# 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 nondisclosure and confidentiality agreement signed separately and in the possession of SEGA. If you have not signed such a nondisclosure 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 described 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 imply 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.



(11/2/94- 002)

SEGA OF AMERICA, INC.
Consumer Products Division

# SATURN
# Sound Driver
# System Interface
## Version 3.03

Doc. # ST-166-R4-012395

# 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-166-R4-012395 **Date** ____________

**Document name** SATURN Sound Driver System Interface Version 3.03

**Corrections:**

| Chpt. | pg. # | Correction |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

**Questions/comments:** ______________________

Where to send your corrections:

Fax: (415) 802-1717
Attn: Evelyn Merritt,
Developer Technical Support

Mail: SEGA OF AMERICA
Attn: Evelyn Merritt,
Developer Technical Support
150 Shoreline Dr.
Redwood City, CA 94065

# Table of Contents

# SATURN Sound Driver System Interface

(This manual covers sound driver version 1.28)

# Change History

4/19/94 • Changed the interface specification for PCM stream playback (left channel and right channel simultaneous playback specification). Driver 1.00

4/26/94 • Changed total byte count of **PCM START** command P4 to word count of 1 channel.
• Changed `00h-FFh` of PCM playback address data to `00h-0Fh`. Driver 1.00

4/27/94 • Changed 68K program area size in system area memory map from 4000h (16KB) to 5000h (20KB).
• Changed 68K work area size in system area memory map from 3000h (12KB) to 4000h (16KB).
• Changed initialization area size for sound system startup in conjunction with the above size changes.
• Added detailed explanations on pan data (32 levels). Changed P1/P2 of **CD-DA LEVEL** command.
• Added **Level** item to P2 of **PCM START** command. Driver 1.00

5/3/94 • Changed MIDI direct control commands P1 from `0-31` to `00h` or `10h` and P2 from `0-255` to `80h-EFh`. Driver 1.00

5/10/94 • Added a detailed explanation of the tempo change parameter (P2).
• Improved timing specification for reading **VOLUME FETCH** command (from 100 ms to selectable timing). Improved the specification for playback start time lag (maximum 16 ms to 4 ms). Driver 1.00

5/24/94 Implementation of feedback from the "System Interface Specification Review Meeting":
• Changed command buffer from 1 screen to 8 screens and changed corresponding work area.
• Changed PCM stream playback from 2 channels to 8 channels and changed corresponding work area and commands.
• Changed the host interface work address from `480h` to `700h`.
• Set new system interface work in 128 bytes beginning from `480h`.
• Optimized the size and location of control commands.
• Added mixer change and mixer parameter change parameters.
• Added explanations on PCM stream playback transfer overhead.
• Changed the priority control (from 8 levels to 128 levels and added replay disable/enable). Driver 1.10

6/8/94 • Changed the **PCM STREAM PLAY** command parameters and updated the accompanying explanations.
• Changed the **MIXER PARAMETER CHANGE** command and updated the accompanying explanations.
• Changed the specification for the **VOLUME ANALYZE START** command (left and right channels are now always output). Driver 1.10

6/15/94 Specifications added and changed by the "Titan Support Meeting":
• Added a **SEQUENCE VOLUME** command.
• Deleted the **FADE IN/FADE OUT** command and changed the fadein/fade out method.
• Changed the specifications for the **MIDI DIRECT CONTROL** command.
• Added a new status for the sequence play position.
• Updated the specifications for the sound priority control and changed the specifications for each of the accompanying command parameters. Driver 1.10

6/22/94
- Added interrupt processing for the host when the play address is updated during PCM stream playback.
- Added the corresponding host interface work and status to accompany the added processing above.
- Clarified the version of the sound driver that corresponds to the manual version (the sound driver version is stored in ASCII format in the header of `SDDRV.TSK`). Driver 1.10

7/6/94
- Added play address update status interrupt.
- Changed the precision of volume data from 8 bits to 16 bits.
- Changed the song play mode in conjunction with deletion of the **FADE IN/FADE OUT** command. Made PCM play address area compatible with stereo playback (1B to 2B).
- Added "sound driver ERROR status work".
- Added notation regarding updating of the data transfer end bit (two locations in the system interface table and sound area map CRNT work). Driver 1.10

7/28/94
- Changed explanation on DSP program load flag and eliminated processing of the related bit operation on the host side.
- Eliminated sequence start P5 (play mode).
- Added the **HARD CHECK** command.
- Added the **PCM parameter change** command.
- Added a sound output mono/stereo mode setting. Added a PCM start P11-12 (playback start position). Driver 1.10

9/2/94
- Changed method of starting sound system.
- Changed sound program (`SDDRVs.TSK`) to include 68k vector table and system information table.
- Simplified method of starting sound system in connection with this.
- Eliminated PCM start P11-12 (playback start position). Driver 1.11

