MAR-61-B-040695
Tone Editor 32X Ver. 1.00
SEGA Enterprises, Ltd.

Tone Editor 32X
Ver. 1.00

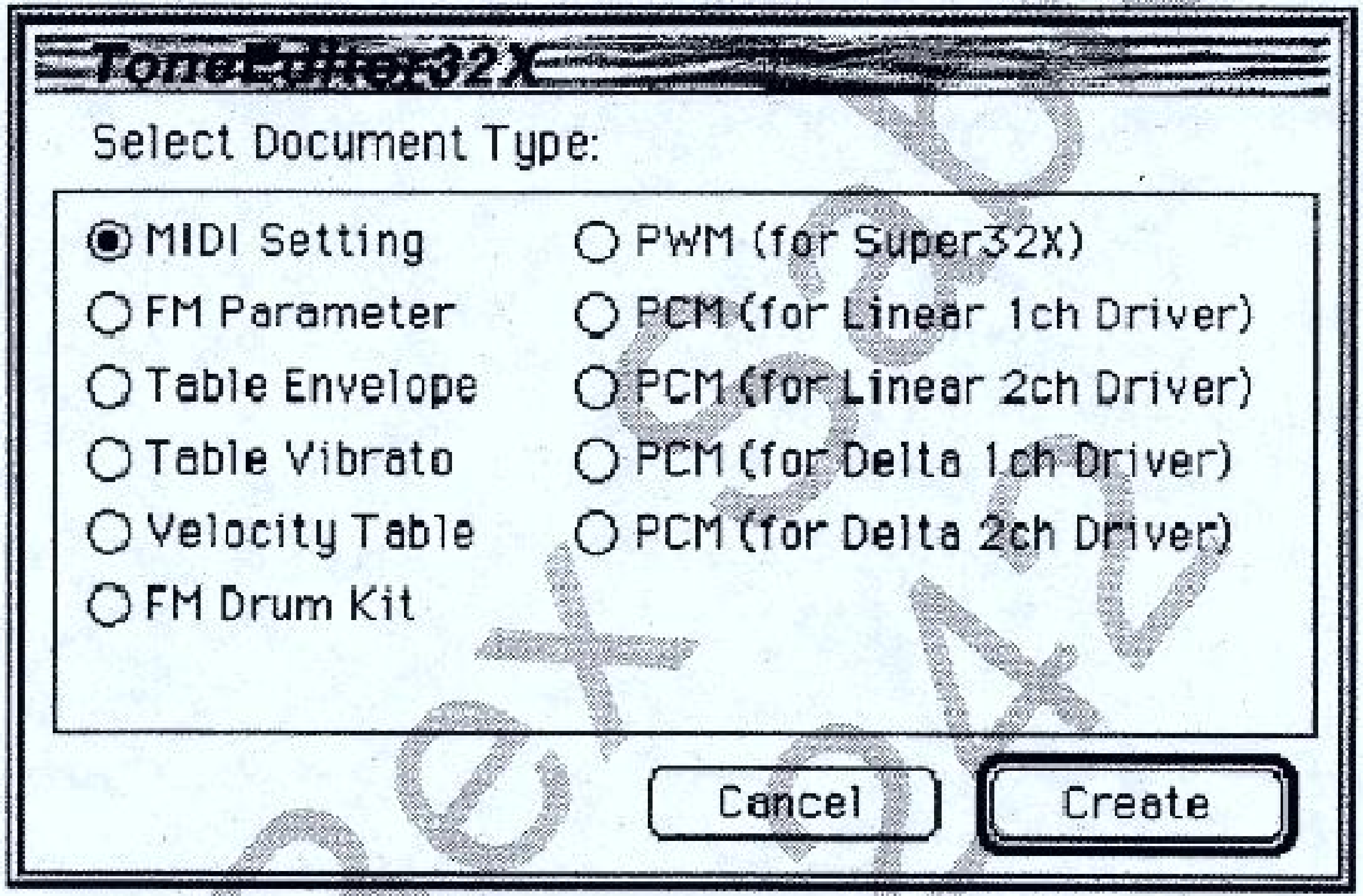
Table of Contents

Overview

Tone Editor 32X is a tool for developing 32X sounds using a MIDI. By replacing the Sound Driver that is to be mounted on a target machine, it is possible to use the Tone Editor 32X to develop mega drive sounds.

The 32X sound function includes a PWM stereo 1 sound source, in addition to the FM sound source 6 channel, the PSG 3 channel and the PCM 1 sound that are available in a conventional mega drive. The Tone Editor 32X allows the user to develop sounds by using all these resources.

When you select the "File" New option from the menu bar after starting the Tone Editor 32X, the following screen appears. On this screen, check off the ° mark by the item to be selected and click on "Create". This moves the cursor to the desired item.


ToneEditor32X

Select Document Type:

[✓ MIDI Setting]
FM Parameter
Table Envelope
Table Vibrato
Velocity Table
FM Drum Kit
PWM (for Super32X)
PCM (for Linear 1ch Driver)
PCM (for Linear 2ch Driver)
PCM (for Delta 1ch Driver)
PCM (for Delta 2ch Driver)

On each item, create the necessary data for generating a sound according to prescribed procedures. After creating the data, save the data in a file, in order to store the data in a format that allows the data to be automatically incorporated into the sound simulator (except for MIDI sequence data).

