information about the data. This function may be particularly useful when used in conjunction with the 'Multiple Triggering' function.

To have graphs automatically updated after each data collection, check the 'Update graph upon collection' box when setting the graph characteristics (see Setting the Characteristics of a Graph above). From then on, whenever you collect data, each graph will be updated either as the data is being collected (with continuous data collection) or upon completion of a section of digitizing (with multiple triggering or on boards without continuous data collection capability). If you are doing 'Multiple Triggering', an update will occur after each trigger. Each update is exactly like doing an 'Add To Graphs' (see Adding Data to the Graphs), with two exceptions. First, only graphs which are set to 'Update graph upon collection' will be sent the data, and second, any graphs using the 'Normal' algorithm for calculating events will actually calculate events for the data.

Saving the Data in a Graph

The data which makes up a graph can be saved into a text file, which can then be read using a word-processor or could be imported into a real graphing program. To save the graph data, click in the window of the graph you want to save to make it active (or select it from the Windows menu), and then select the 'Save' item in the 'File' menu. A standard Macintosh save dialog will appear, where you can enter a filename for the data, and the data from the graph will be saved to this file.

Histograms are saved in a format which looks like this:

```
Num_Spikes  Avg_Inst_Freq  Avg_Height
Channel 1      Number of Trials = 10
10   23.22   2.34
17   42.91   2.17
6    5.89    2.22
etc.
Channel 2      Number of Trials = 8
10   23.22   2.34
17   42.91   2.17
6    5.89    2.22
etc.
```

The top line gives headings of the columns of data. The next line gives the channel number and the number of trials incorporated into the graph for this channel. Following that is the data from all the bins for that channel, with one row for each bin. The types of histogram data saved are the same as those displayed in the graph. Then this is repeated for all the other channels in the graph.

Raster plots are saved in a format which looks like this:

```
1_On  1_Off  2_On  3_On
1.533  1.726  1.644  1.978
2.645  2.779  2.737  3.008
4.422  4.625  4.611  4.918
                    5.227  6.893
                    6.253
```

Each column gives the data from one row of the raster plot, starting from the top row. The top line indicates the channel number for each column of numbers, and whether that column is onset or offset data. For rows of the raster plot which show both onsets and offsets, a column will be saved with the onsets, followed by a column with the offsets.

Saving Data in a Graph to the Clipboard

Instead of saving data from a graph to a file, you can save it to the clipboard so it can be 'Pasted' directly into another application. To do this, first make the graph whose data you want the front most window, by either clicking in it or by selecting it from the 'Windows' menu. Then select the 'Copy Data' command in the 'Edit' menu. The data from the graph will now be placed in the clipboard in exactly the same format as is shown above under Saving the Data in a Graph.

Note that if you happen to have a file in the folder you're using named 'FooBazooKalamazoo' it will be erased when you use the 'Copy Data' function.

Clearing a Graph

To clear the data in a graph, make the graph you want to clear the front most window, by either clicking in it or by selecting it from the 'Windows' menu. Then select the 'Clear Graph' item in the 'Graphs' menu.

If the graph you select is a histogram, 'Clear Graph' will set all its data to 0, set the number of trials to 0, and erase the graph. It is a good idea to clear a histogram whenever you change what analysis types are being displayed. If its a raster plot, you will be given the choice of clearing the whole graph, or just clearing one trace. If you want to clear just one trace, type in the number of the trace in the 'Trace Number' box (the trace at the top of the graph is trace 1) and then click on the 'Trace' button. That trace will be removed and all the traces below it will be moved up on line. If you want to clear the whole graph, click on the 'Graph' button. In addition to clearing all the data, this will also deallocate all memory for storing traces and then reallocate enough to store 'Initial Num Traces'.

You can also clear a graph from within the 'Setup Graph' dialog box using the 'Clear' button (although for raster plots this will not give you the option of clearing just one trace). Selecting 'Setup Graph' does not itself clear the graph.

Changing the Size of a Graph Window

Sometimes you may want to change the size of a graph window, squeezing it into a corner of the screen, or expanding it so you can see the data better. You may also want to adjust the size of a graph prior to printing. Just click the mouse in the lower right corner of the graph window, and adjust its size by moving the mouse while holding down the mouse button.

Saving, Copying, and Printing a Graph

You can save a graph as a picture, copy it to the clipboard, or print it, using the 'Save Picture', 'Copy', and 'Print' commands, respectively. See the Outputting Pictures section (p. 62), or simply make the window of the graph you want active (click in it or select it from the 'Windows' menu), select one of the above commands, and follow the standard procedures for saving or printing. The size of the graph will be exactly its size on the screen (see Changing the Graph Window Size above). You can also save, print, and copy multiple graphs on a single page using the Page Layout function (p. 65).

Removing a Graph

When you are done with a graph, you can completely get rid of the graph and its window, and free the memory it was using, by clicking in the close box in the upper left corner of the graph window.

Etc.

Other Functions

This section contains a few miscellaneous functions, tidbits of information, and other stuff which I thought might be useful, but which didn't really fit in anywhere else. Most of the functions in this part of the manual are in the 'Extras' menu.

