

## **General Notice**

When using this document, keep the following in mind:

1. This document is confidential. By accepting this document you acknowledge that you are bound by the terms set forth in the non-disclosure and confidentiality agreement signed separately and /in the possession of SEGA. If you have not signed such a non-disclosure agreement, please contact SEGA immediately and return this document to SEGA.
2. This document may include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new versions of the document. SEGA may make improvements and/or changes in the product(s) and/or the program(s) described in this document at any time.
3. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without SEGA'S written permission. Request for copies of this document and for technical information about SEGA products must be made to your authorized SEGA Technical Services representative.
4. No license is granted by implication or otherwise under any patents, copyrights, trademarks, or other intellectual property rights of SEGA Enterprises, Ltd., SEGA of America, Inc., or any third party.
5. Software, circuitry, and other examples described herein are meant merely to indicate the characteristics and performance of SEGA'S products. SEGA assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples describe herein.
6. It is possible that this document may contain reference to, or information about, SEGA products (development hardware/software) or services that are not provided in countries other than Japan. Such references/information must not be construed to mean that SEGA intends to provide such SEGA products or services in countries other than Japan. Any reference of a SEGA licensed product/program in this document is not intended to state or simply that you can use only SEGA'S licensed products/programs. Any functionally equivalent hardware/software can be used instead.
7. SEGA will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user'S equipment, or programs according to this document.

NOTE: A reader's comment/correction form is provided with this document. Please address comments to :

SEGA of America, Inc., Developer Technical Support (att. Evelyn Merritt)  
150 Shoreline Drive, Redwood City, CA 94065

SEGA may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.



SEGA OF AMERICA, INC.  
Consumer Products Division

SEGA Confidential

**SATURN**  
**Sound Development**  
**Manual**  
ver. 1.1

Doc. # ST-081-R5-062894

## READER CORRECTION/COMMENT SHEET

### Keep us updated!

If you should come across any incorrect or outdated information while reading through the attached document, or come up with any questions or comments, please let us know so that we can make the required changes in subsequent revisions. Simply fill out all information below and return this form to the Developer Technical Support Manager at the address below. Please make more copies of this form if more space is needed. Thank you.

### General Information:

Your Name \_\_\_\_\_ Phone \_\_\_\_\_

Document number ST-081-R5-062894 Date \_\_\_\_\_

Document name SATURN Sound Development Manual, ver. 1.1

### Corrections:

Chpt.	pg. #	Correction

Questions/comments: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

### Where to send your corrections:

Fax: (415) 802-1440  
Attn: Sr. Coordinator,  
Technical Publications Group

Mail: SEGA OF AMERICA  
Attn: Sr. Coordinator,  
Technical Publications Group  
130 Shoreline Dr.  
Redwood City, CA 94065

# SATURN Sound Development Manual, ver 1.1

1. Overview .....	3
System Overview .....	3
Development Overview .....	5
General Sound Development Procedures .....	5
Tone Bank Data Composition (Tone Components) .....	7
FM (Frequency Modulation) .....	8
Usable MIDI Commands .....	8
MIDI Channels and Voices .....	9
2. Sound Simulator .....	10
Main Window .....	10
The Sound Simulator .....	10
3. How to Use the Sound Simulator .....	12
Menu Bar .....	12
File Menu .....	12
Edit Menu .....	15
Map Menu .....	16
Window Menu .....	17
Option Menu .....	17
4. Sound Driver System Interface .....	24
System Area Contents .....	24
System Interface Area .....	26
Sound Control Command .....	33
Host Interface .....	37
PCM Stream Play .....	39
Appendix: MIDI Specifications .....	42
Transmitted Data .....	42
Recognized Receive Data .....	43
System Messages .....	44
MIDI Exclusive .....	44
Index .....	50

(There is no page 2 in the original Japanese document.)

SEGA Confidential



# 1. Overview

## System Overview

### • Hardware Requirements

Device	Model	Description
Development host machine	Macintosh	Macintosh II series or later with SCSI interface. Operating system: KanjiTalk 7 or System 7 or later RAM: 16 MB or more HDD: 300 MB or larger and 1 GB or larger when making HD recordings
Saturn Target Box		Can operate with sound board above (without main board).
MIDI instrument	MIDI keyboard, etc.	Instrument with MIDI output capability Used for tone development, composition, and creating sound effects
Audio equipment	CD player, DAT, etc.	* Device capable of digital output Used for editing waveforms and HD recording
MIDI interface	Studio 5, etc.	MIDI interface for Macintosh

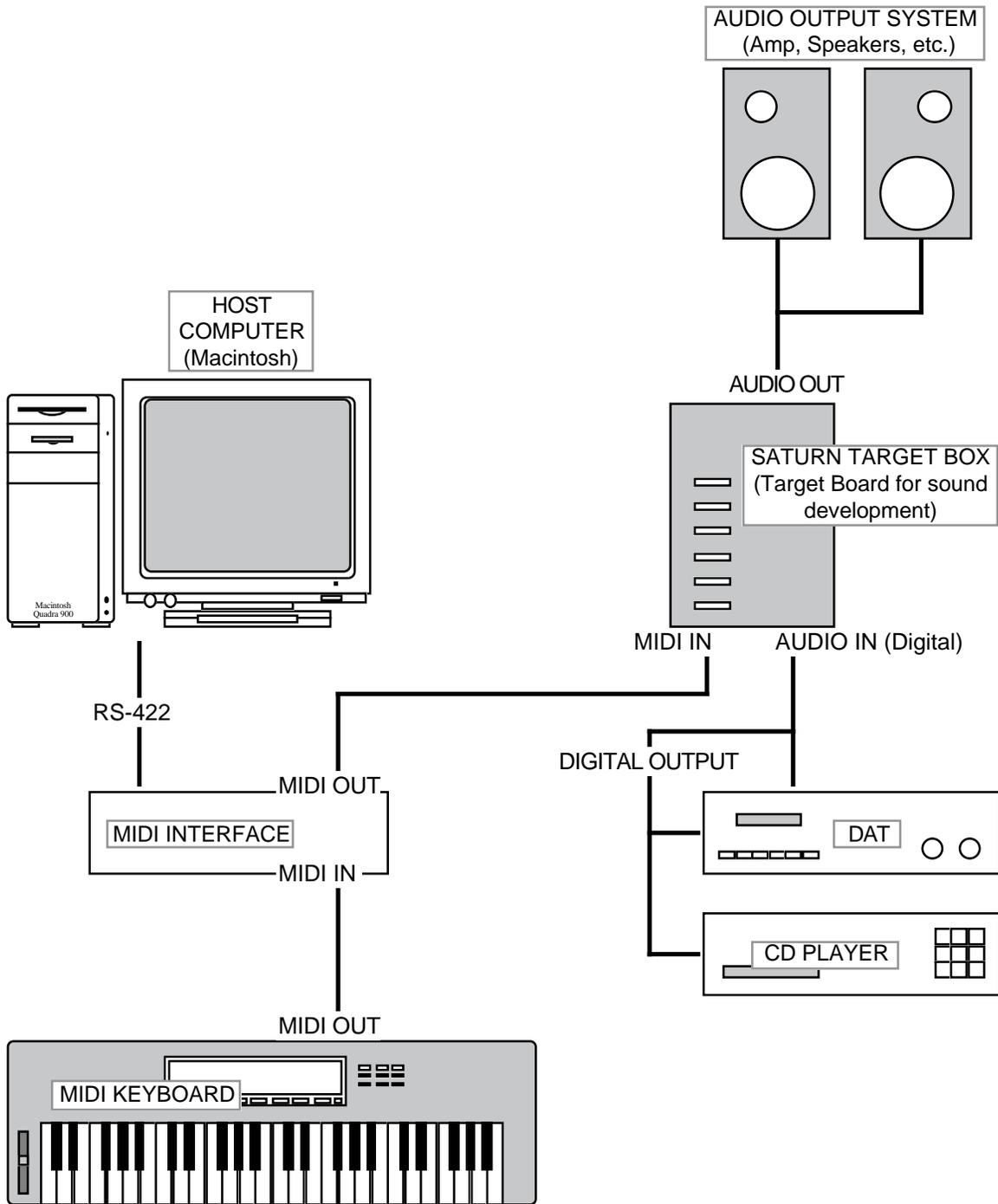
### • Software Requirements

System	Software	User	Description
Tone development tool	Waveform editor	SEGA Products	Used to edit waveforms and HD recordings Can also be used with by waveform edit tools available on the market (devices that support AIFF formats such as Alchemy, SDII) FM/PCM tone development DSP program development
	Tone editor DSP linker	SEGA SEGA	
Composition tools	MIDI sequencer	Products	Digital Performer (Mark of the Unicorn) Studio Vision (Opcode Systems) Cubase Audio (Steinberg), etc.
Sound development support tool	Master system	SEGA	Map Tool Sound Simulator Optimization Linker

The tools listed below are for writing and changing sound drivers and other programs, and are not required for the development of sound data (tunes and sound effects).

System	Software	User	Description
Program development tools	Text editor	SEGA Market item	Used for programming, preparing programs, data, etc. Can also be used by word processors and text editors available on the market (those that support TEXT format) Macro assembler for 68000 Linker for 68000 Remote debugger for 68000
	Assembler	SEGA	
	Linker Debugger	SEGA SEGA	

• Hardware Setting Diagram



## Development Overview

The Saturn sound system makes it possible to produce sounds without having a knowledge of assembly language or other computer programming, as the target board itself is constructed as a MIDI polyphonic multiple sound source with a maximum of 32 sounds.

By creating tone bank data using a dedicated tone editor configured similarly to synthesizer tone editors available on the market, sound development can be performed to compose desktop music (DTM) using a regular MIDI sequencer. It is also possible to use a DSP linker to freely link reverb, delay, chorus and other effects (to the extent that memory will allow) like those generated by digital multi-effectors on the market. Using the mixer function of a tone editor, it is possible to set the volume and type of the DSP effect together with the level or normal position of each tone in the tone bank data. Development of voice sampling can also be performed efficiently as data can be directly captured into the Macintosh from the digital input of the target board allowing waveform editing.

## General Sound Development Procedures

### 1. Tool Start Up

First start up the sound simulator. It is convenient to start up the tone editor, waveform editor, DSP linker, and sequencer at the same time, as required.

### 2. Mapping Sound Memory

When performing sound development, first start up the sound simulator, set sound memory mapping, and then transfer the 68000 sound driver program to the target board. It is convenient to create the sound area map first and make additions and changes whenever necessary.