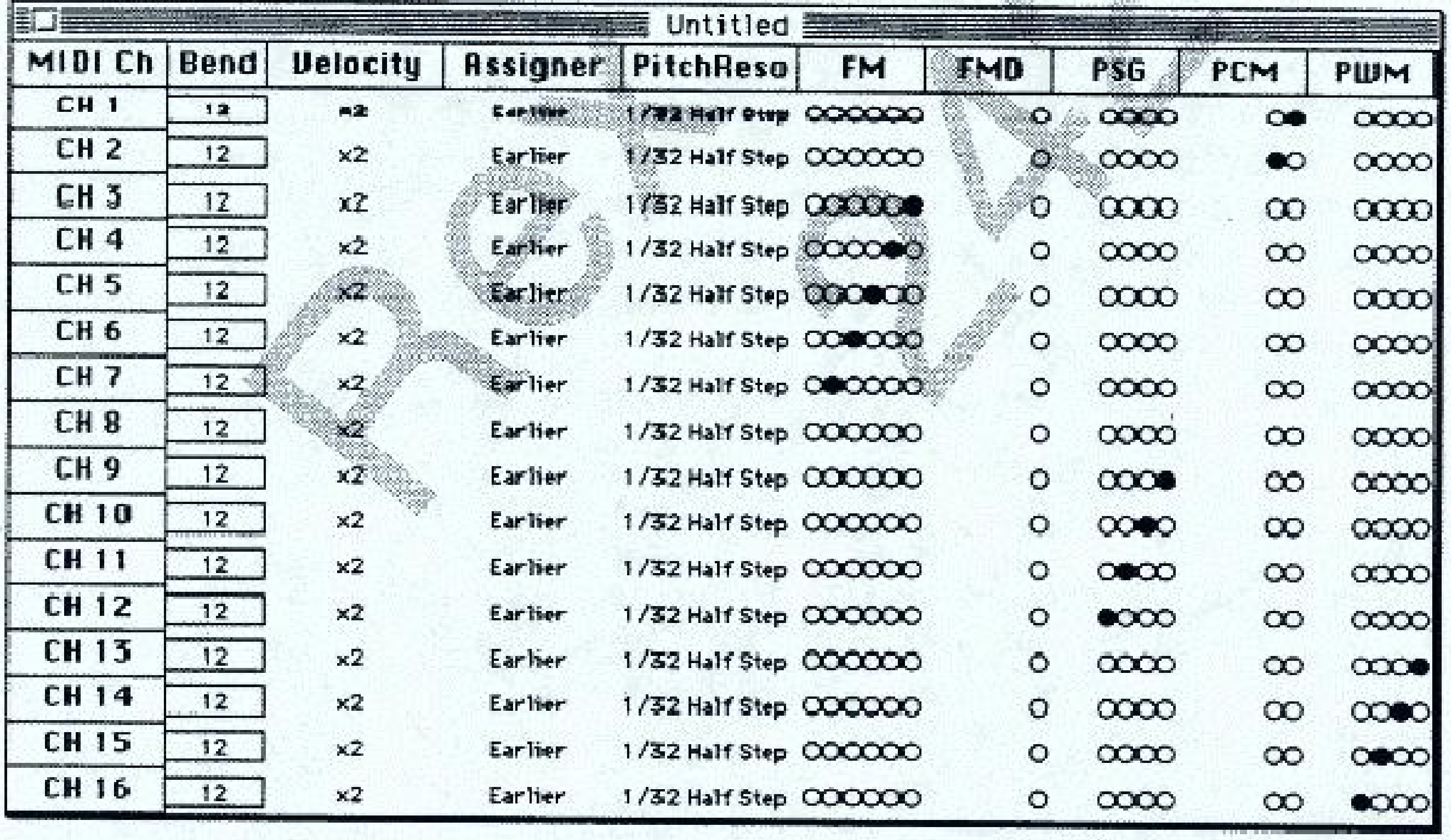
1. MIDI Setting

When using the Tone Editor 32X for the first time, you need to establish the necessary interface. To do so, first, you need to perform MIDI setting, which involves determining the correspondence between the channels associated with MIDI instruments and the sound sources associated with 32X, and defining the functions in which the channels and the sound sources will be operated.

In the graphics below, the item marked with a l indicates the 32X sound source that currently supports a given MIDI channel (MIDI Ch). The sound source channels on the 32X side are lined up from right to left, in ascending order of numbers. It is not necessary that all sound sources be in one-to-one correspondence with MIDI channels. Rather, sound sources can be specified polyphonically from a given MIDI channel.

Both PCM and PWM have only one voice, due to hardware limitations. However, by performing a waveform synthesis, different voices can be output as separate channels. This feature is compatible with all the tools and Sound Drivers that are supplied in a 32X Sound Tool Package.

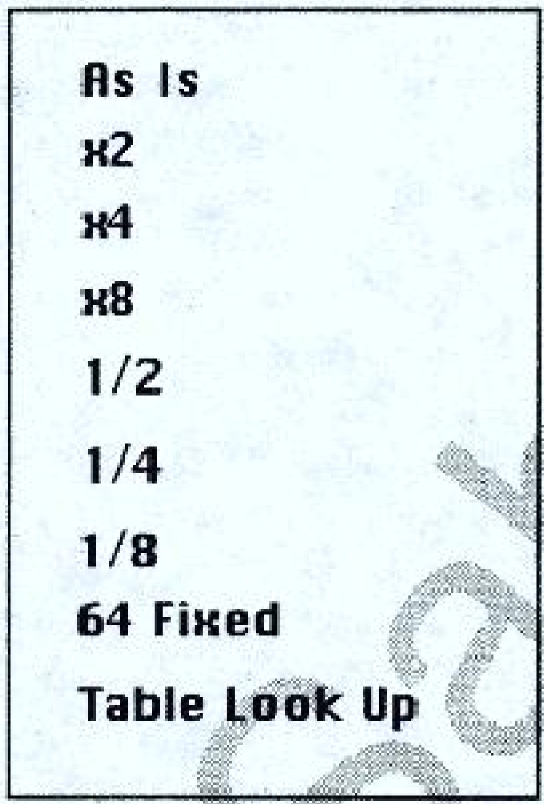
(*) For mega drives and the 32X, the sixth channel of the FM sound source are used by switching the same register as the PCM sound source. Therefore, when a PCM is used, the sixth channel does not work. For details on the sound chips that are incorporated in the mega drive itself, consult the "MEGA DRIVE SOUND MANUAL."


Untitled

MIDI Ch	Bend	Velocity	Assigner	PitchReso	FM	FMD	PSG	PCM	PWM
CH 1	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	○●	○○○○
CH 2	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	●○	○○○○
CH 3	12	x2	Earlier	1/32 Half Step	○○○○○●	○	○○○○	○○	○○○○
CH 4	12	x2	Earlier	1/32 Half Step	○○○○●○	○	○○○○	○○	○○○○
CH 5	12	x2	Earlier	1/32 Half Step	○○○●○○	○	○○○○	○○	○○○○
CH 6	12	x2	Earlier	1/32 Half Step	○○●○○○	○	○○○○	○○	○○○○
CH 7	12	x2	Earlier	1/32 Half Step	○●○○○○	○	○○○○	○○	○○○○
CH 8	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	○○	○○○○
CH 9	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○●	○○	○○○○
CH 10	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○●○	○○	○○○○
CH 11	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○●○○	○○	○○○○
CH 12	12	x2	Earlier	1/32 Half Step	○○○○○○	○	●○○○	○○	○○○○
CH 13	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	○○	○○○●
CH 14	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	○○	○○●○
CH 15	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	○○	○●○○
CH 16	12	x2	Earlier	1/32 Half Step	○○○○○○	○	○○○○	○○	●○○○

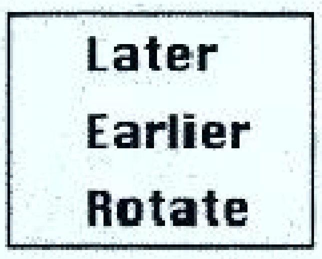
Bend
This field allows you to set and change a bend width. For each channel, specify the desired numerical value by clicking on the channel. The allowable values are 0-12 (one octave).

Velocity
Clicking on this field brings up a popup menu. By selecting an appropriate field in this menu, you can define how to handle a velocity.



As Is ... The velocity data transmitted by the MIDI is used as is.
X2 ... The reaction to the velocity data transmitted by the MIDI is increased by 2.
X4 ... The reaction to the velocity data transmitted by the MIDI is increased by 4.
X8 ... The reaction to the velocity data transmitted by the MIDI is increased by 8.
1/2 ... The reaction to the velocity data transmitted by the MIDI is reduced to 1/2.
1/4 ... The reaction to the velocity data transmitted by the MIDI is reduced to 1/4.
1/8 ... The reaction to the velocity data transmitted by the MIDI is reduced to 1/8.
64 Fixed ... All velocity data transmitted by the MIDI is fixed to 64. This is the same as ignoring all velocities.
Table Lookup ... This causes the program to read the values set in the Velocity Table (more on this later).