Changing Memory Allocation

If you are running Spike Studio under multifinder or System 7 or above, the program is only allocated a certain amount of the total memory available in your computer. Initially the program is set up to request 2 1/2 megabytes of memory, which is pretty reasonable on a 4 meg machine. If you want to change this, click once on the Spike Studio icon (while the program is not running), and 'Get Info' on the program (in the system's 'File' menu). At the bottom of the information is a box entitled Memory Requirements which contains three items: Suggested Size, Minimum Size, and Preferred Size, the last two of which you can change. These numbers indicate how much memory will be allocated to the program when it is run (in Kbytes). The one which is most useful is Preferred Size, which is how much memory will be allocated if there is plenty of memory available. You can adjust this number to suit your needs.

In general, it is best if you allocate as much memory as possible to Spike Studio. The more memory Spike Studio is allocated, the smoother and faster it will run, especially if you are dealing with large amounts of data. For a GW system, the memory allocation also determines the maximum amount of data which can be digitized at once (unless you have an FIFO board). Thus if Spike Studio will generally be the only program running when you are using it, you might want to increase its memory allocation to 3 megabytes (or more if you have more than a 4 meg machine). On the other hand, if you want to multitask Spike Studio with another program (perhaps a statistics or graphing program), you might need to reduce the amount of memory Spike Studio is allocated. When you try to reduce it below 2 1/2 meg, the Macintosh will give you a nasty message, but you can ignore it. Setting the memory allocated to less than 1 meg is a little bit risky, since this is starting to tax the program (it should work fine, and I've tested it with as low as 500 K, but its clunky and any latent bugs are a little more likely appear).

To see how much memory the program actually has available, use the 'Set Memory' function described below.

Another way to increase memory allocated to the program is to run under the finder instead of under multifinder. This will allow more memory for Spike Studio, however you won't be able to multitask (you can't do this under System 7).

Setting Memory Usage

Spike Studio allocates a certain amount of memory to each open trace, and this affects many things. You can only see the part of a trace which is in memory at that time, so if you have a long trace and the memory allocated per trace is not large enough, you may only be able to see a piece of it at a time. The number of events which can be found for each trace is also determined by how much memory is allocated to storing events, so if a trace has too many events there may not be enough memory to store them all. On the other hand, if too much memory is allocated to storing traces, there may not be enough left to collect data, create graphs, or make a large data storage area. Allocating too much memory to each trace may also slow down the speed of opening and moving around large traces. For these reasons, there is a function which lets you optimize how much memory will be allocated for each trace and its associated events.

To change the memory allocation, select the 'Set Memory' function in the 'Extras' menu. A dialog box will appear which both shows the total amount of memory available, and allows you to set the amount used for each trace. There are actually three chunks of memory set aside for each trace. The first is a chunk of memory used to store the digitized data of the trace. The second is memory used to store the calculated events for that trace (p. 71). The third chunk of memory is used to store user added and deleted events for the trace (p. 76). The dialog box lets you input each of these numbers, the first in number of bytes per trace and the last two in number of events per trace. For instance, if you wanted 250 Kbytes of memory allocated to each data trace which was opened, and you wanted to be able to have up to 2000 calculated events and 1000 user events per trace, then you would put 250000, 2000, and 1000 into each of the dialog boxes, respectively.

When you change the amount of memory allocated to each trace, all the traces are reloaded so that the change takes effect immediately. However, as long as you don't change the maximum calculated events or maximum user events per trace, any events which previously existed will be retained. If you do change either the maximum calculated or user events per trace, then all previous events will be erased. So don't do this if you have events you want to save (or save them first, change the memory allocation, and then reload them). No matter what you change in this dialog box, there will be no effect on currently open graphs or an open data storage area.

At the bottom of this dialog box is shown both the amount of memory in the program which is currently free to be used (is not already in use for storing traces, graphs, etc.), and the total amount of memory available to the program. This total does not include memory allocated for graphs, and there is also an extra 5 K bytes available that is kept in reserve.

Filtering Event Files

The event files are normally saved as onset/offset files (see Saving). You may want a file that is just onsets or just offsets, or that is the midpoint between the two. To get such a file from the onset/offset file, you can use the function 'Filter File' in the 'Extras' menu. When you select this function a dialog box will ask if you want onsets, offsets, or midpoints. Check the box next to the one you want. Then a standard Macintosh file requesting dialog will come up, asking for the event file, and when you select an event file, a save dialog box will ask for a filename for the filtered data. Then the event file will be filtered, and the appropriate information will be saved in the new file.

Cutting Data

Sometimes you may not start or end your data taking exactly where you meant to, leaving some extra data at the beginning or the end. You can remove this extra using the 'Cut Beginning' and 'Cut End' functions in the 'Extras' menu.

The 'Cut End' function allows you to cut off a certain amount at the end of the digitized data in the active trace. When you select 'Cut End', the end of the active trace will be shown. A vertical line will be drawn through the data which represents the position where the data will be cut. A dialog box will also appear on the screen with this position displayed in seconds. To change where the data will be cut, type in a new position in the dialog box, and then click on the 'Check Position' button to redisplay the trace with the new cut position shown. When the line is at the place where you want to cut the data, click on the 'Cut' button. All the data to the right of the line will be thrown out.