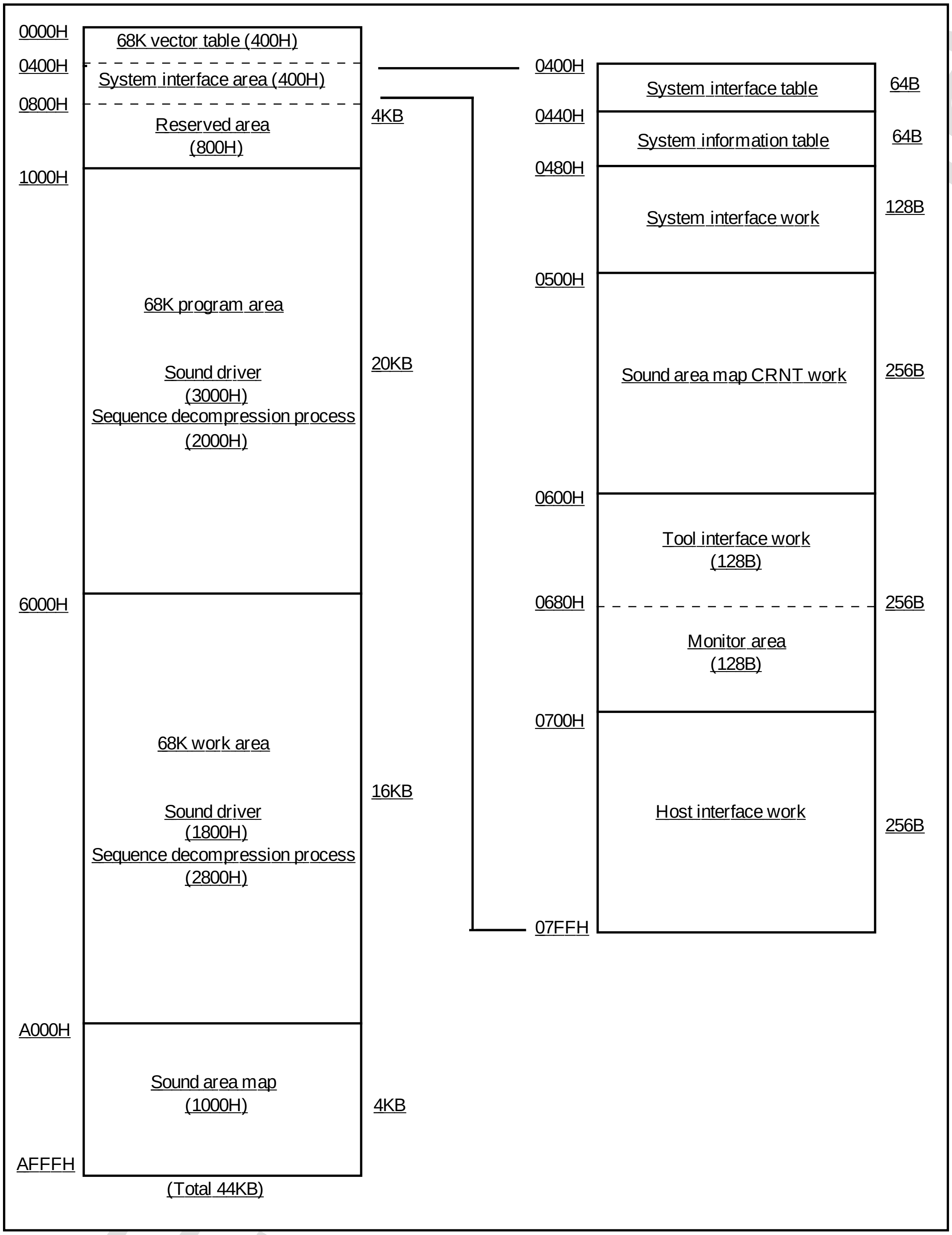
11/14/94
- Added **DSP CLEAR** command (`0Ch`).
- Added **SOUND ALL OFF** command (`0Dh`).
- Added 68000 error code bit map.
- Added explanation on overall structure of sound area map.
- Added explanation on fixes for noise in PCM stream playback loop.
- Corrected part of explanation on PCM stream playback.
- Added explanation on processing method used by the sound driver in response to host commands.
- Corrected part of explanation on sound driver startup method. Driver 1.27

11/24/94
- Added **SEQUENCE PAN** command (`0Eh`).
- Added flag word and control work for 3D sound control. Driver 1.27

12/1/94
- Changed explanation on sound reset and reset clear method added in version 3.01.
- Deleted PCM stream playback loop noise fix added to version 3.01.
- Added explanation on **SEQUENCE PAN** command (`0Eh`).
- Added explanation on flag word and control work for 3D sound control.
- Added explanation on replacing sound data. Driver 1.27

12/16/94
- Added program work and work pointer for Yamaha 3D sound.
- Added COEF/MADRS pointer for Yamaha 3D/QSound and added detailed description of Yamaha 3D work. Driver 1.28

## System Area Memory Map



## System Area Contents

The system area is a fixed area for running the sound driver provided by SEGA, and cannot be used for anything else. Since the mapping of this area cannot be changed, the user need not be concerned about its contents during sound development. The 44 Kbyte (`B000h`) from the top (`00000H`) of the 4Mbit sound memory is used, and the contents of this area are described below.