Assigner
This assigns a polyphonic key. A polyphonic key defines the sound generation priority to be used when the data transmitted by the MIDI is greater than the number of sounds that can be generated simultaneously. Select an assigner for each channel from the popup menu.



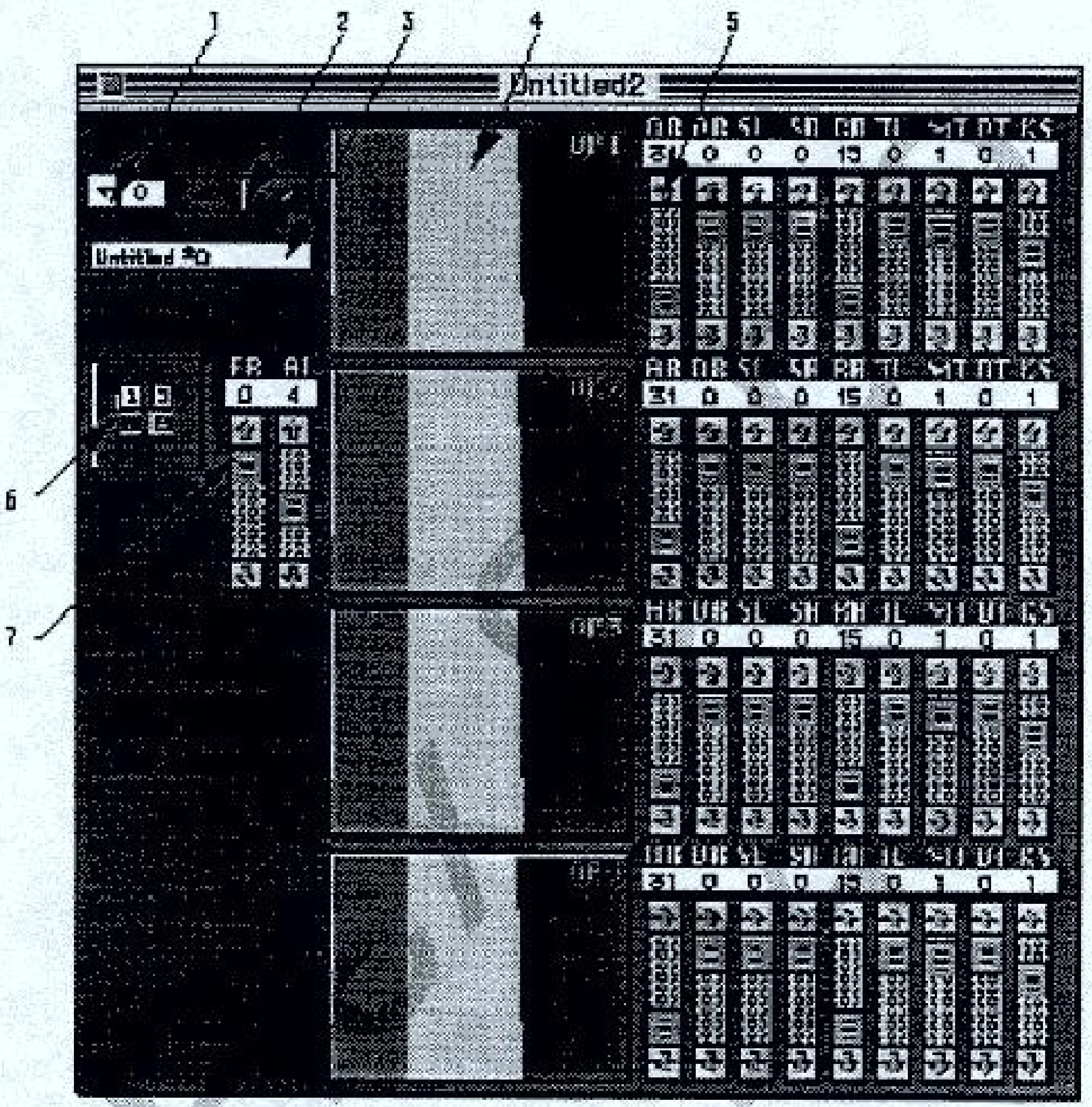
Later ... First in, first served
Earlier ... Last in, first served
Rotate ... This is the last-in, first-served polyphonic to be used when multiple 32X-side sound sources are specified for a MIDI channel.

PitchReso
This field determines the resolution for the pitch bend, detune, and vibrato options. However, because the 68000 Sound Driver does not support this function, perform the necessary modifications from the default "1/32 HalfStep".

2. FM Parameters

The FM Parameters menu allows you to create and edit timbres for an FM sound source. Files that are created on this menu form the timbre data that is stored in a cartridge without any modification. A maximum of 128 timbres can be stored per file. However, only one FM timbre file can be stored per bulk file. Thus, the maximum number of FM timbres that can be set per project is 128.

FM timbre files only contain timbre data and exist separately from sequence data.



Due to space limitations, no attempt will be made in this manual to touch upon the functions of FM sound sources and the roles of parameters. For details on these topics, see the "YM2612 APPLICATION MANUAL".

1. Listing up a timbre data file

This function displays a list of existing timbre data files. To open the desired file on the list, drag the name of the file.

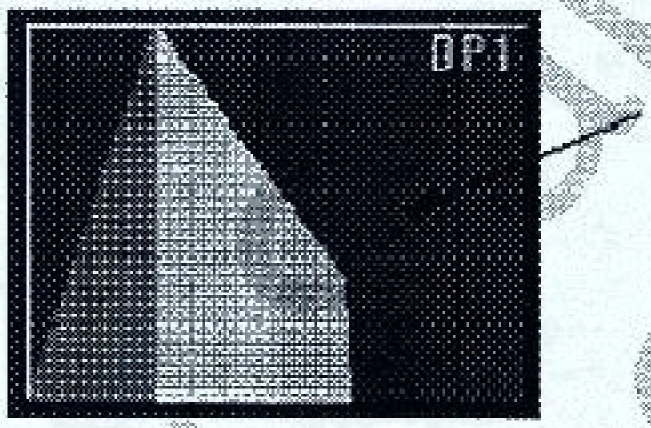
2. Selecting a timbre data file

This function is similar to function 1 above. It allows you to display timbre data files by pressing the UP and DOWN keys.

3. Name of the currently edited file
By clicking on this field, you can name or rename a file by entering the desired file name from the keyboard. You can use this function to assign appropriate file names to keep track of the files.

4. Shapes of Operator Envelopes
This function graphically displays the envelope shape of a given operator. The function displays the ADSR in blue, red, green, and yellow, respectively. Using this function, you can graphically edit the shape of an operator by dragging a colored portion.