If you want to cut all the other traces which were collected at the same time as this one (for instance, if you collected four channels of data and all must be cut at the same place), then check the 'cut all associated files' box. Now when you click on 'Cut', the active trace and the traces from all other channels which were collected at the same time will be cut at the indicated spot (see Final Data, p. 38). There are some cautions associated with cutting all associated traces, however. If you renamed the trace files to something other than <filename><number> (see Final Data, p. 38), then you may get error messages saying that some of the files can't be found, or worse, you may end up cutting files which you didn't intend to have cut. Also, if you collected the files using multiple triggering with separate files for each trigger (p. 27), then you will need to cut the files one at a time, or else you may end up cutting files you don't want to cut and missing files which you did want cut.

'Cut Beginning' works the same way as 'Cut End', except it starts from the beginning of the files and cuts off the data to the left of the line.

Artificial Events

There may be times when you want to find out information about the first bit of time after an event, then the next bit, then the next bit, etc.. You could use this, for example, to create histograms of the number of spikes in each successive hundred millisecond interval after a certain event. You might also want to find out information about a period of time after the onset of each event, or after each marker point. Spike Studio includes a feature that will let you create regularly spaced events of a regular length, called artificial events.

Select 'Artificial Events' from the 'Extras' menu. A dialog box will ask you to make three choices. The first choice is whether the events should be created just for the data shown on the screen, or for the whole data trace. This is selected by setting one of the radio buttons on the left. The second choice is whether the events will be created continuously or after markers. The buttons on the right are used to make this selection. If 'After Markers' is set, an event will be made after each marker. You can make an artificial event after each event onset by storing markers from the events (see Storing Markers). If 'Continuous' is checked, events will be made starting from the first event on the screen or in the file (either calculated or added), or at the beginning of the screen or trace if there are no events. Thus if you want the artificial events to start at a specific point, add an event in (using 'Add Event') with the onset at that point. If 'Continuous' is checked then the events will be made one right after the other to the end of the screen or trace. Finally, the length of each event to be created is set in the Length of Events field.

The artificial events are just like user-added events. You can use any of the analysis or saving functions on these events, and you can get rid of them by selecting the 'Delete User Events' function. All previous events, both calculated and user added, are deleted when artificial events are made, although deletions are not forgotten (so that if you go back and calculate events again, the same calculated events will remain deleted).

One of this function's primary uses is making post-stimulus-time histograms. Application Note 1 which explains how to do this.

Saving the Configuration

You will probably want many of the program settings to be pretty similar from one use to the next. For instance, your board is always likely to be in the same slot in the computer, and you may always collect data at 5kHz on two channels, want to look at 15 seconds of the data at a time and want the data to be rectified. It would be inconvenient to have to reset all these each time you use the program, so Spike Studio allows you to save all of these settings once and have them loaded each time the program is run. This is called the configuration.

To save the configuration, select the 'Save Configuration' function in the 'Extras' menu. This will save the configuration, and in the future whenever you start up the program, things will be set the same way as they were when you selected this function. The configuration is saved in the program itself which is why it's not a configuration "file," and it will follow the program around if you copy the program from one computer to another . (The configuration is in the Resource File, and those of you who know about Resource Files can also change the configuration using ResEdit on the 'init' rsrc - but why bother?).

The list below contains all the items that are currently saved in the configuration. If someone wants more items added, it can be done fairly easily:

Board Type	Board Voltage Range	
Board Slot		
Board Gain	Sampling Rate	Channels to
Collect		
Triggering Parameters	Multiple Triggering Parameters	

Seconds per Window Overlap
Minimum Event Duration Minimum Interevent
Duration
Rectify Inverted Automatic Offset
Use Middle Spike Automatic Redraw
Save Options
Memory Per Trace Max Calculated Events Max User
Events

In addition, the position of the Parameters, Info, Icon Menu, and traces windows are all saved in the Configuration. Thus, if you always want the Parameters, Info, and Icon Menu windows to appear in a certain place, put them there and then select the ‘Save Configuration’ command. Similarly, if you always want trace windows to be certain sizes and appear in certain places, then open as many traces as you want to set window positions for, set them up in exactly the way you want them to appear (with the first trace you opened put in the first position, the second trace you opened in the second position, and so on), and then save the configuration. From now on, whenever you start the program and open the first trace, it will appear in the position of the first window, and the second trace you open will appear in the second position, etc.. (see Positioning of Windows on the Screen, p. 54). This is a little confusing to explain, so the best thing is just to try it and you’ll see how it works.

Saving Digitized Data

You may occasionally want to digitize some data with Spike Studio and then import it into another program to analyze it. You may also want Spike Studio to change the data in some way, such as rectifying it or correcting an offset, before you use the data outside of Spike Studio. The 'Save Digitized Data' function allows you to export digitized data from Spike Studio as either a text file or a binary file, from either one of the currently open traces or from the data storage area, and either unmodified or modified as the trace on the display is modified. To save from the data storage area, you can also use the 'Save' function in the 'File' menu (see Saving the Stored Data, p. 98), and as long as you want to save it as a normal Spike Studio trace file then that is a better option.