### 3. Creating Tone Bank Data Production

Tone bank data is created by the tone editor. Also, SEGA plans to supply a tone library. Tone data banks use four types of data: mixer, voice, layer, and waveform, to make a single tone (Refer to "Tone Bank Data Configuration" on Page 7 of this manual). Voice is used for program changes using MIDI, so 1 bank can hold up to 128 voices. Therefore, using one tone bank data, up to a maximum of 128 different instrument types can be played.

Tone bank data is transferred to the sound simulator target board. At this stage, it can function as a multi-sound source for creating sound using MIDI input. DVA (last in priority) is used as the key sound-generating method. Editing of tones, tone levels, and *pan* can be performed and multiple tone bank data can be stored.

### 4. Editing Waveform

When editing a tone waveform (the basis of tone editing), start the waveform editor to sample and edit the waveform.

## 5. DSP Effect Program Link

The DSP Linker is used to link DSP effect programs. After selecting the desired effect from the DSP library, the wiring is set and the DSP program for that effect is transferred to the target board. Each parameter of the effect can be edited after transfer. The number of voices that can be generated simultaneously for modulation effects is decreased from one to four voices since a slot is used as a modulator. See DSP Linker Manual for details.

Since there are already reverb, echo, chorus and other module libraries, select and link the desired effect from these. Multiple effects can be used simultaneously in the Saturn sound system, not to exceed 128 steps overall. For example, suppose an echo was 20 steps, a chorus was 22 steps and an equalizer was 5 steps, then these three would total 47 steps.

**Note:** Steps are the number of commands in each effect.

## 6. Creating Sequence Data

Musical composition and arrangements are performed using the target board as a sound source. The target board has two MIDI IN systems, and since each has 16 channels, a maximum of 32 sequencer tracks can be accommodated. Since the VOICE number in the tone bank data can be freely selected and the tone changed in each sequence track by MIDI program changes, different tones can be set for all 32 tracks.

## 7. Converting Sequence Data

Composed tunes are ultimately converted to a MIDI standard file by the sequencer software (Performer, Vision, Cubase, etc.), and are then converted by the sound simulator to Saturn format data for use by the target board.

There are two types of sequence data: tune data produced by the MIDI sequencer, and sequence data in the Saturn format in a form that can be compressed and loaded into the sound memory. Other than being compressed, this is the same as deployed MIDI data.

**Note:** Sound effects are composed basically the same way as musical pieces. In this development system, there are no differences in the production process and parameter settings between musical compositions and sound effects.

## 8. Simulation (Actual Equipment Simulation Function)

Saturn format sequence data (Sequence Bank Data Yes) are transferred to the target board. The sound simulator simulates tunes and sound effects under actual game conditions in order to check them (discussed later). As long as there are no problems, the same sound that is generated by the target board with off-the-shelf software is reproduced. At this stage, the final evaluation of the links and balance between the tunes and sound effects is performed (described hereafter). If there are areas that must be changed, each is redone using a tone editor, waveform editor, DSP linker and sequencer software.

## 9. Loading into the Actual Game



## Tone Bank Data Composition (Tone Components)

Each tone bank data component (beginning with the smallest) is explained below.

### Waveform Data

PCM data in AIFF format.

### Layer Data

Layer is the basic tone unit which adds LFO, EG, PITCH and FM settings and other data to the waveform data. Since volume can be controlled by the MIDI velocity value, velocity switches, etc., can be realized by setting a velocity table for each layer. (One layer = one waveform.) Up to 128 layers can be set in one voice.

### Voice Data

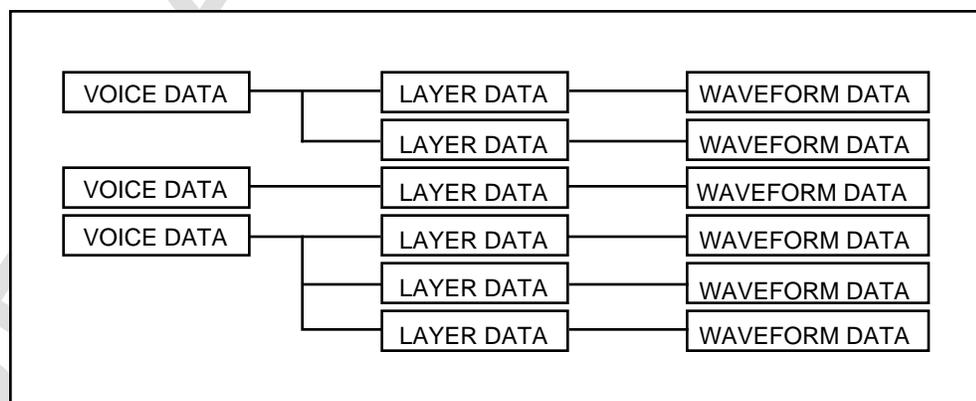
Voice data are data combining multiple layers to which key splits, the volume for each layer, the bend range, the portamento and other data are added. It is also called batch or performance by other synthesizer manufacturers. Up to 128 can be set in one bank. Voice data change according to the MIDI program changes.

Voice data are settings of how many of which layers will be used and which layers will be sounded according to changes in intervals and volume.

### Mixer Data

In addition to the above data, the amount of return from DSP effects and pan (fixed position) and other 16-channel mixing data can be set. Up to 128 can be set in one bank.

An arrangement of the above elements is one tone bank data. One tone bank data has at least one voice and can have up to 128 voices, memory permitting. In this sound system, there can be multiple tone bank data in the same map, and each can be sounded simultaneously as independent sound sources. Therefore, while a tune or sound effect is being played in one bank, it can be replaced by another bank. This makes it possible to build a flexible system that facilitates efficient development and memory utilization.



**Note:** Refer to Tone Editor Manual for more information.

## FM (Frequency Modulation)

FM has become well known through Yamaha's DX7 and other models, but the FM in this system is not limited to a sine wave as the fundamental waveform. The various AIFF format waveform data can be used as both carrier and modulator. Algorithms for combining them can also be freely configured. It is also possible to change the degree of modulation by means of the velocity.

The biggest problem with FM is that the carrier and the modulator use one slot each, thus reducing the number of sounds that can be played simultaneously. However, since various tones can be produced by FM if a base waveform (sine wave) is used as one waveform data, it is highly useful when not wanting to increase the tone bank data due to RAM restrictions.

**Note:** The memory for sound is 1024 KB on the target board, but it is 512 KB in the actual hardware. The area after subtracting sequence data, DSP programs, work RAM, etc., will be assigned to tones. Even if the sampling rate is lowered, it is impractical to load several types of PCM waveforms, comprising large amounts of data, into the memory at the same time.

## Usable MIDI Commands

Of the events contained in a standard MIDI file, those converted by the converter (launched by the sound simulator) are listed below. As long as degradation storage of the PitchBend is not set by the environment, the following can be converted with no degradation.

Note on/off	Note Off is replaced by Gate Time
Poly-Key Pressure	
Control Change	(Bank Select must exist at the top of each track)
Program Change	Must exist at the top of the track
Channel Pressure	
Pitch Wheel Change	Accommodates both 7 bit and 14 bit expressions
Meta Event	Accommodates only tempo
System Message	Exclusive or Start, Stop and Song Position are not accommodated.

If the following conditions are not satisfied, the converter outputs an error message and stops the conversion operation.

- System messages must not be included in standard MIDI files. System messages include Exclusive, Start, Stop, Song, Position Pointer, Song Select, etc.
- The number of events included in a standard MIDI file before conversion must be less than 6143 events per track. However, since Note Off is absorbed by Gate Time after conversion, the number of events becomes much smaller than this. Similarly, meta events must be less than 256, and each meta event must not exceed 127 bytes in length.
- Always convert standard MIDI files to type #1.
- Only one loop start command (No. 31 of the control changes) can exist in each track.



- Always set a bank change and program change at the top of each track (not required in blank tracks). Voice numbers and bank numbers that do not exist cannot be specified by program change or bank select.

### **MIDI Channels and Voices**

Since the Saturn sound system can handle MIDI data for up to 32 channels (tracks) simultaneously, up to 32 instruments can be handled simultaneously when one MIDI channel is used per instrument. In other words, all of the songs and sound effects must be played (sounded) within this. There is no problem when a tune or sound effect is played independently. When any combination of a tune and a sound effect is played simultaneously, each must be assigned to a different channel. In this system, 32 MIDI channels are dynamically assigned (DVA) and sounds are generated.

Therefore, each sequence data must contain this information at the top of the track. In order to play the correct tone, the bank (control change bank select) and voice (program change) must be specified for each track of sequence data.

SEGA Confidential

## 2. Sound Simulator

### Main Window

Edit window (samplemap)					
CRNT 0 No name					
No	Start	- End	Size	Data	File name
01x	10000	-1FFFF	10000	Sequence1	mureseq
02	20000	-3FFFF	20000	DPS WorkRAM1	
03x	40000	-BFFFF	80000	BANK data1	Apus area1 set.bin
04	C0000	-CFFFF	10000	Sequence2	mureseq
05	D0000	-DFFFF	10000	Sequence3	NoName
06	E0000	-E07FF	00800	DPS program1	dsp1.EXB
07	E0800	-E0FFF	00800	DPS program2	dsp2.EXB
08	E1000	-E17FF	00800	DPS program3	

### The Sound Simulator

Sound Simulator is a simulator that produces sound under the same conditions as when sound data that is created by a sound tool is loaded into a game. Normally, the play control cannot be tested until after the game program is created and the sound is added, but the Sound Simulator takes the place of the game program by simulating this control using Macintosh software. The Sound Simulator's functions are classified as follows:

- Sound system startup
- Tone data and song data transfer
- Play control
- DSP (effects program switching
- Creation, compression, and linking of Saturn format data
- Creation of sound area maps

### Sound System Start Up

As with the actual equipment, if the power to the sound board is turned off, the sound system will be erased. Therefore, when the power to the sound board is turned on, the sound system must be started up. Click on "System Startup" to perform the following processes that is performed by the sound simulator when a game is loaded.

- Hardware initialization
- System table and sound area map transfer
- Sound program transfer
- Sound driver startup



### **Tone Data and Song Data Transfer**

In preparation for play, a sound area map is used to transmit both tone and song data to the sound memory. If it is within the range of the map information area size, tone data and song data can be freely substituted, so several tone data and song data areas can be used and other song data can be substituted while a song is playing.

### **Play (Sound Generation) Control**

Controls the start, stop, pause, fade in, fade out, etc., for songs and sound effects. In addition to using the mouse, play can be controlled in real time by adjusting the above functions, and can be assigned to the Macintosh (1-8) to allow real-time play. Effect sound evaluation and combined effect sound evaluation and level matching can be performed while a song is playing.