The shape of an envelope can be copied between operators by dragging any part which is not an envelope (see the arrow in the figure below) on the graphics, and by overlaying the dragged portion on the envelope-shape graphics for the target operator.



5. Parameter-Setting Sliders
A slider allows you to display and manipulate a parameter that represents a continuously varying quantity. By clicking on any point on a slider, you can set a change in the vertical direction on the screen as an FM envelope parameter. In doing so, you can continuously manipulate a value by dragging the parameter.

You can also manipulate a parameter continuously by clicking on the UP and DOWN arrows. To cause a stepwise change -- one step per clicking operation -- click on the UP and DOWN arrows while pressing the *option* key.

How a parameter value changes as a function of slider (scrollbar) movements can be monitored in terms of numerical values that are displayed adjacent to the slider. Conversely, you can change the parameter by clicking on these numerical values.

6. Currently selected algorithm
In addition to using an FM parameter as a monitor, you can also select and set an algorithm by dragging a pointer with respect to a graphic.

7. Algorithm/feedback slider

Similar to the parameter-setting slider, a slider can be used to set an algorithm or to define a feedback scheme.

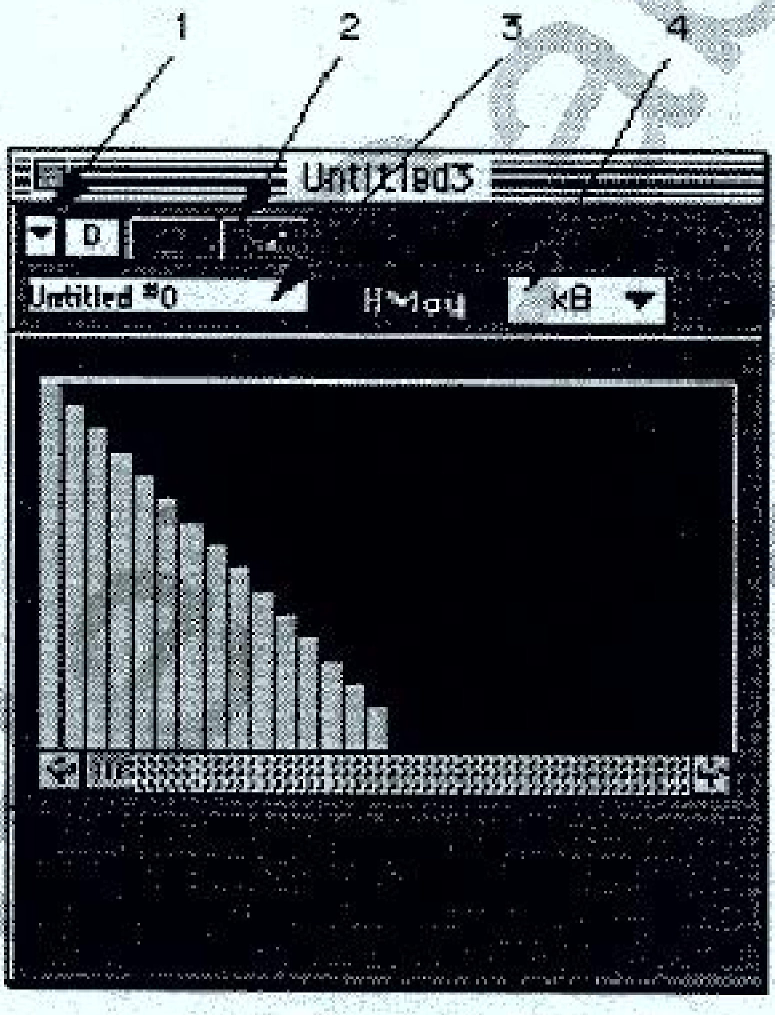
3. Table Envelope

The table envelope function defines the software envelope shape for the PSG. Similar to the FM timbre files, the PSG envelopes are stored as a data array, separate from the sequence data.

In the bar graph that is displayed, the vertical axis (Y) represents the PSG volume; the horizontal axis (X) represents the time axis. In terms of time, one bar corresponds to one V-Interrupt.

To edit the software envelope shape directly, you can drag the pointer on the graphics. The Y can be manipulated by dragging a bar. When the bar is dragged, there is no corresponding change in X. To move the X, drag the X while holding the *option* key down. This operation does not affect the Y.

The yellow bar at the end of the graphics indicates the end of the envelope data. The yellow bar continues to sound until it is keyed off as a sustained sound. To perform keying off, set the end bar to zero (0).



1. Listing a timbre data file

2. Selecting a timbre data file

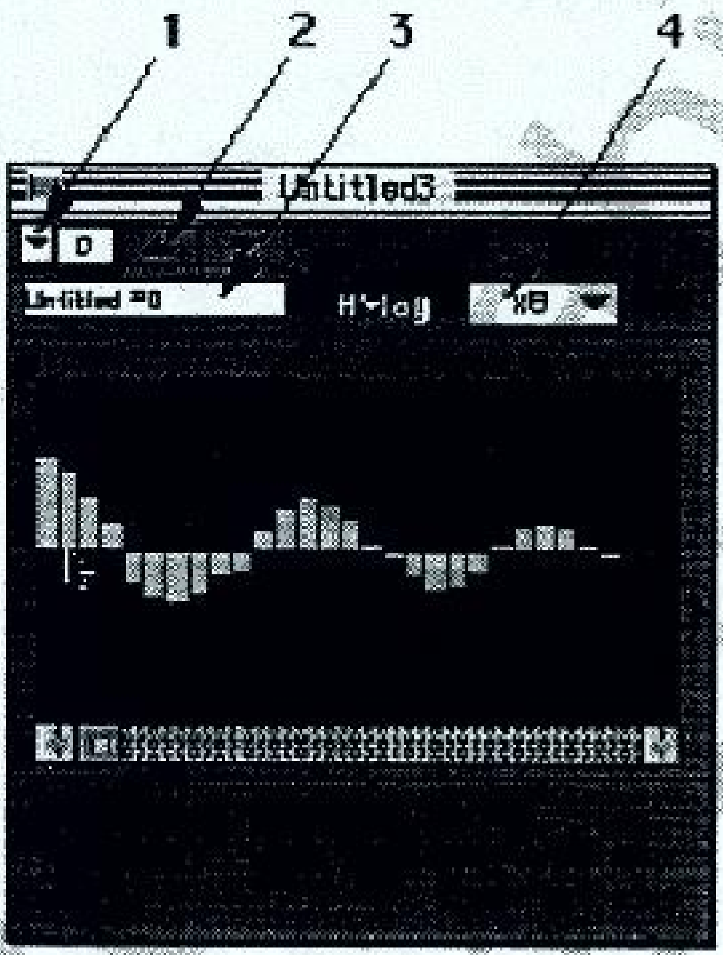
3. Name of the file being edited
See the section on "FM Parameters".

4. Display magnification
The rate of enlargement for the time axis can be chosen from among a 1-, 2-, 4-, or 8-fold increase.