To export the digitized data, first select the window of the trace you want to export to make it active. You can export any open trace, or the data in the data storage area. If you want to export only a piece of the trace, then make an event or a set of events which delineate the piece you want to export.

Now select 'Save Digitized Data' from the 'File' menu. A dialog box will appear with three sets of choices. First, if you are exporting from a trace, you must select whether to export the whole trace or only those parts delineated by events. If you want to export the whole trace, from beginning to end, then check the 'Whole Trace' radio button. Otherwise, check the 'Events Only' button, which will save out only the parts of the trace which lie within the currently defined events for that trace. If there is more than one event, then the data within each event will be saved consecutively in the file after the data from the event before it, with no breaks between events. This choice will be ignored if you are exporting from the data storage area.

The next choice is the format to export the data in. Binary is the format that Spike Studio normally uses to store digitized data (see Final Data). If you import this into another program you will have to convert these numbers into voltages with a conversion factor based your board's set up and gain. Ascii will export the data as a text file, with all the voltages in a column and each voltage written out with 3, 4, or 5 digits of accuracy, depending on what the board gain is.

You can save the data either as it appears on the screen, or as it was collected, unmodified by any of the commands which affect the drawing of the trace. If the 'modified' radio button is checked, then the values in the digitized data will be inverted, rectified and offset exactly as they are on the screen before they are saved. Data from the storage area will be modified in the same way that the data in the bottom trace is modified. If 'unmodified' is checked then the values will be saved just as they came in, with none of the above modifications.

When you have set up the way you want the data to be exported, click OK, and a normal Macintosh file saving dialog will come up, where you can specify the file name. The data will then be saved, as you specified, to this file.

Saving digitized data may slow if the traces are long, since each value is saved individually.

Total Seconds in File

The Total Seconds in File is a piece of information which is displayed in the Parameters window, to the right of the parameters. It is the number of seconds of data contained in the file of data which you are currently looking at. This is not user adjustable in the same way that the numbers to the left of it is adjustable. To change the number of seconds of data in the file, see Cutting Data.

Sampling Rate Info Item

One of the pieces of information displayed in the Parameters window is the sampling rate of the currently active trace. The Sampling Rate is shown to the right of the parameters, and is given in kHz. This is the sampling rate that the currently active trace was collected at.

Clear

The 'Clear' function in the 'Edit' menu will clear any message in the Info window.

Inactive Menu Items

The following menu items do nothing. They are included for compatibility with the Macintosh interface.

In the 'File' menu:

New

The Help command just gives useless messages. If someone found a real need for a help function, I might be persuaded to add a useful one in; however I'd rather have feedback on the manual and make it more helpful, so that the program won't need a help function.

In the Edit menu:

Undo

Paste

Quitting the Program

To quit the program, select the 'Quit' command in the 'File' menu.

About Spike Studio

The 'About Spike Studio' item in the apple menu includes the version number of your copy of Spike Studio, my address in case you lose it, and acknowledgments. The version number works like this. The first two numbers are the real version, for instance v3.2. The third number is kind of a sub-version, perhaps with a few bug fixes or a few not fully tested new features if it's not 0. The final three numbers are your personal identification code (so don't give the program to your friends cause I'll know where it came from).

Acknowledgments

Just in case you never look at the About menu item, I'll acknowledge people again here. Thanks to Avis Cohen, who started me on this project, Julian Vrieslander who provided the core on which the program is built, Nicholas Mellon and Sylvie Ryckebusch who provided code, lots of ideas, and lots of bugs, and Margaret Baker who helped with the manual and also made lots of suggestions in the early stages.

Appendix A

Application Notes

.c2.'Appendix A: Application Notes';Appendix A: Application Notes

This appendix contains step-by-step instructions for how to perform several operations with Spike Studio. I tried to include operations which may be confusing in the manual, or which may require a combination of several functions. I intend these more as recipes than as detailed explanations, since I figure that you can use the rest of the manual as a reference for anything you don't understand. This section will expand as more customers request help doing novel things with Spike Studio.

Contents

Application Note 1: Post Stimulus Time Histograms 127

Example 1 deals with one channel of already recorded data.

Example 2 deals with on-the-fly PST histograms.

Example 3 deals with several channels of already recorded data, and a separate stimulus channel.

Application Note 2: **GW: and NI: Triggered Data Acquisition 132**

Example 1 deals with single trigger acquisition with trigger signal on one of the channels.

Example 2 deals with single trigger acquisition with trigger signal on the external triggering input.

Example 3 deals with multiple trigger acquisition when saving one file per channel.

Example 4 deals with multiple trigger acquisition when

saving multiple files per channel.

Application Note 3: Data Averaging

137

Example 1 deals with averaging data based on events within the data

Example 2 deals with averaging data based on events in another channel of data.

Example 3 deals with averaging data from several data files.