### 68K Vector Table

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

### System InterfaceArea

This is a fixed area for, among other things, interfacing with the sound driver, a sound development system, and a host system during the incorporation of sound-related content into the game. Its size is `400h` beginning from `0400h` and cannot be changed.

### 68K ProgramArea

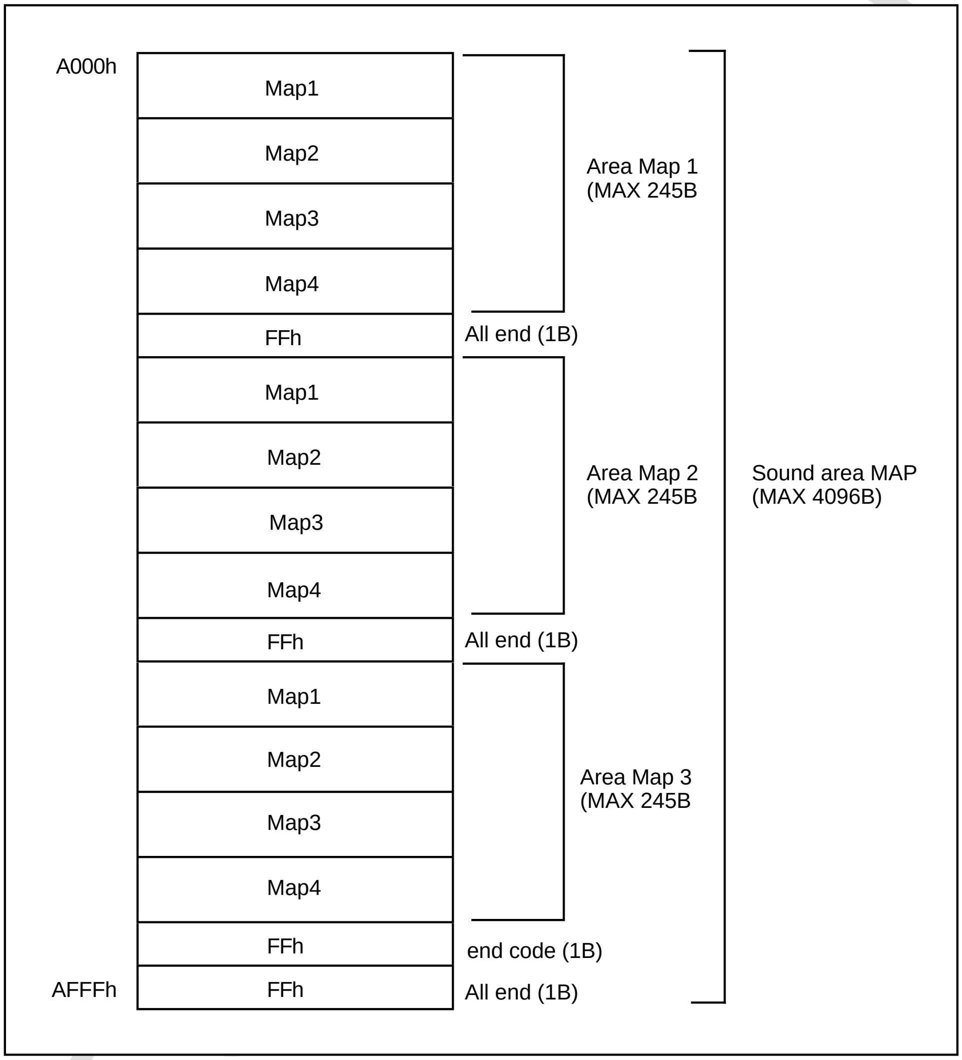
This is the program area for the sound CPU, and is used to store and execute all programs related to sound. The top address and size of this area are stored in the system information table of the system information area.

### 68K Work Area

This is the program work area for the sound CPU. It is also 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 information area.

## Sound Area Map

The entire sound area map is stored here. The sound area map can hold up to 128 area maps in 4096 bytes, with one area map holding up to 32 map data (8B) in 256 bytes. Using the Sound Simulator, one sound area map can be created for one game. Since this area is for storing the entire sound area map, the map data of the currently selected active area is stored in the sound area map CRNT work (`500h-5FFh`) of the system interface area by the **MAPCHANGE** command. The top address and size of this area are stored in the system information table of the system information area.




**Note:** To improve memory efficiency when there are multiple areas, the end code of each area is reduced to one byte. Therefore, byte access is used for access of the sound area map.