4. Table Vibrato

Use the same method as for the table envelope for setting the table vibrato. In the table vibrato, the amount Y, measured from the 0 value on X (the leftmost position), is added to the frequency at a V-Interrupt timing. On the display, a loop is performed until a key-off signal appears at the portion indicated by the red bar. The center line extending parallel to the X indicates that any bars extending up from the center lines are positive values, and any bars extending down from the center line are negative values. These bars represent period data by which musical interval frequencies are added or subtracted.

The length of a vibrato can be changed by clicking on the end position while holding down the command key. To commit a loop point, click on the loop point while holding down the *option* key.



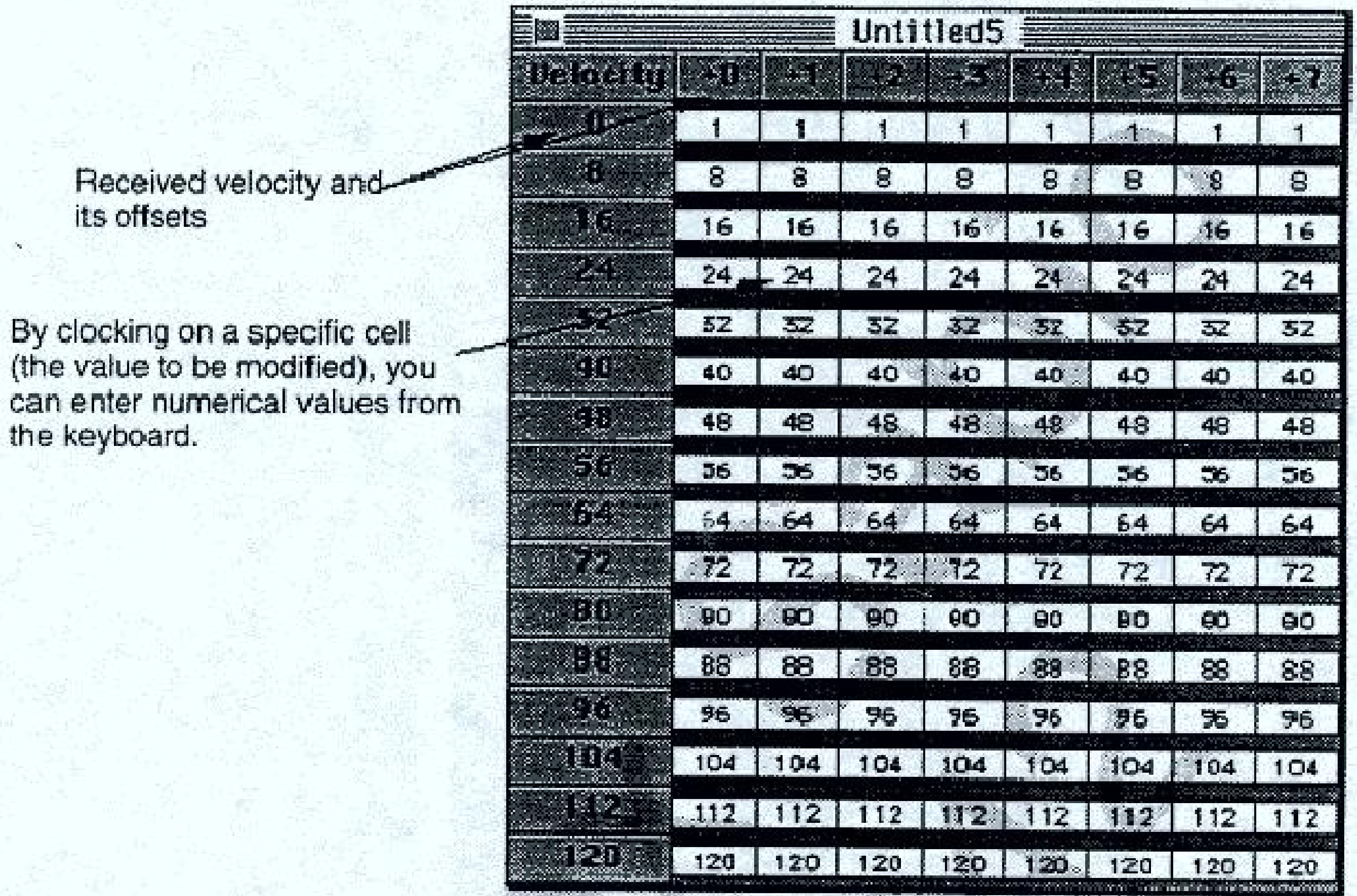
1 Listing a timbre data file
2 Selecting a timbre data file
3 Name of the file being edited
See the section on "FM Parameters".

4 Display magnification
The rate of enlargement for the time axis can be chosen from among a 1-, 2-, 4-, or 8-fold increase.

5. Velocity Table

The velocity data 0-127 that is output by the MIDI is absorbed on the 32X in 16 stages, 1-15. The result is a reduced resolution. Therefore, the MIDI data velocity that is received can be processed after the data is converted into data that corresponds to the velocity table.

To enable the velocity table, first perform the setting described in the section on *MIDI Setting*.



Received velocity and its offsets

By clocking on a specific cell (the value to be modified), you can enter numerical values from the keyboard.