Application Note 1: Post Stimulus Time Histograms

Post stimulus time (PST) histograms are a very common way of analyzing data. They are useful for looking at the response over time of some element to a stimulus. The time after the stimulus is broken into sections, and a certain quantity is measured in each section. This can then be plotted to show response over time.

Making PST histograms in Spike Studio is a two-step process. The first step is setting up the time intervals which will be used for the histogram, and the second step is saving and/or displaying the data within these intervals. In Spike Studio, you can save the data for the histogram to disk, where it can be loaded into a graphing or statistics program, or you can use Spike Studio's graphing functions to display the histogram from within the program.

You may want to read the sections of the manual called Artificial Events (p. 119) and Storing Events (p. 88) before continuing here. Note also that for instantaneous frequencies, the 'Save Freq - Event' function might be even better than a PST, since it gives continuous data (see Saving Instantaneous Frequencies for One Event, p. 86).

Example 1

Suppose you stimulate a neuron to fire in such a way that it fires repetitively, and would like to know the number of spikes fired every 100 msec until it becomes quiet again. You have digitized the data so that you the stimulation starts 3 seconds into the file, and upon examination you see that the last spike occurs 7 seconds into the file. Thus you would like the number of spikes in every 100 msec section of data from 3 to 7 seconds.

The first step is to set up the bins. To do this, select the 'Store Events' function in the 'Extras' menu. A dialog box will come up in which you can define where the bins should be. Put a 3 in the 'Starting' field and a 7 in the 'Ending' field, to indicate that the bins should start at 3 seconds and end at 7 seconds. Put 0.1 in the 'Length' field so each bin will be 0.1 seconds long. Then uncheck the 'From Screen' check box, so that the bins will be set up from the numbers you just inputted instead of from the events on the screen. Finally, click on OK. Note that a little bit of rounding may occur in the numbers you entered. If you get one less bin than you expect, try setting the 'Ending' a little bit later.

The second step is to make a histogram. Select 'New Graph' in the 'Graphs' menu. A dialog box will appear where you need to specify the type of graph you want to make. In the dialog box, check the 'Histogram' box under Type and the 'Threshold' box under Algorithm (see Setting up Graphs for explanations of all these items). Make sure the 'Update upon data collection' box is not checked. Click on the 'Channels' button and check the channel number of the data channel you're making the histogram from, shown in the Channel Number parameter (if the name of the file does not indicate a channel, check channel 1). Now click on the 'Histogram Settings' button. This will bring up another dialog box where you choose the type of histogram you want and set the histogram scales. Check 'Number of Spikes' and uncheck all the others. Also check the Variable Scales box next to 'Number of Spikes', so the scale will automatically be adjusted. Then click OK in both dialog boxes.

You now have the histogram set up, and a window will appear for the histogram. To add the data to the histogram, click in the trace window to make it the active trace (the front most trace), and then select the 'Add To Graphs' command from the 'Graphs' menu. You will now see a histogram of the number of spikes in each 100 milliseconds from 3 to 7 seconds. If you want to save the data, you can use the 'Save Graph' command in the 'Graphs' menu.

If you do not want to actually see the histogram, or you want a more flexible way to make bins, try using the Artificial Events function (p. 119).

Example 2

Spike Studio can make PST histograms as you are collecting data. These are known as on-the-fly PST histograms, and they are a useful way of looking at the data as it's being collected. Suppose that you are making intracellular recordings of spiking patterns on two cells, after stimulations of a third cell. Suppose further that the stimulations are about 2 seconds apart, and the spiking on both the cells dies out after about 500 milliseconds. You would like histograms showing the number of spikes and average instantaneous spiking frequency in each 50 milliseconds after the stimulus, up to 500 milliseconds away, and you would like these histograms to be updated after each stimulation.

First, you will need to set up and store events corresponding to the 50 millisecond bins, as in all the examples above. Using the 'Store Events' function in the 'Extras' menu, set 'Starting' to 0, 'Ending' to 0.5, and 'Length' to 0.05. Unclick 'From Screen' and click on OK.

Now you have to make a graph, as described in Example 1. Choose 'New Graph' from the 'Graphs' menu and check the check boxes for 'Histogram', 'Threshold', and 'Update upon data collection'. This last selection tells Spike Studio to update the graph each time you collect data. Then click on the 'Channels' button, and when the dialog box appears, check the two channels containing the intracellular recordings, then click OK. You have now selected the channels which will make up the graph. Finally, click on the 'Histogram Setup' button. A new dialog box will appear where you should check 'Number of Spikes' and 'Instantaneous Frequency', and their respective boxes in the Variable Scales column.

Now click on OK in both dialog boxes. A new window will appear containing two graphs, one for the number of spikes data and a second for the instantaneous frequencies data. These graphs will be automatically updated with each data collection. For an example of how to do multiply triggered data collection, see below (p. 135).

Example 3

Suppose you have a file with a series of stimuli on one channel and the responses on another, and would like to make PST histograms starting at each stimulus. This is similar to example 2, but instead of making the histograms as you are recording you want to make them after you have recorded.

First 'Open' the stimulus channel. Adjust the 'Time From Start' and 'Seconds per Window' parameters so that both the first stimuli and the stretch of data from which you want to make the PST are showing in the window, while the second stimuli is not showing.