### **DSP (Effect) Program Switching**

When there are several DSP (effect) programs on a map, a DSP program can be changed by clicking on "Effect Change" to switch to the desired DSP program. The DSP program will not run by simple being transferred, "Effect Change" must still be clicked even if there is only one DSP program.

### **Creation, Compression, and Linking of SATURN Format Data**

Song data created with the MIDI sequencer can be changed to SATURN format in a compressed state. Multiple songs can be put into one song data bank (sequence bank), and the "Make Sequence Bank" function can be used to link multiple compressed data. It is assumed that multiple song (effect sound) data is in a sequence bank, so even if only one song is in the bank, the "Make Sequence Bank" function must be used.

### **Sound Area Map Creation**

Sound is controlled by each individual game area. A memory map for each area is created in accordance with the tone and song data size, and the effects performed for that area.

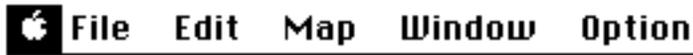
The game program looks at this map and transfers tone and song data. The sound driver also controls tone and play based on this map. Thus the sound area map is a very important memory map that is at the heart of sound development (control). One sound area map is made for each game. When this data is loaded into a game, it is passed to the game program and then is transferred to the sound system when the system is started up.

### 3. How to Use the Sound Simulator

This section explains how to use the functions for each Sound Simulator menu item.

#### Menu Bar

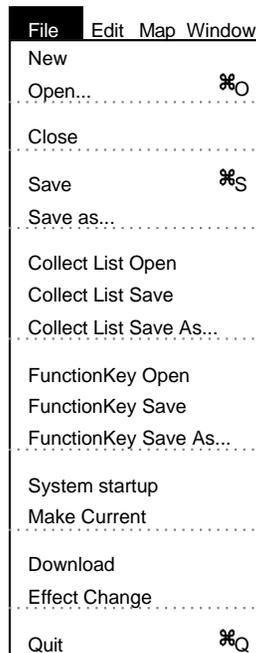
There are five menus in the Sound Simulator menu bar:



Following is an explanation of each menu

#### File Menu

When the File menu is opened, a menu like the following is displayed.



The File menu is the menu that is used to open and save map files, etc. The function of each item is given below.

#### New

Select to create a new map.

#### Open...

Select to open maps that have already been created.

#### Close

Select to close a map that is being edited. The simulator does not quit.

#### Save

Select to save the map being edited.

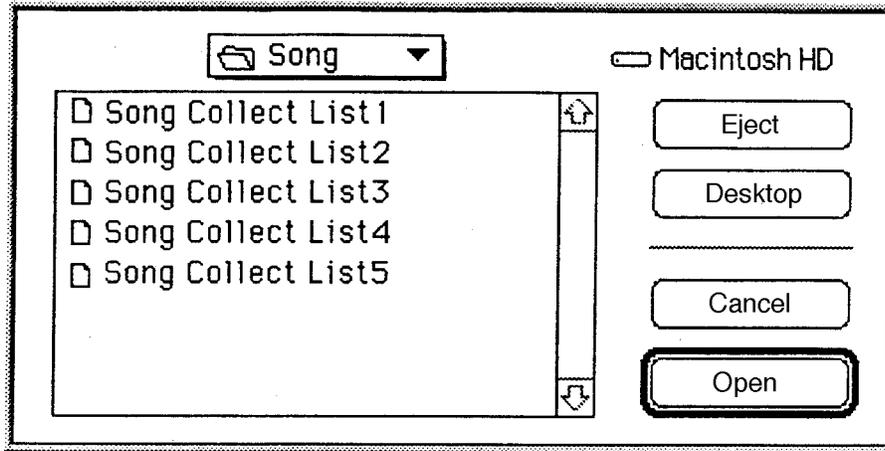
#### Save as...

Select to save the map being edited under a different name.



**Collect List Open**

Opens the collect list file. Selecting Collect List Open opens the following dialog box.



The existing collect list files are displayed. A collect list is a compilation of multiple file names, so it eliminates the need to select a file each time. It is used when converting standard MIDI files and when creating sequence banks. Files that are opened here contain both of the above kinds of collect lists.

**Collect List Save**

Saves the currently active collect list file.

**Collect List SaveAs**

Saves the currently active collect list file under a different name.

**FunctionKey Open**

Calls the saved function key setup settings.

**FunctionKey Save**

Saves all the set function key setup settings.

**FunctionKey SaveAs...**

Saves the set function key setup settings under a different name.

**System Startup**

When System Startup is selected data, such as when the sound driver is transferred to the target board and then the sound system on the target board is launched.

**Make Current**

When Make Current is selected, switch so that the map currently selected becomes active. The sound driver then runs this map as the current map (currently valid map). At this time, if there is an autoloader designation in the map file information, file data is automatically transferred.

**Download**

Transfers the currently selected file data to the target. Clicking on the file to be transferred causes the file to be displayed in black reverse type to show that it has been selected. To select several files at the same time, hold down the Shift key while clicking on the files to be selected. To deselect a selected file, click on the file again.

**Effect Change**

Switches the DSP programs when there are multiple DSP programs in the current map. Clicking on the DSP program to be switched causes it to be displayed in black reverse type to show that it has been selected. Multiple files cannot be selected at the same time.

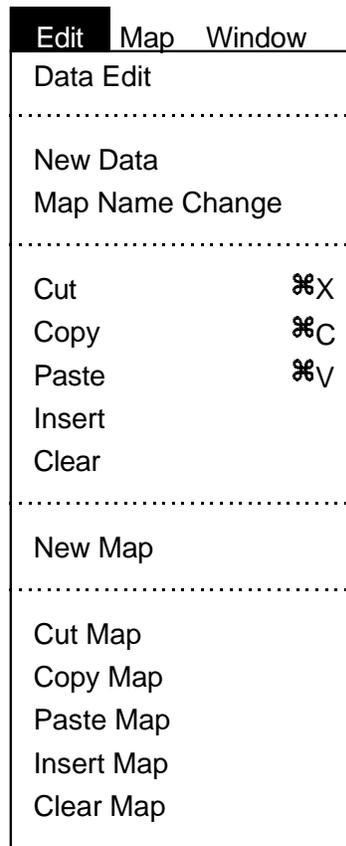
**Quit**

Quits the Sound Simulator and closes the window.



## Edit Menu

Opening the Edit menu displays a menu like the following.



## Data Edit

Selecting “Data Edit” opens the following dialog box.

The image shows a dialog box with the following fields and controls:

- Start: 10000
- Size: 10000
- Type: Bank data
- Load File: mureseq
- File size: 0263E
- Auto Loader:
- Cancel button
- OK button

Here, the map information data is changed. The start address, area size, data type, transfer file, and autoload specifications can be set. Specifying the transfer file causes the specified file size to be displayed in bytes to be used as a guide for determining and changing the area size.

### **New Data**

Selecting new adds new data to the current map. The data type can be the default "BANK data" or the data type selected last in the Data Edit dialog box. Size is 00000.

### **Map Name Change**

Each area map for each area can be assigned a name of its own. This makes it easier to recognize names in locations with many areas.

### **Cut**

When selected, a confirmation menu appears prompting whether to update the address after cutting, or do nothing. The cut data is stored and can be retrieved by performing an insert operation.

### **Copy**

When selected, the currently selected data is stored. It can be retrieved by performing an insert operation.

### **Paste**

When selected, the currently stored data is pasted.

### **Insert**

When selected, the currently stored data is inserted. The insertion position is just before the currently selected data, which is displayed in reverse. When data has not been selected, the insertion is added at the very end.

### **Clear**

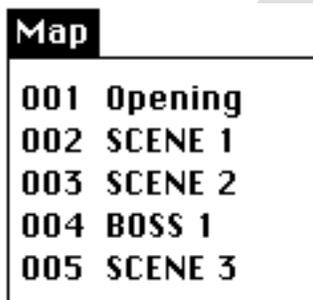
When selected, all of the currently selected data, except the data type, is cleared.

### **New Map, Cut Map, Copy Map, Paste Map, Insert Map and Clear Map**

These functions perform the same editing described above, but for maps instead of data.

### **Map Menu**

Opening the Map menu displays a window like the following.

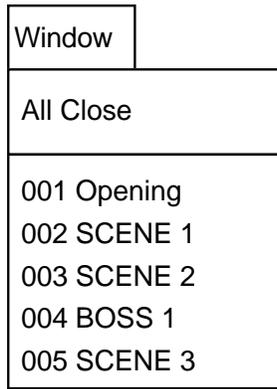


Here, the area map to be switched is selected. The area maps switched are those that will be edited, so this operation does not switch the current map. The difference between the Map menu and the Window menu is that the latter is only displayed to reference other map windows and cannot be used to select a map. A maximum of 128 individual map data can be registered by the Sound Simulator.



## Window Menu

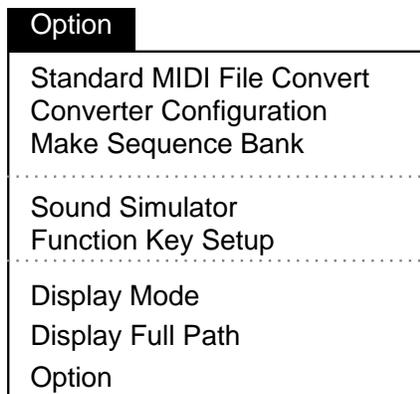
Opening the Window menu displays a window like the following.



Use this menu to refer to other map information.

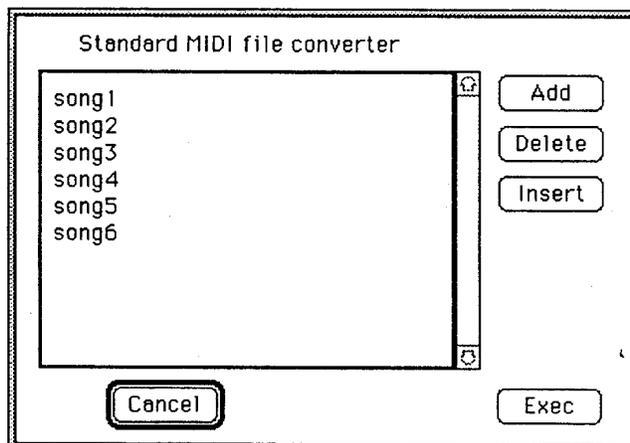
## Option Menu

Opening the Option menu displays a menu like the following.