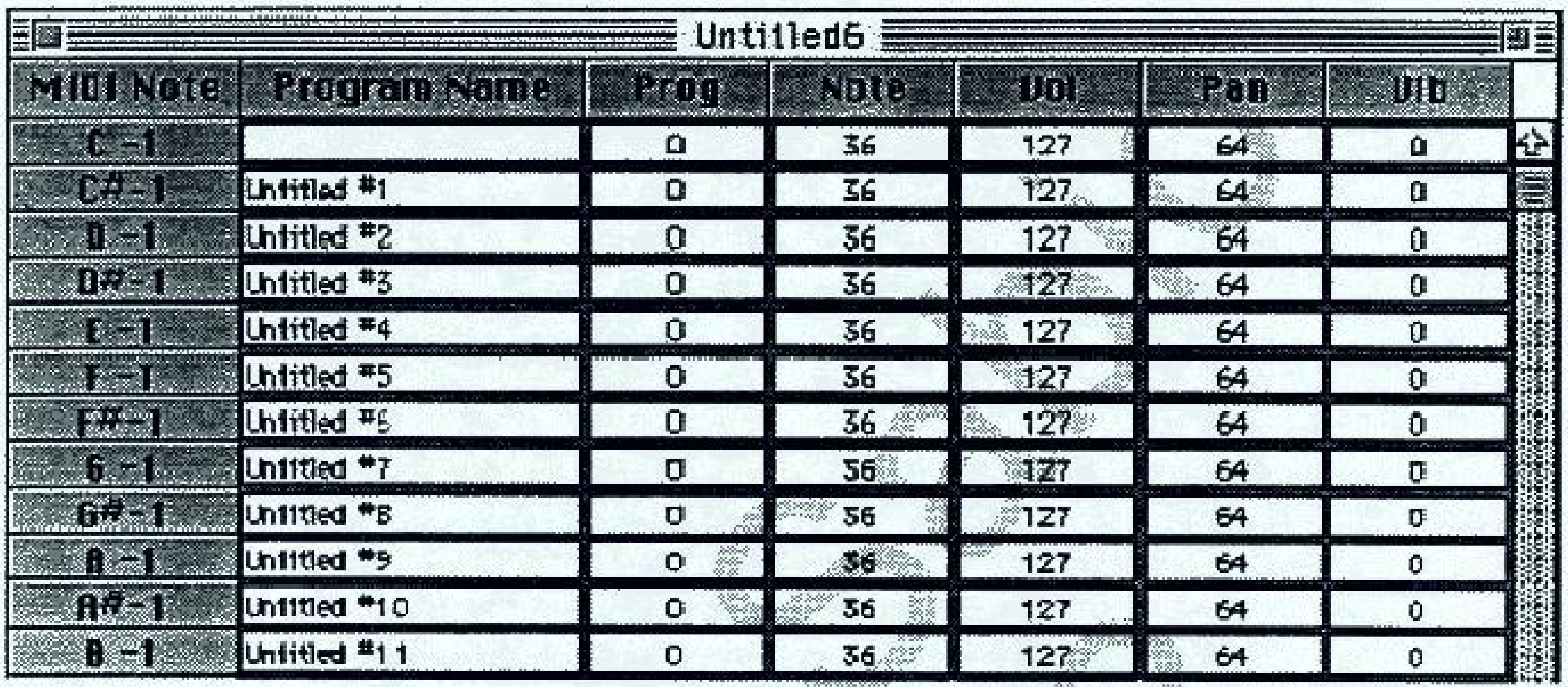
Untitled5

Velocity	+0	+1	+2	+3	+4	+5	+6	+7
0	1	1	1	1	1	1	1	1
8	8	8	8	8	8	8	8	8
16	16	16	16	16	16	16	16	16
24	24	24	24	24	24	24	24	24
32	32	32	32	32	32	32	32	32
40	40	40	40	40	40	40	40	40
48	48	48	48	48	48	48	48	48
56	56	56	56	56	56	56	56	56
64	64	64	64	64	64	64	64	64
72	72	72	72	72	72	72	72	72
80	80	80	80	80	80	80	80	80
88	88	88	88	88	88	88	88	88
96	96	96	96	96	96	96	96	96
104	104	104	104	104	104	104	104	104
112	112	112	112	112	112	112	112	112
120	120	120	120	120	120	120	120	120

6. FM Drum Kit

The FM Drum Kit refers to a function that is designed to save memory by putting the data array having all information, such as musical interval, timbre, PAN, and vibrato numbers, in one-to-one correspondence with the "note assign" that is contained in the 68000 Sound Driver. Thus, by assigning a note number to the 68000 Sound Driver, the FM Drum Kit can execute a series of predetermined procedures.

Before creating an FM Drum Kit, you need to complete the tasks defined in the sections on FM Parameters and Table Vibrato.


Untitled6

MIDI Note	Program Name	Prog	Note	Vol	Pan	Vib
C -1		0	36	127	64	0
C#-1	Untitled #1	0	36	127	64	0
D -1	Untitled #2	0	36	127	64	0
D#-1	Untitled #3	0	36	127	64	0
E -1	Untitled #4	0	36	127	64	0
F -1	Untitled #5	0	36	127	64	0
F#-1	Untitled #6	0	36	127	64	0
G -1	Untitled #7	0	36	127	64	0
G#-1	Untitled #8	0	36	127	64	0
A -1	Untitled #9	0	36	127	64	0
A#-1	Untitled #10	0	36	127	64	0
B -1	Untitled #11	0	36	127	64	0

Set the following six items relative to the "note assign" that is indicated in the MIDI Note:

Program Name ... Assign a name as a part of the Drum Kit to be created.
Prog ... Select the timbre created using an FM parameter.
Note ... Set the desires musical interval by defining C-1 as 0. Therefore, the initial value 32 denotes C2 (the sound "do" in the octave 2).

Vol ... Set the volume, with a maximum of 127 and a minimum of 0.

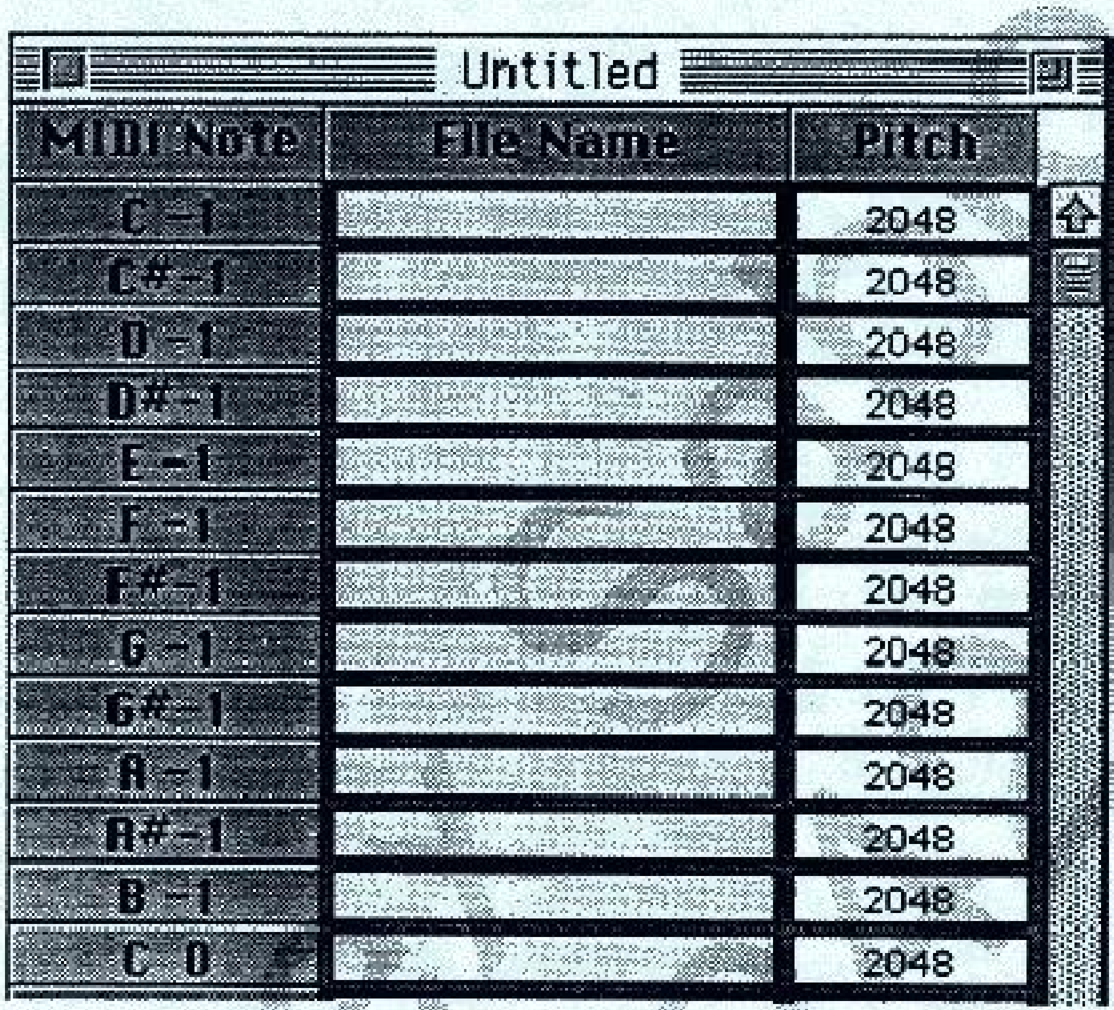
Pan ... Set the desired panning position. Since the pan positions for an FM sound source can be determined by setting the right and left as on and off, there are only three types of positions. The values 0-41 indicate left output, the values 0-41 a left output only, the values 42-84 right-and-left symmetric output, and the values 85-127 indicate left output only.