In this case we want the bins to start exactly where the stimulus starts. You can do this two ways. One way is to make bins using the 'Store Events' function as in example 1, starting the bins at 0. Then adjust the 'Time From Start' parameter so that the stimulus is exactly at the beginning of the screen. Also adjust the 'Seconds per Window' parameter so that all the response will be shown on the screen.

An alternate way of making the bins is to use the ‘Artificial Events’ function. Adjust ‘Time From Start’ so that the stimulus is near the beginning of the screen and adjust ‘Seconds per Window’ so that the response goes until close to the end of the screen. Then either add an event with its onset at the beginning of the stimuli, or calculate events so that an event is calculated beginning at the stimuli. Make sure there are no other events on the screen before this stimulus event. Now use the ‘Artificial Events’ function to make an event corresponding to each bin you want (see Artificial Events, p. 119). Store these events with the ‘Store Events’ command in the ‘Analysis’ menu, checking the ‘From Screen’ box.

As in Example 1, set up a histogram graph using the ‘New Graph’ function in the ‘Graph’ menu. Check the check boxes for ‘Histogram’ and ‘Threshold’, and make sure ‘Update upon data collection’ is unchecked. Click on the ‘Channels’ button, and in the new dialog box check the channels which you want included in this graph, then click OK. Then click on the ‘Histogram Setup’ button. Another dialog box will come up where you should choose the type of histogram you want and set the histogram scales. Finally click OK in both dialog boxes. A window will come up with the graphs.

Now ‘Open’ the first of the channels you want included in the graph, and use the ‘Add To Graph’ command in the ‘Graphs’ menu to add the data from that channel to the histogram. Load in the next channel, if there is another one, and again use the ‘Add To Graph’ command to add the data from this channel. Repeat this for each channel which should be included in the graph.

Next, make the stimulus channel the active trace again, by clicking in its window or selecting it from the ‘Windows’ menu. Using the ‘Time From Start’ parameter or the Set Position mouse function, move the trace so that the next stimulus and the stretch of data after it are showing on the screen. If you used the first method of making bins above, make sure the stimulus is at the left edge of the screen. If you used the artificial events method of making bins, then again make the appropriate artificial events, using ‘Store Events’ to store them. In this case, make sure that the events you make here correspond exactly to the events you made for the first stimulus (ie. same number and same size), as the program will be adding the data from these events in the same set of histogram bins as the data from the first events.

Switch back to the first of the response channels again. Use ‘Add To Graph’ to add the data from this channel to the graph. This data will be added to the data already in the bins from the first stimulus. Repeat for each channel, as above, and repeat this whole procedure for each additional stimulus.

Application Note 2: GW: and NI: Triggered Data Acquisition

Spike Studio allows you to make use of the triggering features in GW or NI boards to perform triggered data acquisitions, and has a multiple triggering feature for collecting data after a series of triggers. These are both explained under the appropriate sections in the manual. Below are some examples of how to do triggered acquisitions.

SB: The Macintosh sound input does not, as far as I know, have any triggering functions. I probably can add something in if someone wanted me to, but currently Spike Studio doesn't include any triggering for sound boards. Sorry.

Example 1

Suppose you are recording two channels of data, channel 1 which contains a stimulator pulse and channel 2 containing the response to that stimulus. You want to start your data acquisition at the beginning of the stimulation, which happens to be a square wave that goes from 0 to 5 volts.

To do this, set up the 'Channels to Collect' and 'Sampling Rate' in the 'File' menu as you would normally. Next, select the 'Set Trigger' function from the 'File' menu. In the dialog box which appears, check the 'Trigger from Data' box so that the program will know that it should look for the trigger on one of the channels. If any other trigger locations are checked, uncheck them. Also check the 'Low to High' box so that the program will be looking for a low-to-high transition. In the 'Trigger Channel' box, enter 1 to set the trigger channel as channel 1. Finally, put a voltage level of 2.5 into the 'Trigger Level' box (2.5 is a voltage level that the square wave will cross). (NI: Also set 'Pre Trigger Amount' to 0). Now check the 'Trigger the Data Acquisition' box at the top of the dialog, then click 'OK'.

From here on, data acquisition works as it does normally. Select 'Get Data' from the 'File' menu and enter the appropriate things (this will depend on which type of board you have). When you push return, the computer will freeze as it waits for the trigger. When the stimulator pulse arrives, data collection will begin. (GW: You won't see any difference after the stimulator pulse, since the computer will remain frozen). When the proper amount of data has been collected, the first channel of data will appear on the screen. If you then look at the data from channel 1, you should see that it starts just as the stimulus pulse crosses the 'Trigger Level' which you set above.

One note of caution. If you have set a 'Trigger Level' which is too high, the computer will not be triggered, will not collect data, and will remain frozen. On GW boards, you can interrupt the computer by moving the mouse. After several seconds, the program will unfreeze and issue an error message. On NI boards, if no trigger is seen for 30 seconds then the program will unfreeze and issue an error message.