## System InterfaceArea

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, sound development systems, the host system during game content 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. It is stored at a fixed address in the sound memory. It also provides a work area for the system interface during sound development as well as the various systems on the development target.

| Address | Offset | Size | Information Data | 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 | bit 7 0: no data<br>1: DSP program load complete<br>bit 6 0: no data<br>1: SCSP BIN load complete<br>bit 5 0: no data<br>1: sequence data load complete |
| 0411 | +11 | 1B | Sound driver load flag | Sound driver program work byte |
| 0412 | +12 | 4B | System interface work pointer | Top address of system interface work (0480h) |
| 0416 | +16 | 2B | Sound driver ERROR status | Sound driver error status store work: word |
| 0418 | +18 | 2B | Hardware check return statues | HARDWARE CHECK command return status store work: word |
| 041A-041F | +1A | 6B | — | Reserved area |
| 0420 | +20 | 4B | 68000 error bit map | 68000 error status bit map |
| 0424 | +24 | 4B | System interface work pointer | Top address of system interface work (0480h) |
| 0428-043F | +2B | 24B | — | Reserved area |

**Note:** The values in parentheses may change with software upgrades.

### DSP Program Load Flag

Only used in sound development. Do not use.

#### System Interface Work Top Address

The same contents at +12 are stored at +24. This spec was added in consideration of the efficient access of 4-byte boundaries by the SH2. To maintain backward compatibility, +12 will not be eliminated.

#### 68000 Error Status Bit Map

Since only the last error data generated can be accessed in the +16 sound driver error status store work, the 68000 error status bit map was added at +20 to facilitate the lookup of up to 32 error status types at the same time in bits.

### 68000 Error Status Bit Map

| | |
|---|---|
| Bit 31: | 68K CPU address error |
| Bit 30: | 68K CPU bus error |
| Bit 29: | 68K CPU exception processing |
| Bit 13: | RAM area for DSP is insufficient for the effect change. |
| Bit 12: | RAM setting for DSP is insufficient for the effect change. |
| Bit 11: | RAM for DSP does not exist in map for the effect change. |
| Bit 10: | DSP program has not been downloaded for the effect change. |
| Bit 9: | DSP program does not exist in map for the effect change. |
| Bit 8: | Sequence bank has not been downloaded at sequence start. |
| Bit 7: | Sequence song number does not exist in sequence bank at sequence start. |
| Bit 6: | Sequence bank does not exist in the map at sequence start. |
| Bit 5: | Program number does not exist in tone bank for the MIDI program change. |
| Bit 4: | Control number and MIDI channel of the tone bank have not been set for the MIDI program change. |
| Bit 3: | Control number and MIDI channel of the tone bank have not been set for the MIDI program change (mixer change). |
| Bit 2: | Tone bank does not exist in the map when changing mixer or tone bank, or tone bank has not been downloaded. |
| Bit 1: | Map does not exist at (`A000h -`) when changing map. |
| Bit 0: | Mixer does not exist in tone bank when changing mixer. Also, tone bank has not been set. |

- Bits 31 to 24 are not automatically cleared, and therefore should be cleared by the host as required.
- Bits 23 to 0 are automatically cleared by the next normal operation.

## 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 |

**Note:** The values in parentheses may change with version upgrades.

## System Interface Work (480H–4FFH: 128B)

This is a work area for exchanging information between systems.

| pointer + xx | Size | Information Data | Contents |
|---|---|---|---|
| +00 | 1B | Interrupt control word | Bit 7 0: PCM play address update interrupt disabled (Host to Sound)<br>1: PCM play address update interrupt enabled |
| +01 | 1B | Interrupt mode | Type of interrupt for host (Sound to Host) |
| +02 | 1B | PCM play number | PCM stream play number during PCM play address update interrupt<br>Bit 7: PCM play number 7 (no update when bit = 0 and update when bit = 1)<br>Bit 6: PCM play number 6<br>Bit 5: PCM play number 5<br>: :<br>Bit 0: PCM play number 0 (Sound to Host) |
| +03 | 1B | Sound control word | Bit 7: 0: sound output STEREO mode (Host to Sound)<br>1: sound output MONO mode |
| +04 | 1B | 3D-Sound control flag | Bit 7: YAMAHA 3D 0: not used 1: in use<br>Bit 6: Qsound 0: not used 1: in use |
| +05 | 1B | QSound COEF | Qsound COEF pointer |
| +06 | 1B | YAMAHA 3D COEF | YAMAHA 3D COEF pointer |
| +07 | 1B | YAMAHA 3D MADRS | YAMAHA 3D MADRS pointer |
| +08-39 | 32B | YAMAHA 3D work | YAMAHA 3D Control Data Storage Area |
| +40-87 | 48B | 3D program work | YAMAHA 3D Program Work Area |
| +88-111 | 24B | — | Reserved area |
| +112-127 | 16B | Titan support | Reserved area for Titan compatibility (cannot be used on Saturn) |

#### Interrupt Control Word

When a PCM stream is played, an interrupt to the host (SH2) can be generated at the point the PCM play address is updated. When an interrupt is necessary, set the specified bit of this control word to “1."  Also, reset this interrupt on the host side. This interrupt request is reset by writing “1” to the corresponding bit (MCIRE:Bit5) in the SCSP interrupt request reset register. Refer to the hardware manual or SCSP manual for more information regarding this process.  The following are the host interrupt types:

- `00h`: none
- `01h`: PCM address update interrupt
- `02h-FFh`: reserved

### 3D-Sound Control Flag/YAMAHA3D Work

This is the interface area for controlling 3D sound. (This is not yet supported in version 1.27 of the sound driver. This area is reserved for adding this function.)