Vib ... Set the number for a created table vibrato.

7. PWM (for Super 32X)

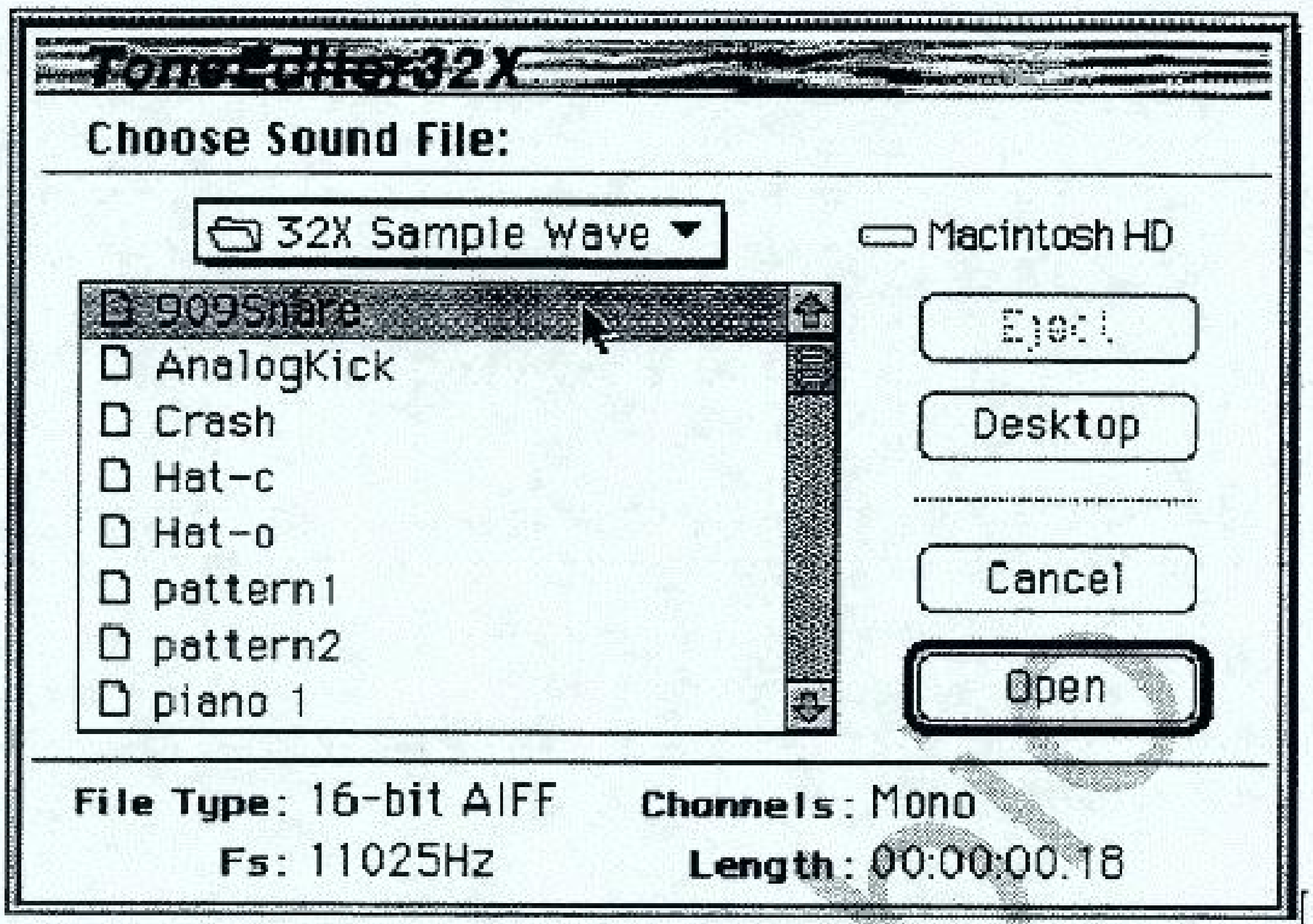
A sound source is required to use a PWM sound source. Therefore, sampling must first be conducted in order to use a PWM sound source. The following sampling data files can be used: 8-, 16-, or 24-bit AIFF file [Sound Designer 2] or a formatted file. These sampling data files are compatible with any of the following formats: monaural, stereo, or 4-ch stereo, as long as the data contained in them is less than equal to a 11025 Hz sampling frequency.

After sampling data files have been set up, allocate them to "note assigns", as in the case of the FM Drum Kit. The sampling data now corresponds to the received "note assign", and generates sounds in a defined pitch.


Untitled

MIDI Note	File Name	Pitch
C -1		2048
C# -1		2048
D -1		2048
D# -1		2048
E -1		2048
F -1		2048
F# -1		2048
G -1		2048
G# -1		2048
A -1		2048
A# -1		2048
B -1		2048
C 0		2048

File name
By entering the names of the created sampling data files, you can establish correspondence between the note assigns and the files. To save memory, files are normally assigned in descending order. To open the popup menu, click on the desired cell. On the popup menu, select the desired sampling data file.



File Type: 16-bit AIFF Channels: Mono
Fs: 11025Hz Length: 00:00:00.18

Pitch
Note-assigns can be pronounced with a pitch. The displayed numerical value is called an address-up counter; it is a parameter that is used to read samples on a phase basis. This parameter (N) is treated internally in the PWM driver as a 12-bit fractional number. It is incremented at every 11025 Hz, which is the PWM driver's fixed cycle. When N>=1, one sample is read. The address-up counter is output using the same Fs (original pitch) as when the address-up counter is sampled at 2048 (800H).

Note that the address-up counter cannot be set to a value greater than 2048, i.e., the address-up counter cannot be output at a frequency higher than the original pitch. Therefore, when assigning an address-up counter to a part that requires a musical interval in a piece of music, be careful to arrange the music properly so that this restriction is complied with.

For the sake of illustration, assume that the sampled octave is denoted OCn, an 11025 Hz sampling rate is used, and the sampled musical interval is A ("ra"). The table below provides a list of the address-up counters that are calculated from the 12-temperament. To the extent that the musical interval is lowered, however, the area-sing noise increases. Therefore, be careful to ensure a proper balance between the pitch and the sound quality.