GW: While this is useful as a way to recover from faulty settings, it can also be dangerous in that you may inadvertently interrupt triggered data collection by accidentally moving the mouse. Be aware of this, especially if an error message appears after a triggered data collection.

Example 2

Suppose that instead of having a stimulator pulse on one of the channels, you want to connect your board directly to the stimulator. If the stimulator puts out a digital pulse when it stimulates, you can hook this to a special input on the digitizing board, which will allow the program to trigger off of this pulse. This is called an external trigger, and you should look in the manuals which came with your digitizing board for more information on how to use it with you board. The advantage of external triggering is that you can save your input channels for real data, instead of using one of them to collect the stimulus

The procedure is similar to the one outlined in Example 1. The only difference is in the 'Set Trigger' dialog box. When you select 'Set Trigger', check the 'External Triggering' box and either the 'Low to High' or 'High to Low' box as appropriate. The 'Trigger Channel' and 'Trigger Level' inputs are now irrelevant. (NI: I believe that the 'Pre Trigger Amount' is also now irrelevant). Finally, check the 'Trigger the Data Acquisition' box, and click 'OK'. Now follow the same procedures as outlined in Example 1. The acquisition will start at either the leading or falling edge of the pulse, depending on whether you set 'Low to High' or 'High to Low'.

GW: Note that the mouse will not interrupt this type of triggered data acquisition. The only way it can be interrupted is by restarting the computer (or hitting the programmers switch might work).

Example 3

Suppose you are stimulating a preparation 12 times in a row, with 10 seconds between each stimulation for the preparation to recover. You would like to digitize 3 channels of data plus the channel giving the stimulation for 1 second after each stimulation. Furthermore, you would like to digitize all the stimulations at once instead of going back and doing them one at a time, but your computer doesn't have enough memory to hold the entire length of data. However, once all of the stimulations are recorded, you would like the data from each channel to be saved in a separate file, as is normally the case, with the data following each stimulation saved subsequently in the file to the data from the previous stimulation.

To do this use the Multiple Triggering function. First select the appropriate settings for 'Channels to Collect' and 'Sampling Rate' in the 'File' menu. Next, select 'Set Trigger' from the 'File' menu and set up the information in the dialog box, following the instructions in either Example 1 or Example 2 above, depending on where you want to trigger from. Each of the triggering events will be governed by these settings. However before clicking 'OK', click on the 'Multiple Triggering' button. This will bring up a second dialog box where you can set up the parameters for multiple triggering.

In the Multiple Triggering dialog box, set the 'Number of Triggers' equal to 12, since you are going to be stimulating 12 times. Since you want one data file per channel, with the data after each stimulation stored in the same file, check the 'One file per channel' box. Finally check the 'Multiple Triggering' box to turn on multiple triggering, and then click 'OK'. Now click 'OK' in the 'Set Trigger' dialog box.

To start the data acquisition, select 'Get Data' and go through the data collection process as you would normally. When you are asked how much data you want to collect, enter 1 second, so that one second of data will be collected after each stimulation. Now play your data into the digitizing board, including all 12 stimulations. As the program sees each stimulus, it will tell you that it has digitized it, and if something is wrong you can interrupt the process by pressing a key on the keyboard. When all 12 stimulations have taken place, there will be four new files on the disk corresponding to the four channels of data, and each file will be 12 seconds long. From 0 to 1 seconds will be the data after the first stimulation, from 1 to 2 seconds will be the data after the second stimulation, and so on.

Example 4

Suppose that, as above, you are stimulating a preparation 12 times in a row, with 10 seconds inbetween each stimulation for the preparation to relax. You would like to digitize data on channels 1, 3 and 4, plus channel 6 which is recording the stimulation, and you would like to collect data for 1 second after each stimulation. Instead of having all the data from each channel in one file, however, you would like to have separate files for each stimulation.

Go through all the same steps as in Example 3 for setting up the triggering and multiple triggering. However, in the Multiple Triggering dialog box, check 'One file per trigger' instead of 'One file per channel'.

Now when you select 'Get Data', the normal dialog box will appear, where you should choose the folder in which to save the data files. Then an additional dialog box will appear, prompting you for filenames for each channel. There will be input boxes labeled Channel 1, Channel 3, Channel 4, and Channel 6. Put a filename into each of these boxes. You may decide to call the data from the first channel "CellA", from the third channel "CellB", from the fourth channel "CellC", and from the sixth channel "Stimulator". The only limitation is that the names must not end in a number, and no two names should be the same.

The data acquisition will proceed as before. When it is finished, there will be 48 new data files on the disk, each 1 second long. They will be called "CellA1", "CellB1", "CellC1", and "Stimulator1" for the data after the first stimulation, "CellA2", "CellB2", "CellC2", and "Stimulator2" for the data after the second stimulation, and so on.

Application Note 3: Data Averaging

Spike Studio enables you to average data which follows a series of marked points. Below are three examples of using the data averaging feature. Before looking at these examples, you may want to read the Data Arithmetic section of the manual (p. 93).