## Standard MIDI File Convert

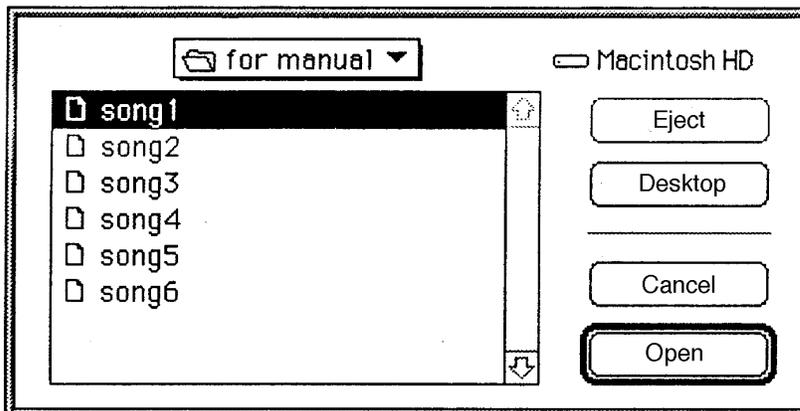
Selecting "Standard MIDI File Convert" from the Option menu will open the following dialog box.



Displayed here is a list of the songs selected by Add called a Collect List. Nothing is displayed until additions and insertions are made using Add and Insert. Opening the Collect List with "Collect List Open" allows the list to be read without having to register an addition or insertion each time.

### **Add**

Selecting "Add" opens the following dialog box.



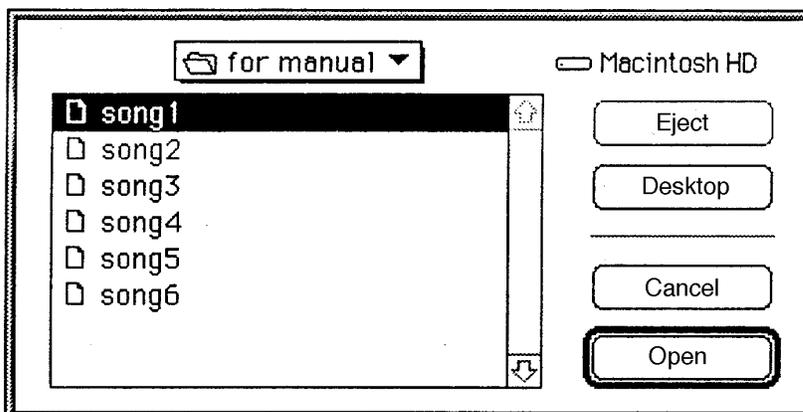
Here, song names are added to the Collect List. Selecting the song name to be added adds that song to the end of the list.

### **Delete**

Select a song name from the Collect List and click on "Delete" to remove that song name from the list.

### **Insert**

Selecting "Insert" opens the following dialog box.



Here, song names are inserted into the Collect List. Click on the place in the Collect List where to insert the song name and then select the song name. The song name will then be inserted directly above the song name in the Collect List that was selected.

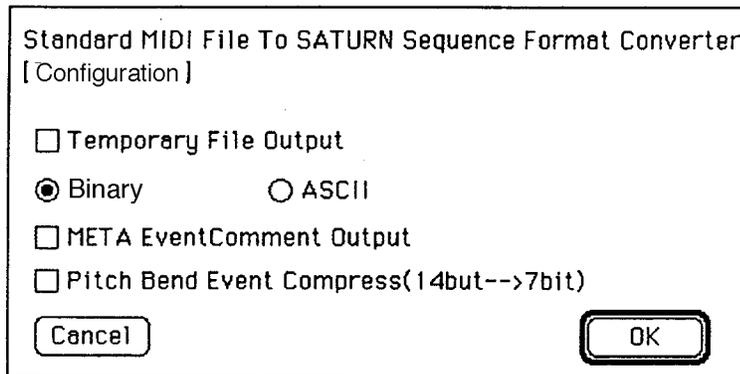


## **Exec**

Specifying the song name (standard MIDI file) to be converted and clicking on "Exec" will start up the converter, and a SATURN compressed format file will be output. The output file format and compression mode, etc., can be specified by selecting "Converter Configuration" from the Option menu. The final output file will be output with the specified file name plus the extension ".CNV." Also, temporary files will be output with the specified file name plus the extension ".TMP."

## **Converter Configuration**

Selecting "Converter Configuration" from the Option menu displays the following dialog box.



The meaning of each check box is shown below.

### **Temporary File Output**

Sets whether or not the temporary file before repeat detection is also output.

### **Binary      ASCII**

Allows user to select whether the final file will be output in ASCII format or binary format. If it is output in ASCII format, MIDI note information and control change, etc., can be easily checked.

### **META Event Comment Output**

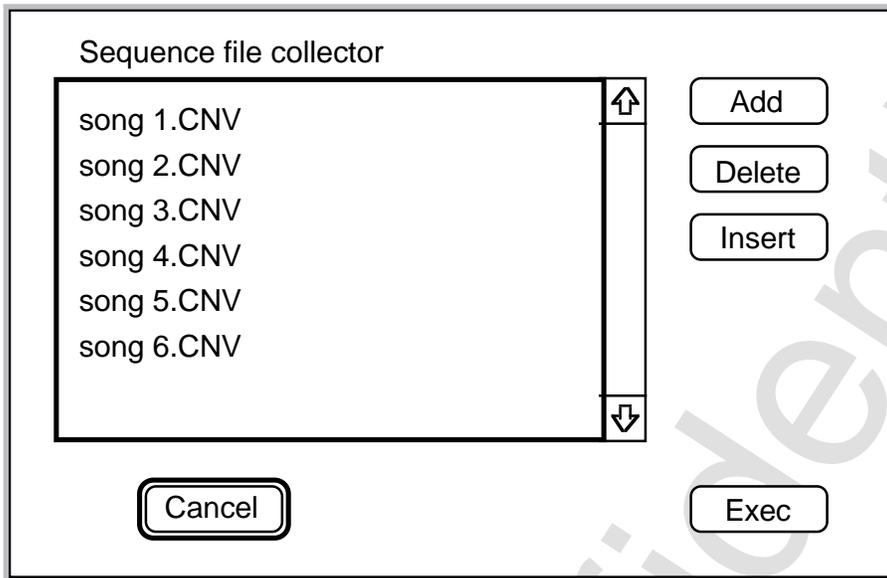
Sets event portions outside of the conversion target as comments whether or not to output to an ASCII file. Only valid when ASCII format is specified.

### **Pitch Bend Event Compress**

Sets whether PitchBend is output in the original 14-bit accuracy, or if the accuracy is reduced and compressed to 7 bits.

### Make Sequence Bank

Selecting "Make Sequence Bank" from the Option menu opens the following dialog box.

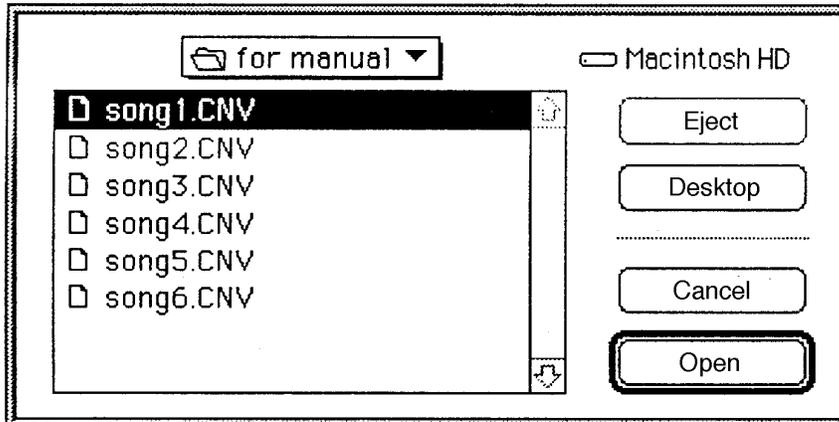


Displayed here is a list of compressed file names converted by Standard MIDI File Converter called a Collect List. Nothing is displayed until additions and inserts are made using Add and Insert.

"Collect List Open" allows the Collect List to be read without having to register an addition or insertion each time.

### Add

Selecting "Add" opens the following dialog box.



Here, compressed file names are added to the Collect List. Selecting a compressed file name to be added will add the compressed file name to the end of the list.

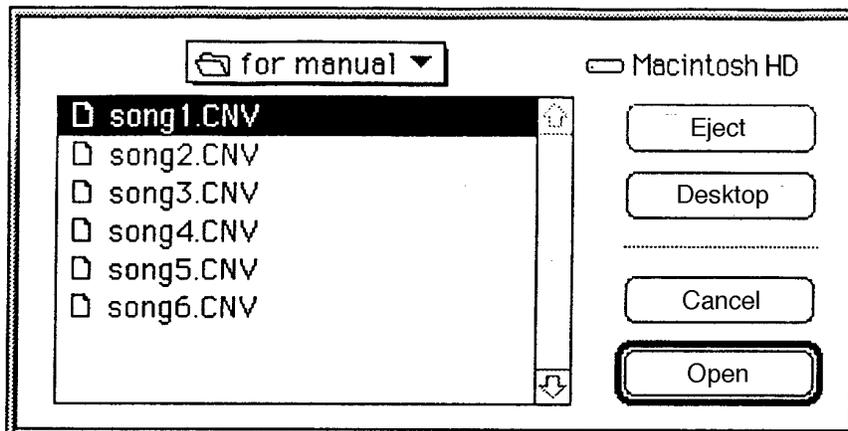
### Delete

Select a compressed file name from the Collect List and then click on "Delete" to delete it from the Collect List.



### Insert

Selecting insert opens the following dialog box.



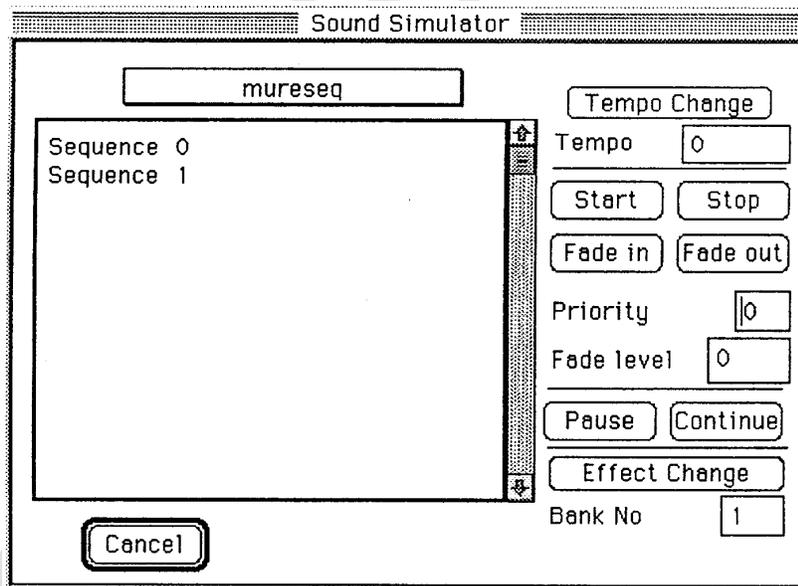
Here, the compressed file name is inserted in the Collect List. Click on the location in the Collect List to insert the file and then select the compressed file name. The compressed file name will be added to the Collect List just above the name of the file that was selected.