## Yamaha 3D Work (32B)

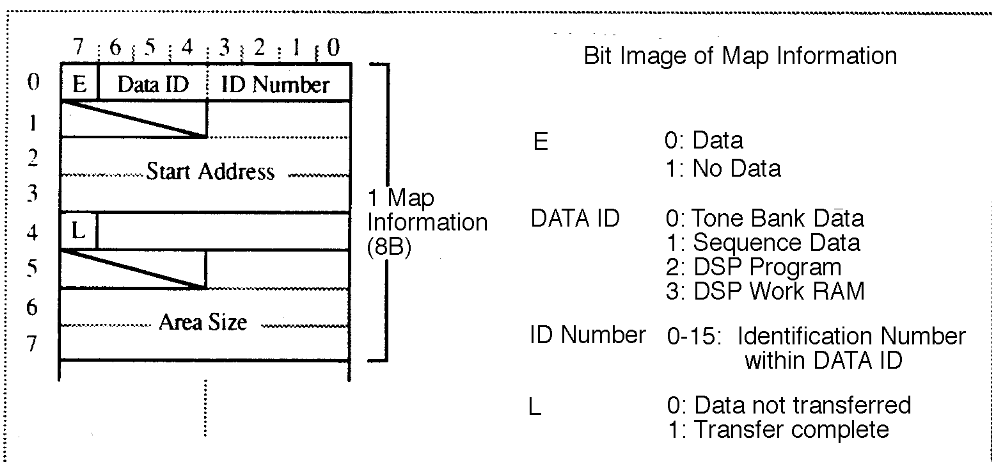
| Offset | Description |
|---|---|
| 00h | CH1 Distance Target position (0 --> 127) |
| 01h | CH1 Horizontal Target position (0 --> 127) |
| 02h | CH1 Vertical Target position (0 --> 127) |
| 03h | CH1 Bit 7 0: nop 1: start movement instruction |
| 04h | CH1 Distance Unit movement (1 --> 127) |
| 05h | CH1 Horizontal Unit movement (-128 --> 127: 2's complement) |
| 06h | CH1 Vertical Unit movement (-128 --> 127: 2's complement) |
| 07h | CH1 Reserved |
| 08h | CH1 Current distance (0 --> 127) |
| 09h | CH1 Current horizontal axis (0 --> 127) |
| 0Ah | CH1 Current vertical axis (0 --> 127) |
| 0Bh | CH1 Reserved |
| 0Ch | CH1 Reserved |
| 0Dh | CH1 Reserved |
| 0Eh | CH1 Reserved |
| 0Fh | CH1 Reserved |
| 10h | CH2 Distance Target position (0 --> 127) |
| 11h | CH2 Horizontal Target position (0 --> 127) |
| 12h | CH2 Vertical Target position (0 --> 127) |
| 13h | CH2 Bit 7 0: nop 1: start movement instruction |
| 14h | CH2 Distance Unit movement (1 --> 127) |
| 15h | CH2 Horizontal Unit movement (-128 --> 127: 2's complement) |
| 16h | CH2 Vertical Unit movement (-128 --> 127: 2's complement) |
| 17h | CH2 Reserved |
| 18h | CH2 Current distance (0 --> 127) |
| 19h | CH2 Current horizontal axis (0 --> 127) |
| 1Ah | CH2 Current vertical axis (0 --> 127) |
| 1Bh | CH2 Reserved |
| 1Ch | CH2 Reserved |
| 1Dh | CH2 Reserved |
| 1Eh | CH2 Reserved |
| 1Fh | CH2 Reserved |

(The offset is the offset position from 488H.)

## Sound Area Map CRNT Work (500H-5FFH: 256B)

The area map (an active map may contain up to 32 map data items ) of the currently selected area is stored in this area. When a map change is received from the host system, the sound system transfers the area map data of the active area to this region. One map data is comprised of 8 bytes, and one area can hold up to 32 map data items (8 x 32 = 256B). Since the map data is stored in random order and the number of map data items can vary, data is searched by type and data number. Data is stored from the top. When there is no more data, the data end bit of the map data is set.

| pointer + xx | Size | Interface Data | Contents | | |
|---|---|---|---|---|---|
| +00 (hex) | 1B | End mark<br>Data ID<br>ID number | Data end bit<br>Data type<br>Data number | (bit 7)<br>(bits 4-6)<br>(bits 0-3) | 0/1<br>0-3<br>0-15 |
| +01 | 3B | Start address | Area start address | (bits 0-19) | 00000-FFFFF |
| +04 | 1B | Load mark | Transfer end bit | (bit 7) | 0/1 |
| +05 | 3B | Area size | Area size | (bits 0-19) | 00000-FFFFF |
| : | | | | | |
| +F8 | 1B | End mark<br>Data ID<br>ID number | Data end bit<br>Data type<br>Data number | (bit 7)<br>(bits 4-6)<br>(bits 0-3) | 0/1<br>0-3<br>0-15 |
| +F9 | 3B | Start address | Area start address | (bits 0-19) | 00000-FFFFF |
| +FC | 1B | Load mark | Transfer end bit | (bit 7) | 0/1 |
| +FD | 3B | Area size | Area size | (bits 0-19) | 00000-FFFFF |