Example 1

Suppose you have a channel which has recorded a series of EPSP's, and you would like to average these together to get an idealized EPSP. You define the start of an EPSP as the crossing of a certain threshold, and each EPSP is less than 100 msec long, so you only need to average 100 msec after the start of each EPSP.

To do this, first 'Open' the channel of data. Then move the 'Lower Threshold' so that it is at the threshold level which defines the start of an EPSP. Move the 'Upper Threshold' so that it is above the tops of all the EPSP's. Now calculate 'All Events' from the 'Analysis' menu. You should get an event onset at the beginning of each EPSP - check to make sure they all look reasonable. Next select 'Store Markers' from the 'Analysis' menu. This stores each of the event onsets as a marker point. Now you have marked off the beginning of each of the data sections that you want to average.

The next step is to 'Setup Data Storage' in the 'Data' menu. Selecting this item sets aside memory to hold the averaged data. When you select this function, a dialog box will ask how much data you would like the storage area to hold. Enter 100, so you will be averaging the first 100 msec after each marker, then click 'OK'. If the computer tells you there is not enough memory, there are a few things you can do to try to free up memory. First try closing any extra traces or graphs. If that's not enough, you can reduce the amount of memory allocated to each trace using the 'Set Memory' function in the 'Extras' menu. See Setting Memory Usage (p. 117) and Changing Memory Allocation (p. 116) for more detailed information.

Now select the 'Average Data' function from the 'Data' menu. A message will appear telling you that data has been averaged from the number of marker points which you set above. The averaged data is now in the data storage area, and you will see it in the Data Storage Window. To save it, click in the Data Storage window to make it the active window, or select it from the 'Windows' menu, and then use the 'Save' function in the 'File' menu. A normal Macintosh save dialog box will appear, into which you should type the filename you would like to save the averaged data under. Finally, 'Open' that file from the 'File' menu to look at the averaged data.

Example 2

Suppose instead of having only one channel of data, you have a channel of data and a channel which contains a series of stimulations, and you would like to average the data after each of the stimulations. Furthermore, suppose you have two different types of stimulations recorded in the same file, and you would like to average the data after type A stimulations separately from that after type B stimulations.

To do this, 'Open' both the stimulation file and the file which contains the data. Make the stimulation trace active, by clicking in it or selecting it from the 'Windows' menu. Using the event discrimination algorithm (p. 71), pick out the onsets and offsets of the stimulations. Now go through the file and delete the stimulations which are of type B. If type A and type B stimulations are in separate parts of the file, then you can have the program automatically pick out only type A by first adjusting 'Time From Start' and 'Seconds per Window' so all type A and no type B stimulations are shown on the screen, and then calculating 'Events on Screen' (press the 'e' key as a shortcut). As another alternative, you can use the 'Add Event' function to delineate only stimulations of type A (p. 77). Now select 'Store Markers' from the 'Analysis' menu, to store the onsets of each stimulation as a marker point.

Now 'Setup Data Storage', increasing memory available if needed as was done above in Example 1. Once you have the data storage area set aside, select 'Input Trace' from the 'Data' menu and select the data trace from the dialog box. This tells the computer that the data it is averaging should come from the upper channel on the screen. Then select 'Average Data', and finally select 'Save' from the 'File' menu, naming the file "TypeA_average".

Now go through and pick out only type B stimulations, using any one of the methods mentioned above, and making sure that all the previous events are deleted. Again select 'Store Markers', 'Average Data', and 'Save', this time saving the file as "TypeB_average".

You can repeat this procedure as many times as needed.

Example 3

If you have a prep which you subject to a variety of different stimulations, you may find that you want to average together data from parts of the recording which are separated by long stretches of time. For instance, following from the examples above, if you are recording EPSP's from a series of cells in a prep, you may want to average together all the EPSP's from all the cells when they were stimulated in a certain way. However, there may be quite a bit of recording inbetween the different cell stimulations, so that it is inconvenient to digitize all the data at once. Instead, you would like to digitize the data for one cell, pick out the EPSP's, and store the average of these EPSP's in memory, then digitize the data from the next cell and average the second cells EPSP's with those from the first cell, and so on.

To do this, follow either Example 1 or 2 above (depending on whether you have a second stimulation channel) for the data from the first cell. You should set up markers corresponding to the beginning of each piece of data which you would like averaged, and set up a data storage area of the appropriate length. However, instead of using the 'Average Data' function, select the 'Add Data From Markers' function from the 'Data' menu. This will add the data after each of the markers to the data storage area, without dividing by the number of markers. On the bottom of the screen, a message will tell you how many markers there were, which you should write down.

Now digitize or load in the second cell. Again, pick out appropriate markers. However, do not ‘Setup Data Storage’ again, as this will erase the data from the first cell. Instead, select ‘Add Data From Markers’, and again write down the number of markers there were. Repeat this procedure for each cell whose data you want to include in the average.

When the data from all the cells has been included, select the ‘Scale Data’ function from the ‘Data’ menu. Click on ‘Divide By’, and type in the total number of markers used (add together the number of markers from each cell). The data storage area will now contain the average of all the EPSP’s from all the cells, and you can save this to a file using the ‘Save’ function in the ‘File’ menu.