### Exec

Links all the compressed files specified in the Collect List and outputs them as a one-sequence databank. The names of the output files can be freely specified.

### Sound Simulator

Selecting "Sound Simulator" displays the following window.



If the sequence bank has been transferred to the currently active map, the sequence bank can be selected from the pull-down menu in the file list box. Selecting the sequence bank displays a list of the sequence names in the bank like that shown above. Here, select the sequence to play and use the buttons described below to control the play.

Tempo Change: Changes the play tempo.  
 Tempo: Specifies the value of the tempo to be changed. Plus values mean faster, minus values mean slower.  
 -32786—>+32767 is X2 (1/2 when minus) at 4096.  
 Start: Starts play.  
 Stop: Stops play.  
 Fade in: Generates sound while the sequence fades in.  
 Fade out: Generates sound while the sequence fades out.  
 Pause: Temporarily stops play.  
 Continue: Restarts play after pause.  
 Priority: Specifies sequence play priority. In the range 0—>127, 0 has the highest priority.  
 Fade Level: Specifies the Fade Level of Fade In and Fade Out. In the range of 0—>255, 255 changes the slowest.  
 Effect Change: Switches the DSP program.  
 Bank No.: This is the number of the DSP program bank that contains the DSP program to be switched, and it can be specified from 0 to 15.  
 Cancel: Quits Sound Simulator and closes the window.

**Note:** Please refer to the chapter on System Interface for a detailed description of each control command.

### Function Key Setup

Selecting Function Key Setup displays the following window. All functions of Sound Simulator's Start, Stop, Pause, Continue, Fade In, and Fade Out can be assigned using keys 1~8.

Key	PlayNo	Bank	Sequence	Function	Priority	Fade Level
1	0			Non		
2	0			Non		
3	0			Non		
4	0			Non		
5	0			Non		
6	0			Non		
7	0			Non		
8	0			Non		

(0-7)  (0-15)  (0-127)  (0-7)  (0-255)

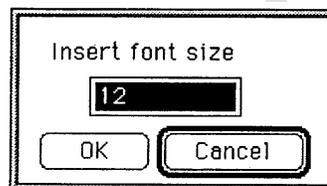
Here, the sequence data is controlled by a bank number (0-15) and sequence number (0-127) in the currently active map. Select by clicking the key to be specified and input the data using the input boxes at the bottom. A maximum of 8 sequences can be played at one time, and a play (sound generation) control number is specified when play (sound generation) starts (Start/Fade In). After that, Stop, Pause, Continue, Fade In, and Fade Out can only be controlled with a play control number. The bank sequence is not needed (it is ignored).



Play No.:	Selects the play control number from 0-7.
Bank:	Selects the sequence bank number from 0-15.
Sequence:	Selects the sequence data number from 0-127.
Function:	Selects the functions to be assigned to the key from the pull-down menu. Start, Stop, Pause, Continue, Fade In, Fade Out
Priority:	Selects the sequence play priority from 0-127. 0 has the highest priority.
Fade Level:	Selects the Fade Level for Fade In and Fade Out from 0-255. 255 changes the slowest.
New:	Up to 32 pages can be recorded for 1-8 key assignment. New assignment pages can also be added. Add the name of each page by inputting it in the input box between the lower left < and >.
<—/—>:	Used to selected the key assignment pages.
Use:	Select a page using <— —> and then confirm the active key assignment using Use.
Set:	Assigns the data set using the bottom input box to a key during selected.
Exit:	Saves all of the setting values set in this window and then closes the window.

### Display Mode

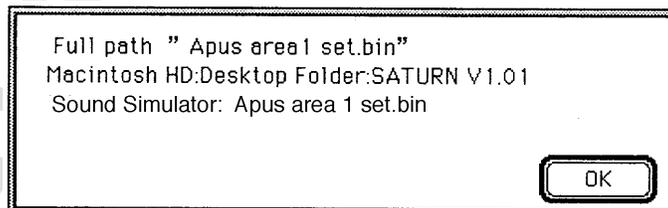
Selecting "Display mode" opens the following dialog box.



Specifies the font size of display characters. Choose a large font size when the screen is small and the characters are hard to read.

### Display FullPath

Selecting "Display FullPath" opens the following dialog box.



When data or a folder has been moved by dragging or otherwise, it is possible that specified files will not be in their previously recorded (registered) folder position, and file data cannot be read. After selecting the file name by clicking on it, select "Display FullPath" to display the full path of the folder in which the file is registered.

## 4. Sound Driver System Interface

This chapter explains how the game program controls the sound system, so those not concerned with this area need not read this section.

### System Area Contents

The system area is a fixed area that operates the sound driver provided by the SEGA sound memory, and cannot be used for any other purpose. This area's mapping also cannot be changed, and when developing sound there is no need to know the contents of this area. 44 KB in the 4 Mb of sound memory can be used from the first (starting) number (00000h) to (B000h), and the contents are as described below.

#### 68K Vector Table

This is vector table for program interrupt processing by the sound CPU (68K). The size is fixed at 400H from 0000h and cannot be changed.

#### System InterfaceArea

This is fixed area for interfacing among various systems, such as sound drivers, tone development systems, the host system for game assembly, etc., and this system. It extends from 0400H to 400H and cannot be changed.

#### 68K ProgramArea

This is the program area for the sound CPU and is used to store and execute all programs related to sound. The top address and size of this area are stored in the system information table of the system interface area.

#### 68K Work Area

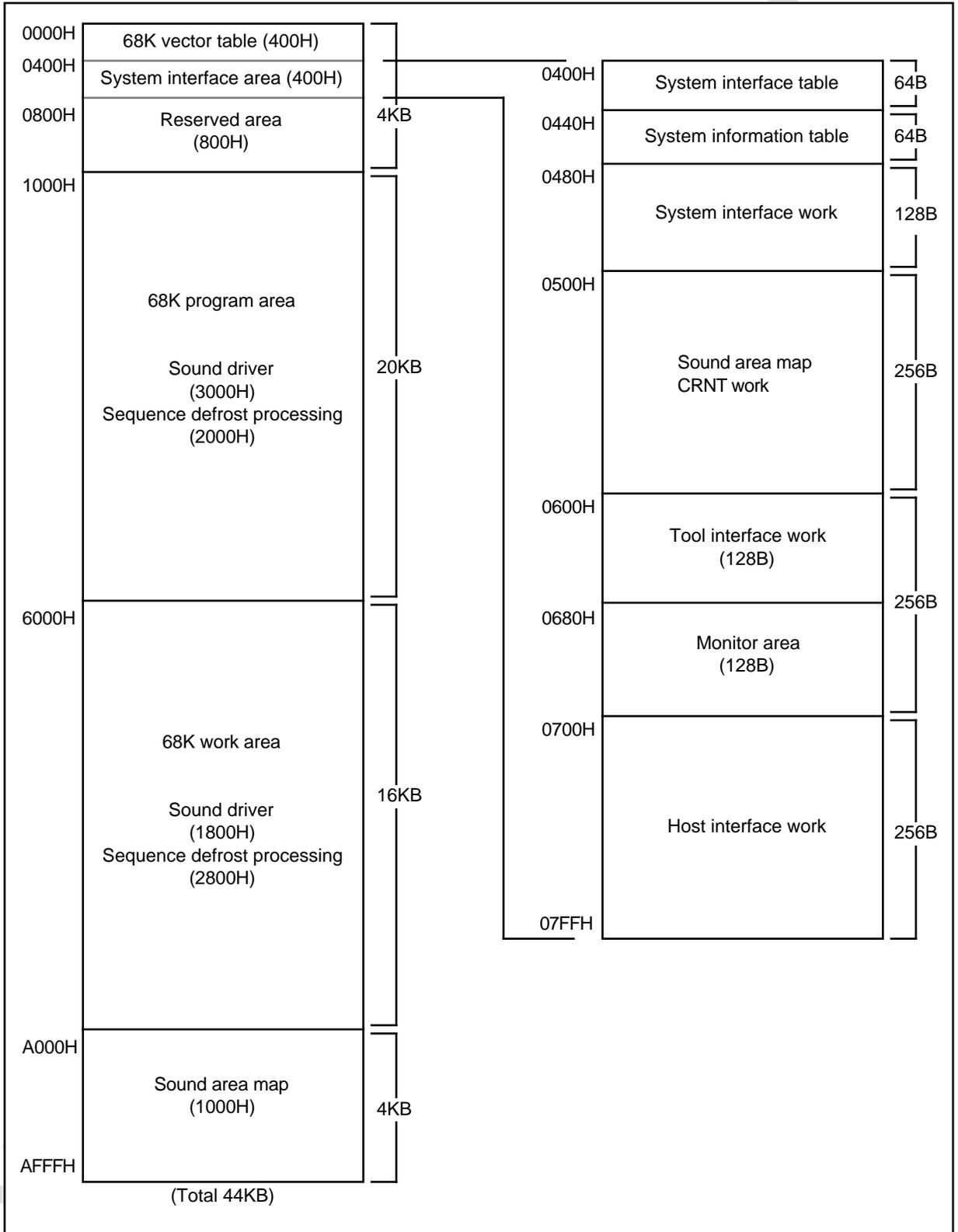
This is the program work area for the sound CPU and is used as a work area by all sound-related programs. The top address and size of this area are stored in the system information table of the system interface area.

#### Sound Area Map

The sound area map is stored here. Up to 128 area maps can be held in one sound area map (one area map can hold up to 32 map data). Using a sound simulator, one sound area map can be made for one game. Since this area is only for storing the entire sound area map, the map data of the currently selected area references sound area map CRNT work of the system interface area. The top address and size of this area are stored in the system information table in the system interface area.



## System Area Memory Map



## System Interface Area

In the Saturn sound system, a system interface area provided in the fixed area of the sound memory is used to exchange information between sound drivers, tone development systems, the host system for game assembly and other systems. It comprises an information table that stores system information and a work area for exchanging information.

### System Interface Table (400H–43FH: 64B)

This is an information table for controlling the interface between each of the systems during sound development or game assembly, and it is stored at a fixed address in the sound memory. A work area for sound development and a system interface in actual equipment is included.