### Precautions

- When the host system has transferred tone bank data, sequence data, DSP program data, or when the data content has been exchanged, be sure to set the transfer end bit of the corresponding map to 1. The sound driver uses this bit to confirm whether the transfer of data is complete.
- DSP WORK RAM can only be set in 8Kbyte (2000h) units. Also, be sure to set the address and size to an even number. An odd number cannot be set.

• Since there can only be one DSP WORK RAM area in the sound memory, ID number (0-15) has no meaning when Data ID = 3 (DSP WORK RAM).

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

Information on the RAM area used by the Wave Editor, Tone Editor, and DSP Linker is stored in this work area. The lower 128 bytes are monitor data area for monitoring the playback status.

| pointer + xx | Size | Interface Data | Contents |
|---|---|---|---|
| +00 (hex) | 2B | — | Reserved |
| +02 | 2B | — | Reserved |
| +04 | 4B | Area Start Address | Wave Editor RAM area start address |
| +08 | 4B | Area Total size | Total Wave Editor RAM area size |
| +0C | 2B | — | Reserved |
| +0E | 4B | Area Start Address | Total Tone Editor RAM area start address |
| +12 | 4B | Area Total size | Total Tone Editor RAM area size |
| +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_ModElementSize | 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<br>Bits 16-23 (2nd byte): MIDI note number 0-127<br>Bits 08-15 (3rd byte): MIDI velocity 0-127<br>Bits 00-07 (4th byte): Reserved |
| | | : | |
| | | : | |
| | | : | |
| +FC-FF | 4B | Voice 32 monitor | Bits 24-31 (1st byte): Program (Voice) number 0-127<br>Bits 16-23 (2nd byte): MIDI note number 0-127<br>Bits 08-15 (3rd byte): MIDI velocity 0-127<br>Bits 00-07 (4th byte): Reserved |

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

This is a work area used for communications between the host system and the sound system. Commands are received in this area from the host system and then data such as status and timing flags are returned. Basic control is performed in the areas listed below, but since specific commands are required by projects (games), an unused area may be used for this purpose when required. Other data such as modes and status can similarly be added or changed as required.

| pointer + xx | Size | Interface Data | Contents |
|---|---|---|---|
| +00 (hex) | 16B | Command block1 | Command block 1 from host to sound system (Host to Sound) |
| +10 | 16B | Command block 2 | Command block 2 from host to sound system (Host to Sound) |
| +20 | 16B | Command block 3 | Command block 3 from host to sound system (Host to Sound) |
| +30 | 16B | Command block 4 | Command block 4 from host to sound system (Host to Sound) |
| +40 | 16B | Command block 5 | Command block 5 from host to sound system (Host to Sound) |
| +50 | 16B | Command block 6 | Command block 6 from host to sound system (Host to Sound) |
| +60 | 16B | Command block 7 | Command block 7 from host to sound system (Host to Sound) |
| +70 | 16B | Command block 8 | Command block 8 from host to sound system (Host to Sound) |
| +80 | 2B | song 1 mode/status | Play sequence 1 mode/status (Sound to Host) |
| +82 | 2B | song 2 mode/status | Play sequence 2 mode/status (Sound to Host) |
| +84 | 2B | song 3 mode/status | Play sequence 3 mode/status (Sound to Host) |
| +86 | 2B | song 4 mode/status | Play sequence 4 mode/status (Sound to Host) |
| +88 | 2B | song 5 mode/status | Play sequence 5 mode/status (Sound to Host) |
| +8A | 2B | song 6 mode/status | Play sequence 6 mode/status (Sound to Host) |
| +8C | 2B | song 7 mode/status | Play sequence 7 mode/status (Sound to Host) |
| +8E | 2B | song 8 mode/status | Play sequence 8 mode/status (Sound to Host) |
| +90 | 2B | Total volume L | Total volume data L (0000h-7FFFh) (Sound to Host) |
| +92 | 2B | Total volume R | Total volume data R (0000h-7FFFh) (Sound to Host) |
| +94 | 2B | H-vol L | Treble range volume data L (0000h-7FFFh) (Sound to Host) |
| +96 | 2B | H-vol R | Treble range volume data R (0000h-7FFFh) (Sound to Host) |
| +98 | 2B | M-vol L | Midrange volume data L (0000h-7FFFh) (Sound to Host) |
| +9A | 2B | M-vol R | Midrange volume data R (0000h-7FFFh) (Sound to Host) |
| +9C | 2B | L-vol L | Bass range volume data L (0000h-7FFFh) (Sound to Host) |
| +9E | 2B | L-vol R | Bass range volume data R (0000h-7FFFh) (Sound to Host) |
| +A0 | 2B | PCM 0 adrs | PCM play number 0 play address (00h-0Fh) (Sound to Host) |
| +A2 | 2B | PCM 1 adrs | PCM play number 1 play address (00h-0Fh) (Sound to Host) |
| +A4 | 2B | PCM 2 adrs | PCM play number 2 play address (00h-0Fh) (Sound to Host) |
| +A6 | 2B | PCM 3 adrs | PCM play number 3 play address (00h-0Fh) (Sound to Host) |
| +A8 | 2B | PCM 4 adrs | PCM play number 4 play address (00h-0Fh) (Sound to Host) |
| +AA | 2B | PCM 5 adrs | PCM play number 5 play address (00h-0Fh) (Sound to Host) |
| +AC | 2B | PCM 6 adrs | PCM play number 6 play address (00h-0Fh) (Sound to Host) |
| +AE | 2B | PCM 7 adrs | PCM play number 7 play address (00h-0Fh) (Sound to Host) |