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
OCn-5										64	68	72
OCn-4	76	81	86	91	96	102	108	114	121	128	136	144
OCn-3	152	161	171	181	192	203	215	228	242	256	271	287
OCn-2	304	323	342	362	384	406	431	456	483	512	542	574
OCn-1	609	645	683	724	767	812	861	912	967	1024	1085	1149
OCn	1218	1290	1367	1448	1534	1625	1722	1825	1933	2048		

If a sampling is performed at any other setting, the sampled musical-interval frequency (Hz) is denoted I, the sampling rate is denoted f, the musical frequency (Hz) that is actually output is denoted O, and the address-up counter N is calculated using the formula given below:

$N = 2048 * O/I * f/11025$ (where $N <= 2048$)

Notes:

- An N value greater than 2048 can cause the system to malfunction. The PWM driver does not change incorrect setup values.

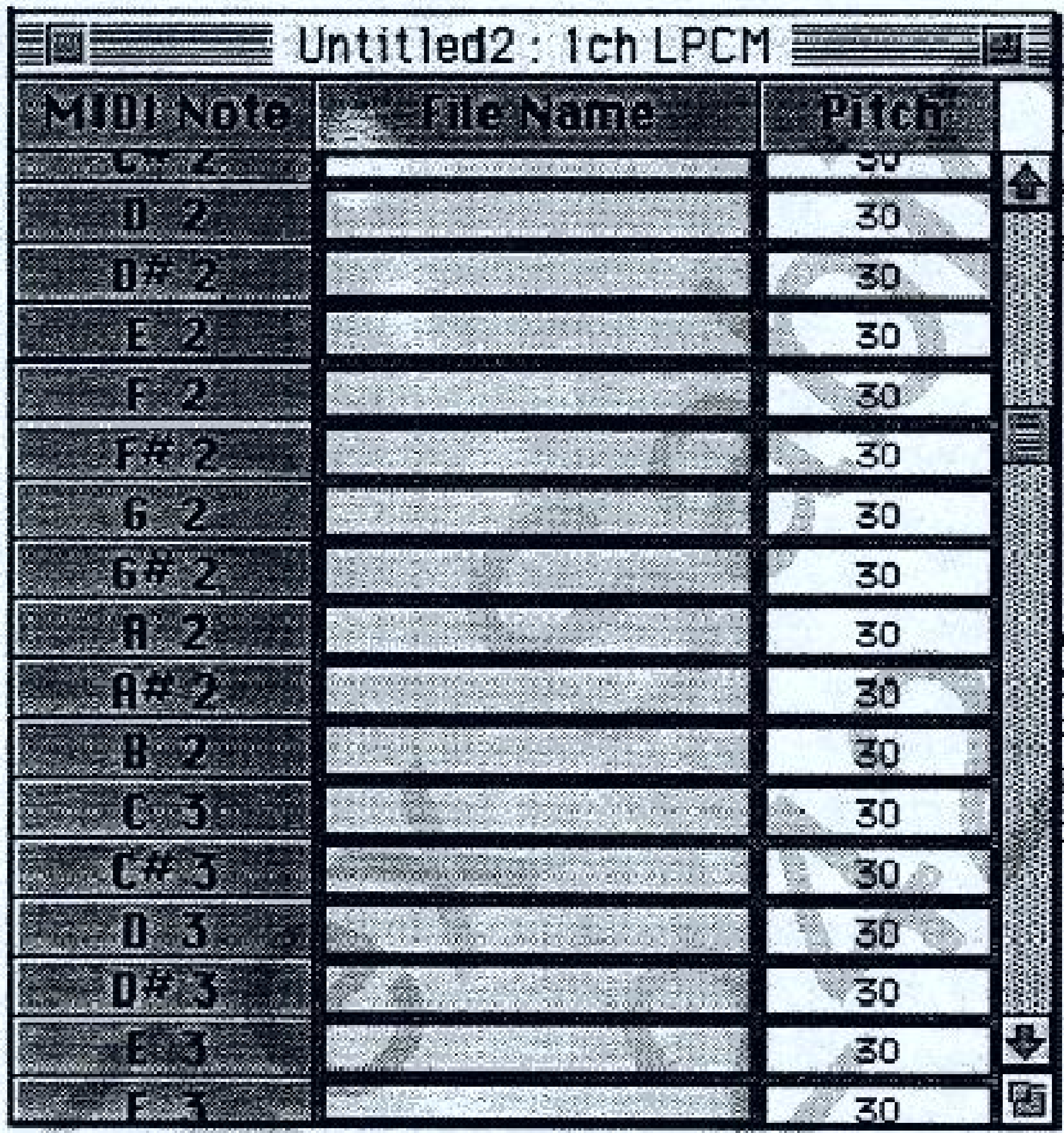
- Further details on musical interval frequencies may be found in books on music or in the "PSG Manual" for the Mega Drive.

- Because the 32X uses a bank-switchable memory map (see the "Super 32X Hardware Manual"), the 32X is designed with the assumption that sound data is allocated to areas that are not subject to bank-switching, i.e., from 0x880000 to 0x8FFFFF. Consequently, if the sound data is too large to fit within the indicated addresses, an error results in the sound simulator. In most cases, the amount of sampling data is the cause of the sound data becoming exceedingly large. Therefore, when creating sampling data, be careful that its amount remains within reasonable bounds.

8. PCM

For the Tone Editor 32X, you can use any PCM driver from among the 8-bit linear PCM 1ch, 8-bit linear PCM 2ch, 4-bit delta PCM 1ch, and 4-bit delta PCM 2ch. To select a PCM driver, check off the desired PCM driver in the "Select Document Type" menu. Notice that the playback rate may differ from driver to driver.

As in the case of a PWM sound source, PCM sound sources use either 8-bit or 16-bit AIFF file or a "Sound Designer 2" format file as a sound source. Because PCM sound sources use the same sampling data file assignment method as a PWM sound source, the following describes differences between PCM and PWM sound sources.



MIDI Note	File Name	Pitch
D 2		30
D# 2		30
E 2		30
F 2		30
F# 2		30
G 2		30
G# 2		30
A 2		30
A# 2		30
B 2		30
C 3		30
C# 3		30
D 3		30
D# 3		30
E 3		30
F 3		30

Pitch
Allowable values are 1-256. The lower the parameter value, the higher the musical interval. In the case of a 2-ch driver, for which the pitch is fixed, the Pitch column is not displayed.

- The Tone Editor 32X contains an internal, built-in PWM driver and a sound driver, which are replaceable. To replace a sound driver, name the file for the new sound driver as either "MD_DRV.BIN" (binary file) or "MD_DRV.S28 (Motorola-S format file), and store the file name in the folder in which Tone Editor 32X is located. When this file is not found, the Tone Editor 32X loads the built-in sound driver by default. Any of the sound drivers that are provided should be loaded using this method. Similarly, the PWM Driver should be named as "SH_PRG.P".