Address	Offset	Size	Area	Contents
0400	+00	4B	System information table pointer	Top address of system information table (0440h)
0404	+04	4B	Host interface work pointer	Top address of host interface work (0700h)
0408	+08	4B	Sound area map CRNT work pointer	Top address of sound area map CRNT work (0500h)
040C	+0C	4B	Tool interface work pointer	Top address of sound tool interface work (0600h)
0410	+10	1B	DSP Program load flag	BOOT ROM program work
0411	+11	1B	Sound Driver load flag	Sound Driver program work
0412	+12	4B	System Interface Work pointer	System interface work area start address (0480h)
0416–043F	–	42B	–	Reserved area

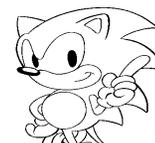
The numerical values in parentheses show changes as a result of version upgrades.

### System Information Table (440H–47FH: 64B)

This is an information table in which system information of the sound system is stored at a fixed address in the sound memory.

Address	Offset	Size	Information Data	Contents
0440	+00	4B	68K program start address	Top address of 68K program area (1000h)
0444	+04	4B	68K program size	Size of 68K program area (5000h)
0448	+08	4B	Sound area map start address	Top address of sound area map area (A000h)
044C	+0C	4B	Sound area map size	Size of sound area map area (1000h)
0450	+10	4B	68K work start address	Top address of 68K work area (6000h)
0454	+14	4B	68K work size	Size of 68K work area (4000h)
0458 - 047F	–	40B	–	Reserved area

The numerical values in parentheses show changes as a result of version upgrades.



**System Interface Work (480H-4FFH: 128B)**

This work area is used for exchanging information between systems.

pointer+xx	Size	Information Data	Contents
+00	1B	PCM ch 1 stream play slot	PCM stream play ch 1 slot number
+01	1B	PCM ch 2 stream play slot	PCM stream play ch 2 slot number
+02	1B	PCM ch 3 stream play slot	PCM stream play ch 3 slot number
+03	1B	PCM ch 4 stream play slot	PCM stream play ch 4 slot number
+04	1B	PCM ch 5 stream play slot	PCM stream play ch 5 slot number
+05	1B	PCM ch 6 stream play slot	PCM stream play ch 6 slot number
+06	1B	PCM ch 7 stream play slot	PCM stream play ch 7 slot number
+07	1B	PCM ch 8 stream play slot	PCM stream play ch 8 slot number
-	120B	-	Reserved Area



### Tool Interface Work (600H–6FFH: 256B)

Work that stores RAM area information used by the waveform editor, tone editor and DSP linker. The second-half 128 bytes are a monitor data area for monitoring the play status.

pointer + xx	Size	Data	Contents
+ 00 (hex)	2B	–	Reserved
+ 02	2B	–	Reserved
+ 04	4B	Area Start Address	Start address of waveform editor RAM area
+ 08	4B	Area total size	Total size of waveform editor RAM area
+ 0C	2B	–	Reserved
+ 0E	4B	Area Start Address	Start address of tone editor RAM area
+ 12	4B	Area total size	Total size of tone editor RAM area
+ 16	2B	–	Reserved
+ 18	4B	TrgtMem_DSPprogAddress	DSP linker dedicated area
+ 1C	4B	TrgtMem_DSPprogSize	DSP linker dedicated area
+ 20	32B	TrgtMem_Filename	DSP linker dedicated area
+ 40	4B	TrgtMem_DSPRAMSize	DSP linker dedicated area
+ 44	2B	TrgtMem_RBL	DSP linker dedicated area
+46	4B	TrgtMem_ModElementAddress	DSP linker dedicated area
+4A	4B	TrgtMem_ModeElement Size	DSP linker dedicated area
+4E	1B	TrgtMem_NumberOfElements	DSP linker dedicated area
+ 4F - 7F	49B	–	Reserved
+ 80 - 83	4B	Voice 1 monitor	Bits 24-31 (1st byte): Program (voice) number 0-127 Bits 16-23 (2nd byte): MIDI note number 0-127 Bits 08-15 (3rd byte): MIDI velocity 0-127 Bits 00-07 (4th byte): Reserved
		:	
		:	
		:	
+ FC - FF	4B	Voice 32 monitor	Bits 24-31 (1st byte): Program (voice) number 0-127 Bits 16-23 (2nd byte): MIDI note number 0-127 Bits 08-15 (3rd byte): MIDI velocity 0-127 Bits 00-07 (4th byte): Reserved

### Host Interface Work (700H-7FFH: 256B)

This is a work area for communicating between the host system and the sound system. Commands are received via this area from the host system, and it returns status/timing flags, etc. Basic control is performed in the areas listed below. Since specific commands are required depending on the project (game), unused area is allotted as required. Modes, status, etc., can similarly be added or changed as required.

pointer + xx	Size	Interface Data	Contents
+ 00 (hex)	16B	Command block 1	Command block 1 from host to sound (Host -----> Sound)
+ 10	16B	Command block 2	Command block 2 from host to sound (Host -----> Sound)
+ 20	16B	Command block 3	Command block 3 from host to sound (Host -----> Sound)
+ 30	16B	Command block 4	Command block 4 from host to sound (Host -----> Sound)
+ 40	16B	Command block 5	Command block 5 from host to sound (Host -----> Sound)
+ 50	16B	Command block 6	Command block 6 from host to sound (Host -----> Sound)
+ 60	16B	Command block 7	Command block 7 from host to sound (Host -----> Sound)
+ 70	16B	Command block 8	Command block 8 from host to sound (Host -----> Sound)
+ 80	2B	Song 1 mode/status	Play sequence 1 mode/status (Sound -----> Host)
+ 82	2B	Song 2 mode/status	Play sequence 2 mode/status (Sound -----> Host)
+ 84	2B	Song 3 mode/status	Play sequence 3 mode/status (Sound -----> Host)
+ 86	2B	Song 4 mode/status	Play sequence 4 mode/status (Sound -----> Host)
+ 88	2B	Song 5 mode/status	Play sequence 5 mode/status (Sound -----> Host)
+ 8A	2B	Song 6 mode/status	Play sequence 6 mode/status (Sound -----> Host)
+ 8C	2B	Song 7 mode/status	Play sequence 7 mode/status (Sound -----> Host)
+ 8E	2B	Song 8 mode/status	Play sequence 8 mode/status (Sound -----> Host)
+ 90	1B	Total Volume L	Total volume data L (0-255) (Sound -----> Host)
+91	1B	Total Volume R	Total volume data R (0-255) (Sound -----> Host)
+92	1B	H-vol L	High-Pitch Range Volume data L (0-255) (Sound -----> Host)
+93	1B	H-vol R	High-Pitch Range Volume data R (0-255) (Sound -----> Host)
+94	1B	M-vol L	Medium-Pitch Range Volume data L (0-255) (Sound -----> Host)
+95	1B	M-vol R	Medium-Pitch Range Volume data R (0-255) (Sound -----> Host)
+96	1B	L-vol L	Low-Pitch Range Volume data L (0-255) (Sound -----> Host)
+97	1B	L-vol R	Low-Pitch Range Volume data R (0-255) (Sound -----> Host)
+98	1B	PCM ch 0 adrs	PCM ch 0 play address (00h-0Fh) (Sound -----> Host)
+99	1B	PCM ch 1 adrs	PCM ch 1 play address (00h-0Fh) (Sound -----> Host)
+9A	1B	PCM ch 2 adrs	PCM ch 2 play address (00h-0Fh) (Sound -----> Host)
+9B	1B	PCM ch 3 adrs	PCM ch 3 play address (00h-0Fh) (Sound -----> Host)
+9C	1B	PCM ch 4 adrs	PCM ch 4 play address (00h-0Fh) (Sound -----> Host)
+9D	1B	PCM ch 5 adrs	PCM ch 5 play address (00h-0Fh) (Sound -----> Host)
+9E	1B	PCM ch 6 adrs	PCM ch 6 play address (00h-0Fh) (Sound -----> Host)
+9F	1B	PCM ch 7 adrs	PCM ch 7 play address (00h-0Fh) (Sound -----> Host)
+A0-FF	96B	Reserved	-



### Command Block

For host to sound commands, 8 16-byte command blocks are provided. As a result, a maximum of 8 commands can be issued at the same time. Due to parameter word access and other transmission efficiency reasons, the offset's +1th is not currently used. Each parameter is one byte.

+0	Command Code
+1	(Reserved)
+2	Parameter 1 (P1)
+3	Parameter 2 (P2)
+4	Parameter 3 (P3)
+5	Parameter 4 (P4)
+6	Parameter 5 (P5)
+7	Parameter 6 (P6)
+8	Parameter 7 (P7)
+9	Parameter 8 (P8)
+10	Parameter 9 (P9)
+11	Parameter 10 (P10)
+12	Parameter 11 (P11)
+13	Parameter 12 (P12)
+14	Parameter 13 (P13)
+15	Parameter 14 (P14)

**Note:** The sound memory access from the host can be 1 byte (byte), 2 bytes (word), and 4 bytes (long word), but because of the hardware, the processing time for byte access and word access is the same. Also, one long word access is slightly faster than two word accesses.

### Play Sequence Mode/Status

+0	Song play mode
+1	Status

#### [song play mode]

00: Stop  
01: Play  
02: Fade in  
03: Fade out  
04: Play pause  
05: Fade in pause  
06: Fade out pause

#### [status]

00: normal  
01-7F: error code  
80-FF: timing flag

### PCM Play Address

This is the number of samples from the start of the area at the current playing data position. When playing at 8 bits, it refers to the number of bytes, and when playing at 16 bits, refers to the number of words. The number of samples is 16 bit data, but only the first 4 bits are stored here. Therefore, the monitoring can only be done in 4K sample units.

**Note:** Use caution, since the sound CPU cannot operate while the host system is accessing the sound memory. Keep reading and writing of the sound memory by the host system to a minimum and do not allow continuous access over long periods.

SEGA Confidential



## Sound Control Commands

The table below describes the commands and parameters issued to the sound system from the host system. Parameter length can vary depending on the command. Basic control is performed using these commands, but since specific commands become necessary depending on the project (game), unused command codes are allotted as required. Commands and parameters can similarly be added or changed as required.