| | | | |
|---|---|---|---|
| + B0 | 2B | Sequence 0 position | Sound control number 0 play position (0000h-FFFFh) (Sound —> Host) |
| + B2 | 2B | Sequence 1 position | Sound control number 1 play position (0000h-FFFFh) (Sound —> Host) |
| + B4 | 2B | Sequence 2 position | Sound control number 2 play position (0000h-FFFFh) (Sound —> Host) |
| + B6 | 2B | Sequence 3 position | Sound control number 3 play position (0000h-FFFFh) (Sound —> Host) |
| + B8 | 2B | Sequence 4 position | Sound control number 4 play position (0000h-FFFFh) (Sound —> Host) |
| + BA | 2B | Sequence 5 position | Sound control number 5 play position (0000h-FFFFh) (Sound —> Host) |
| + BC | 2B | Sequence 6 position | Sound control number 6 play position (0000h-FFFFh) (Sound —> Host) |
| + BE | 2B | Sequence 7 position | Sound control number 7 play position (0000h-FFFFh) (Sound —> Host) |
| + C0 - FF | 64B | Reserved | |

### PCM Play Numbers 0-7 Play Address

These are data positions during playback which are denoted as the number of samples relative to the start of the PCM Stream playback buffer. Byte numbers apply when playback is in 8 bits, and word numbers apply when playback in 16 bits. Although the number of samples is 16-bit data, only the upper 4 bits can be monitored, which means that PCM data replacement can only be controlled in 4K sample units.

The Play address is set to "MONO" for mono play, and set to the "Rch play address" for stereo playback. Since the right and left channel play addresses are the same during stereo playback, the "Lch play address" is not set.

| | |
|---|---|
| +0 | MONO or Rch play address |
| +1 | Lch play address |

**Note 1**
Although access to the sound memory by the host can be performed in 1-byte (byte), 2-byte (word) and 4-byte (long word) units, note that the hardware takes the same amount of processing time whether it is byte access or word access. Also, one long word access is slightly faster when compared with two word accesses.

**Note 2**
Use caution since the sound CPU cannot operate while the host system is accessing the sound memory. Keep reads and writes of the sound memory by the host system to a minimum and do not allow continuous access over long periods. Refer to the *PCM Stream Playback* section for guidelines.

## Command Block

Since eight 16-byte command blocks have been provided for commands from the host to the sound system, a maximum of eight commands can be issued at the same time.

**Command Block**

| | |
|---|---|
| +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) |

- The sound driver processes the command block in order from block 1 to block 8 and then immediately begins processing block 1 again. Also, since the driver does not look at the next command block until it is completely finished processing each command block, the command processing order set by the host is guaranteed.
- After the sound driver detects a command and finishes processing it, it clears that command to `00h`. Until then, the command is being processed (or waiting for processing).
- Since there are command sequences, such as map changes and mixer changes, where their order is important, set commands so that the sound driver processes them in the correct order. (If open blocks from 1 to 8 are simply searched and commands are set into them, then commands may be executed in an order different from the order written by the host.)

### Play Sequence Mode/Status

Since the playback status of the sequence is continually updated as mode and status data, refer to the mode and status areas shown below to find the sequence's current status. A fade is set by a sequence volume command, and when the fade is completed, either a Play or Stop is set.

| | |
|---|---|
| +0 | Song sound mode |
| +1 | Status |

**Song Playback Mode**

| | |
|---|---|
| `00`: | Stop |
| `01`: | Play |
| `02`: | Fade |
| `03`: | Play pause |
| `04`: | Fade pause |

**Status**

| | |
|---|---|
| `00`: | Normal |
| `01-FF`: | Status code |

### Status Detail

| | |
|---|---|
| `80h`: | Out of sequence resolution range |
| `81h`: | No sequence tempo data |
| `82h`: | No sequence event data |
| `83h`: | Out of sequence tempo data range |
| `FFh`: | Data error during sequence data decompression |

## Starting the Sound Driver

The sound driver is started by transferring the sound program to the sound memory and clearing the CPU reset signal. However, since the contents of memory cannot be guaranteed at power-on, memory initialization is required. Use the following procedure to start the sound driver.