Command Name	Command Data	Parameter Data
Reserved	00 (hex)	Nothing
Sequence Start	01	P1 0-7: sound control number P2 0-15: sequence bank number P3 0-127: sequence song number P4 0-255: Fade level (setting not required) P5 0-127 priority level P6 bit7: 0: Play disable 1: Play enable
Sequence Stop	02	P1 0-7: sound control number
Sequence Pause	03	P1 0-7: sound control number
Sequence Continue	04	P1 0-7: sound control number
Fade In	05	P1 0-7: sound control number P2 0-15: sequence bank number P3 0-127: sequence song number P4 0-255: Fade level (0 is fastest) P5 0-127: Priority level P6 bit7: 0: Play disable 1: Play enable
Fade Out	06	P1 0-7: sound control number P2 -0-255: Maximum speed at Fade level 0.
Tempo Change	07	P1 0-7: sound control number P2 -: dummy P3-P4 +32767-->-32768: The relative tempo value for the standard tempo (0000h) is 2x when 1000h(4096) and 1/2 when minus.
Map Change	08	P1 0-255: area number of sound area map that can be changed
MIDI Direct Control	09	P1 00h or 10h: 00h-MIDI port#0 10h-MIDI port#1 P2 80h-EFh: MIDI command P3 00h-7Fh: MIDI data 1 P4 00h-7Fh: MIDI data 2
Volume analyze start	0A	nothing
Volume analyze stop	0B	nothing
CD-DA Level	80	P1 00h-E0h: CD-DA level left 00h(off),20h,40h,60h,80h,A0h,C0h,E0h(MAX)'s 8 steps P2 00h-E0h: CD-DA level right 00h(off),20h,40h,60h,80h,A0h,C0h,E0h(MAX)'s 8 steps
CD-DA Pan	81	P1 0-31: CD-DA pan left; 32 steps P2 0-31: CD-DA pan right; 32 steps



The meaning of each parameter is explained below.

**Sound Generation Control Number**

Up to eight sequences can be controlled at the same time when these sequences are played, and are specified by sound control numbers 0-7 when the sequences are started. Subsequent stop, pause, continue and other commands are executed according to these sound control numbers.

**Sequence Bank Number**

When multiple sequence data banks where sequence data is stored are mapped in the currently active map, this number specifies the number of the sequence data bank in the active map. Since a maximum of 16 of the same banks can be held in one map, this value will be any number from 0 to 15. If there is only one sequence data bank, this number is always 0.

**Effect Bank Number**

When multiple DSP program banks that store DSP microprograms are mapped in the currently active map, this number specifies the number of each DSP program bank in the active map. Since up to 16 banks can be held in one map, the value is any number from 0 to 15. When there is only one sequence data bank, this number is always "0".

**Sequence Song Number**

Multiple sequence data can be stored in sequence data banks. Specifies the number of the sequence data in the sequence data bank when there are multiple numbers of sequence data stored. Up to 256 sequences can be stored, area size permitting.

**Mixer Number**

Sets the mixer number to be switched when multiple mixers are kept in a tone bank. Since a maximum of 128 mixer data can be kept in one tone bank, use settings 0-127.

**Effect Select**

This is the effect input channel for when a signal is input to the DSP. Since 16 channels are used for DSP input, use settings 0-15.

**Priority Level**

Specifies the priority level in 128 steps when sequences are played. The highest priority is "0", and the lowest is "7". Sequences of the same priority level are all played simultaneously, but sequences with lower priority are not played. A sequence waiting to be played starts to generate sound after the sound generation of the sequence with a higher priority is complete. The waiting sequence now has highest priority.

### Play Disable and Enable

Specifies whether play will be stopped by sound priority control or whether play will continue when a sequence that could not be played at first is changed to a status where play is possible. Continuous play is done using play enable (bit 7=1).

### Fade Level

During Fade In, the time from when playing begins until maximum volume is specified in 256 steps (0-255). At "0" (the highest speed), the maximum volume is reached at the same time playing begins. During Fade Out, the time from when playing stops until minimum volume is specified in 256 steps (0-255). At "0", the minimum volume is reached when playing stops.

### chA/chB Channel Numbers

Since PCM stream play can play up to a maximum of 8 channels, channel numbers 0-7 should be used. During monoaural, make sure the chA and chB channel numbers are set to the same number. When different channel numbers are set, stereo play can be processed. During stereo play, prepare two areas (A/B) in the PCM stream buffer and divide the Pan into left and right. The Pan has 32 steps and any fixed position (pan) for which the position can be set can also be changed in accordance with time.

**Note:** Since a load equivalent to the data transfer is imposed on PCM stream play by play rate, there are some cases where all 8 sounds cannot play. A rule of thumb is given in the item "Load During Data Transmission," so please refer to this item.

#### Pan (32 Step) Details

	00h	01h	02h	--->	0Fh	10h	11h	12h	--->	1Fh
Left	Max	---	---	--->	Off	Max	Max	Max	Max	Max
Right	Max	Max	Max	Max	Max	Max	---	---	--->	Off



## Host Interface

Since the sound system runs in RAM, all data, including the sound system program, are lost when the power is turned off. Therefore, the sound system must be restarted each time the power is turned on. When developing tones, the sound simulator, not the host system, initializes the sound system. When installing games, however, the host-side system must do all initialization. Also, the sound CPU is reset when the system power is turned on and cannot operate until the host system releases this reset. Therefore, the sound system should be started as described below.

### Starting the Sound System

1. Initialize the SCSP registers so that the SCSP can access.
2. The system area (from 0000h to B000h) is initialized with all zero clear.
3. Transfer the 128 byte information table part of the system interface cable to a fixed area (400h-47Fh).
4. Transfer the sound program to the sound memory. Reference the system information table (page 26) for the transfer destination and size.
5. Transfer the sound area map to the sound memory. Reference the system information table (page 26) for the transfer destination and size.
6. Set the sound CPU reset vector for the sound CPU. Reference the system information table (page 26) for the address. SSP does not need to be set.
7. Cancel reset of the sound CPU.

The above procedures will start the sound CPU and run the program from the reset vector in the sound memory. The SCSP registers, canceling reset and other hardware subjects are described in greater detail in the "SCSP User's Manual".

### Transfer File Name

Information Table:	SYSTBL.TSK (System interface table information table part 128 bytes)
Sound Program:	SDDRV.TSK (Sound driver main part)
Sound Area Map:	Determined separately for each game.
Tone Data Bank:	Determined separately for each game.
Sequence Data:	Determined separately for each game.
DSP Program:	Determined separately for each game.

### **Preparing for Play**

The sound controller is prepared by starting the sound system, and preparation for play is performed next. Because transfer of tune data and sound effect data (referred to in combination as sequence data) to the sound memory and playing of tunes and sound effects are all performed according to the sound area map, the initial area is first specified. Whenever there is a subsequent area change, the map change command must be implemented for the area to be changed.

1. Issue the Map Change command for the initial area.
2. Transfer the tone data to the sound memory. Reference sound area map (page 28) CRNT work (page 28) for destination and size.
3. Transfer sequence data to the sound memory. Reference sound area map CRNT work for destination and size.
4. Transfer the DSP program to the sound memory. Reference sound area map CRNT work (page 28) for destination and size.
5. Perform sequence data control such as start and stop.

### **Hardware Interface**

Two communication methods are used for communications between the host system and the sound system. One, which uses the sound memory, is command handshake which utilizes the ability of the host system to access (R/W) the sound memory. The other method communicates using interrupts, wherein the host is connected to the sound system by two signals and the sound system is connected to the host system by one signal, for a total of three interrupt signals. In this version, which allows selection of the optimum system depending on the game and system content or system, sound memory is used.

### **Software Interface**

The host system and the sound system are connected via SCU, but because memory access is one way (R/W) from the host, the communication protocol is implemented by a handshake that uses the host interface work area. The host system confirms that the command data area of the host interface work area is 00H and writes the command and necessary parameters, and the sound system initializes the command data area at 00H when it receives the command. Since the sound system continually updates the song mode/song status area regarding the play status, etc., of each sequence, have the host system reference (read) this area.

The CD-BIOS performs all control for the CD-DA and sends the sounds to the sound system. The sound system can change the volume and the left/right pan of the sound and apply effects. In order to apply effects to the CD-DA, a special DSP program is required. Normally, effects are altered through effect changes in the sequence data, but effect changes should be performed from the host if necessary.



## PCM Stream Play

### Overview of Play Method

1. Transmit the PCM data to the PCM play area (PCM stream buffer).
2. Issue [PCM start: 85h] using the host interface sound control command.
3. After that, write the PCM data to the PCM stream buffer play end position at specified times.
4. When the play area ends, the sound chip repeats play from the start of the area.
5. Repeating this operation allows continuous play of PCM data.
6. When PCM play is finished, issue [PCM stop: 86h] using the host interface sound control command.

Example of playing PCM in stereo 2ch using the PCM stream buffer:  
(The buffer is logically divided in two.)

chA buffer 1

### Buffer Allocation Method

Since sound memory allocation (sound area map) is created before the game is created, this sound area map will have a PCM stream play buffer if there is a situation to perform PCM stream play. Mapping cannot be specifically used for PCM stream play, but can be used to allocate an extra area for tone and sequence data storage. The size can be freely set. Select the optimum size for each situation as necessary. Set address and size by searching the sound area map. Since the currently active sound area map is stored in a target memory fixed address, searching is simple.

### Buffer Rewrite Timing

PCM play is fixed at 44.1 KHz by the hardware and therefore 1 word (16 bits, 2 bytes) is played every 22.68  $\mu$ sec from play start. Therefore, since using V int, it is 16 msec,

$$16,666 \text{ msec} \div 22.68 \text{ msec} = 735 \text{ words}$$

which makes it acceptable to rewrite 735 words every 16 msec. If the PCM stream buffer has a minimum of 2 areas, processing can logically be done. To leave an allowance for timing, the buffer can have several areas depending on the situation. If an allowance is made initially for the time lag until play starts, reading can then be performed at the correct time, which makes it easy to estimate how much reading has been done currently by the timer, etc.

In addition, to obtain more accurate rewrite timing, the current memory address during play is stored, and the timing to can be obtained from this information. In this case, time calculation and play address estimation are not necessary, thus allowing programming. However, the accuracy of the data that can be referenced applies only to the first 4 bits of a 16-bit sample, and therefore, monitoring accuracy is per every 4-K sample.

### Load During Transfer

When using DMA, do so in such a way that the sound CPU can operate using the burst write mode. When continuously transferring during DMA operation, the sound CPU cannot be run. The 735 words in the following equation is the numerical value during 44.1 KHz monaural play, and thus the load is half (368 words) when the rate is 20 KHz.

During DMA burst write: 1 word transfer = 4 clock (1 clock = 35 nsec)  
When the allowance is 1 word transfer = 6 clock (= 154.35  $\mu$ sec) 154  $\mu$ sec is required to transmit 735 words as the SH transmission capability.

However, if another play or DSP operation overlaps, a condition arises whereby only about 20-30 percent of the sound memory can be accessed during a single sound chip cycle (22  $\mu$ sec). In that case, that part incurs an extra wait.

When that happens, a 16-word transfer takes 22  $\mu$ sec, 735 words at  $735/16 = 46$ ,  $22 \mu\text{sec} \times 46 = 1,012 \mu\text{sec}$ , and approximately 1  $\mu$ sec is required.

### Time Lag Until Play Start

The sound driver draws commands from the host every 2 msec and therefore it can take a maximum of 2 msec from the time the host issues a command until play starts. This interval can be changed, but actually is thought to go as high as 4 msec.



**Handling Play Frequencies Other Than 44.1 KHz**

The sound chip plays at 44.1 KHz at all times. When playing at other frequencies, handle the pitch parameter should be used. For example, when playing at the half frequency of 22 KHz, play with the pitch at 2x. If play is performed with the play pitch at 1/2, the original pitch reappears. The pitch can be changed by  $\pm 8$  octaves and each octave can be divided into 1,024 steps. This range of accuracy can be used at any frequency. Since simple proportional calculations are used, the calculations can be done ahead of time if the frequency to be used is known. During play, this is passed to the sound as the parameter called "pitch."

**Precautions when Playing at Other than 44.1 KHz**

Changing the pitch changes the play speed (the speed of data read operation from the PCM stream buffer). The speed described in "Buffer Rewrite Timing" is 44.1 KHz, and at other speeds, time control commensurate to (accommodating) the pitch must be performed. The pitch and read speed are inversely proportional, and this calculation is therefore also simple. When reading the same data, if the pitched is halved, the read time is doubled. When the pitch is doubled, the read time is halved.

SEGA Confidential

# Appendix MIDI Specification

## Transmitted Data

### Channel Messages

Transmit nothing

### System Messages

Transmit nothing

### System Exclusive Messages

Refer to "MIDI EXCLUSIVE".

SEGA Confidential



# Recognized Receive Data

## Channel Messages

note off

<u>status</u>	<u>second</u>	<u>third</u>	<u>description</u>
1000 nnnn	0kkk kkkk	0xxx xxxx	note off

note on

<u>status</u>	<u>second</u>	<u>third</u>	<u>description</u>
1001 nnnn	0kkk kkkk	0000 0000	note off
1001 nnnn	0kkk kkkk	0vvv vvvv	note on
			vvv vvvv = 1 - 127

control change

<u>status</u>	<u>second</u>	<u>third</u>	<u>description</u>
1011 nnnn	0000 0000	0vvv vvvv	bank change# LSB
1011 nnnn	0000 0001	0vvv vvvv	modulation
1011 nnnn	0000 0101	0vvv vvvv	portamento time
1011 nnnn	0000 0111	0vvv vvvv	master volume
1011 nnnn	0000 1010	000v vvvv	pan pot
1011 nnnn	0001 0000	0vvv vvvv	mixer change
1011 nnnn	0001 0001	0vvv vvvv	effect pan pot
1011 nnnn	0010 0000	0vvv vvvv	bank change# MSB
1011 nnnn	0100 0000	0vvv vvvv	damper
1011 nnnn	0100 0001	0vvv vvvv	portamento off
			vvv vvvv = 0 - 63
1011 nnnn	0100 0001	0vvv vvvv	portamento on
			vvv vvvv = 64 - 127
1011 nnnn	0101 1011	0vvv vvvv	effect change
1011 nnnn	0111 1011	0000 0000	all note off

program change

<u>status</u>	<u>second</u>	<u>third</u>	<u>description</u>
1100 nnnn	0ppp pppp	— —	voice change

program change

<u>status</u>	<u>second</u>	<u>third</u>	<u>description</u>
1110 nnnn	0bbb bbbb	0bbb bbbb	bender change

## System Messages

1111 0000	Exclusive message	Refer to MIDI EXCLUSIVE
1111 0001	quater flame (sic) Message	Ignore
1111 0010	song position data	Ignore
1111 0011	song select	Ignore
1111 0110	tune request	Ignore
1111 0111	End of exclusive	Refer to MIDI EXCLUSIVE
1111 1000	timing clock	Ignore
1111 1010	start	Ignore
1111 1011	continue	Ignore
1111 1100	stop	Ignore
1111 1110	active sensing	Ignore
1111 1111	system reset	Used

## MIDI Exclusive

### Saturn System Exclusive

1st Byte = 1111 0000 (F0H) : Exclusive Status  
 2nd Byte = 0100 0011 (43H) : YAMAHA ID  
 3rd Byte = 0111 1001 (79H) : DIV Exclusive  
 4th Byte = 0iii iiiH (iiH) : Device ID Header  
 5th Byte = 0000 0001 (01H) : Saturn ID  
 6th Byte = 0fff ffff (ffH) : Command Code  
 7th Byte = 0ddd dddd (ddH) : data  
 : : : :  
 LastByte = 1111 0111 (F7H) : End of Exclusive

**Note:** The device ID is for identification when multiple MIDI devices are connected to the Saturn Development Board. Please set to "0" in normal cases.

### Command Code List

Command	Description	Receive	Transmit
00H	data Dump Request	0	
01H	data Set	0	
02H	Acknowledge	0	0
03H	Not Acknowledge	0	0
04H	Reset	0	
05H	SCSCBIN data change	0	
41H	data Dump		0

Note:

Receive : SATURN Target BOARD EXT.MIDI Instrument  
 Transmit : SATURN Target BOARD EXT.MIDI Instrument



## MIDI Exclusive Format

R: receive; T: transmit

Command=00H : data Dump Request R

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0000 0000	data Dump Request
0000 mmmm	byte size (see Note 1-1)
0000 mmmm	
0000 kkkk	start address (see Note 1-2)
0000 kkkk	
0sss ssss	check sum (see Note 1-3)
F7H	E0X

Receives this message, and transmits Command = 41H message or Command = 03H message.

Command=01H : data Set R

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0000 0001	data Set
0000 mmmm	byte size (see Note 1-1)
0000 mmmm	
0000 kkkk	start address (see Note 1-2)
0000 kkkk	
0000 hhhh	write data (see Note 1-4)
0000 1111	
~	
0000 hhhh	
0000 1111	
0sss ssss	check sum (see Note 1-3)
F7H	E0X

Receives this message, and transmits Command = 02H message or Command = 03H message.

Command=02H : Acknowledge R, T

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0000 0010	Acknowledge
F7H	E0X

Transmits this message when processing completed.

Command=03H : Not Acknowledge R, T

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0000 0011	Not Acknowledge
F7H	E0X

Transmits this message when processing error.

Command=04H : Reset R

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0000 0100	Reset
F7H	E0X

Transmits this message when SATURN reset.

Command=05H : Active SCSPBIN data change R

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0000 0101	SCSPBIN data change
0000 cccc	channel# (see Note 1-5)
0000 0aaa	mode# (see Note 1-7)
0000 dddd	Number (see Note 1-6)
0000 dddd	
00vv vvvv	select# (see Note 1-8)
0000 hhhh	write data (see Note 1-4)
0000 llll	
0sss ssss	check sum (see Note 1-3)
F7H	E0X

Receive this message, and transmits Command = 02H message or Command = 03H message.



Command=41H : data Dump T

<u>Byte</u>	<u>Description</u>
F0H, 43H, 79H, iiH, 01H	Exclusive Header
0100 0001	data Dump
0000 mmmm	byte size (see Note 1-1)
0000 mmmm	
0000 kkkk	start address (see Note 1-2)
0000 kkkk	
0000 hhhh	Dump data (see Note 1-4)
0000 1111	
~	
0000 hhhh	
0000 1111	
0sss ssss	check sum (see Note 1-3)
F7H	E0X

Receives this message, and transmits Command = 02H message or Command = 03H message.

**Note 1-1:** mmmm mmmm = byte size (01H ~ FFH)

**Note 1-2:** kkkk...kkkk = SATURN memory Address (000000H~FFFFFFH)

**Note 1-3:** sss ssss = check sum (00H ~ 7FH)  
7th byte + 8th byte + ... + check sum = 000 0000B

**Note 1-4:** hhhh llll = byte data (00H ~ FFH)  
write/Dump data

**Note 1-5:** cccc = channel number (00H ~ 0FH)

**Note 1-6:** dddd dddd = MIXER, VL, PEG, PLFO, VOICE or LAYER Number

**Note 1-7:**      aaa = mode#                    (see Note 2-1)  
                  0: MIXER change  
                  1: VL change  
                  2: PEG change  
                  3: PLFO change  
                  4: VOICE change  
                  5: LAYER change

**Note 1-8:**      vv vvvv = select#            (see Note 2-1)

SEGA Confidential



**Note 2-1 mode#-select# table**

mode# =	0 (MIXER change)	mode# =	4 (VOICE change)
select#	00H EFSDL [EFREG0]	select#	00H Play Mode
	01H EFPAN [ // ]		01H Bend Range
	02H EFSDL [EFREG1]		02H Portamento time
	03H EFPAN [ // ]		03H Layer Number-1
	- -		04H Volume Bias
	1EH EFSDL [EFREG15]	mode# =	5 (LAYER change)
	1FH EFPAN [ // ]	select#	00H start MIDI note
	20H EFSDL [EXTS0]		01H end MIDI note
	21H EFPAN [EXTS1]		02H PEON
mode# =	1 (VL change)		03H PLON
select#	00H encoded-Rate0		04H D2R
	01H Velocity-Point0		05H D1R
	02H Velocity-Level0		06H AR
	03H encoded-Rate1		07H KRS
	04H Velocity-Point1		08H DL
	05H Velocity-Level1		09H RR
	06H encoded-Rate2		0AH Mod.Wheel
	07H Velocity-Point2		0BH TL
	08H Velocity-Level2		0CH MDL
	09H encoded-Rate3		0DH LFOF
mode# =	2 (PEG change)		0EH PLFOWS
select#	00H DLY		0FH PLFOS
	01H OL		10H ALFOWS
	02H AR		11H ALFOS
	03H AL		12H ISEL
	04H DR		13H IMXL
	05H DL		14H DISDL
	06H SR		15H DIPAN
	07H SL		16H BASE note
	08H RR		17H Fine tune
	09H RL		18H GN & Layer#
mode# =	3 (PLFO change)		19H GN & Layer#
select#	00H DLY		1AH VL#
	01H FRQR		1BH PEG#
	02H HT		1CH PLFO#
	03H FDCT		1DH reserved
	1EH reserved		
	1FH reserved		

Specification submitted to the MIDI Association (February 28, 1994).