### Sound System Startup

The following operation will start the sound CPU and run the sound program from the beginning of the program. SCSP registers, resets and other hardware issues are described in greater detail in the SCU, SCSP and SMPC user's manuals.

1. Reset sound CPU (halt the 68000).
   Execute the SMPC I/F library's SOUND OFF.
2. Initialize the SCSP registers to 4Mbit.
   Write `02h` to the SH2 address `25B00400` in bytes.
3. Zero-clear the system area (`B000h` from `00000h`).
   Zero-clear addresses `25A00000` to `25A0AFFF`.
4. Transfer the sound program (`SDDRVS.TSK`) to the sound memory from `00000h`.
   Transfer the sound program (`SDDRVS.TSK`) to the area from SH2 address `25A00000`.
5. Transfer the sound area map to the sound memory.
   Transfer the sound area map to the area from SH2 address `25A0A000`.
6. Clear reset of the sound CPU (start the 68000).
   Execute the SMPC I/F library's SOUND ON.
7. Other settings should be made via the system interface work (`480h-4FFh`) control words.

### Sound Control

Sound data is required to play songs and sound effects. Preparation for playback is completed by transferring the sound data to the sound memory. After this, the playback is controlled by commands such as those that start and stop song playback and start and stop sound effects playback. Use the following steps to transfer sound data and to control sound.

1. Issue the **DSP CLEAR** command (`0Ch`)
2. Issue the **SOUND ALL OFF** command (`0Dh`).
3. Transfer the sound block data corresponding to the area from `B000h` in the sound memory.
   Transfer the sound block data to an area from SH2 address `25A0B000`.
4. Issue the **MAP CHANGE** command (`08h`) corresponding to the area.
5. Set the transfer complete bit of the sound area map CRNT work to 1.
6. Issue the **EFFECT CHANGE** command (`83h`) as required.
7. Issue the **MIXER CHANGE** command (`87h`) as required.
8. Perform control of sequence data such as start and stop.
9. Should the area change, repeat from 1 and 8 above.

**Note:** If the map is changed or if sound data in-use is replaced while sound is still playing or the DSP is operating, the system may crash. Be sure to perform steps 1 and 2 above to avoid this..

## Sound Data

There are three types of sound data, each of which can be located freely in memory according to the area map. However, if the amount of data (number of files) becomes large, it becomes difficult for the host to manage it. As a result, file transfer and transfer size errors may occur more frequently. Since it is convenient to include all 4Mbit worth of data (does not actually include the system area and the DSP ring buffer parts) in one file, the sound developer should make one large file for each area as the sound block data. The sound block data can be created with the sound simulator.

| | |
|---|---|
| • Tone bank data: | Tone definition data comprising wave data and tone parameters |
| • Sequence data: | Performance data in which the playback data for songs and sound effects are stored |
| • DSP program: | The DSP microprogram itself and coefficient data |
| • Sound block data: | One file containing all sound data located in the sound memory according to the area map |

## Replacing Sound Data

Each set of sound data (tone bank data, sequence data, DSP program) can be replaced as required by the game scene or conditions. Transfer new data as required. The procedure is shown below.

1. Set the transfer end bit of the sound area map CRNT work to 0.
2. Transfer the new data according to the MAP data (transfer address and area size) of the sound area map CRNT work.
3. Set the transfer end bit of the sound area map CRNT work to 1.

Take the following precautions when transferring data:

- If the transfer data is greater than the area size, other data will be destroyed. Therefore, do not transfer data greater in size than the area size.
- Tone bank data and sequence data cannot be changed while they are being played back. When replacing data, perform a transfer after stopping playback.

## Sound Control Commands

Below are descriptions of the commands and parameters issued to the sound system from the host system. The parameters vary in length depending on the command. Basic control is performed using the following commands, but since special commands become necessary depending on the project (game), unused command codes may be allocated as required. Commands and parameters can similarly be added or changed when required.

| Command Name | Command Data | Parameter Data |
|---|---|---|
| Reserved | 00 (hex) | Nothing |
| SEQUENCE START<br>(S - -) | 01 | P1 0-7: Sound control number<br>P2 0-15: Sequence bank number<br>P3 0-127: Sequence song number<br>P4 0-31: Priority level |
| SEQUENCE STOP<br>(S - -) | 02 | P1 0-7: Sound control number |
| SEQUENCE PAUSE<br>(S - -) | 03 | P1 0-7: Sound control number |
| SEQUENCE CONTINUE<br>(S - - ) | 04 | P1 0-7: Sound control number |
| SEQUENCE VOLUME<br>(S - -) | 05 | P1 0-7: Sound control number<br>P2 0-127: Sequence volume<br>P3 0-255: Fade rate |
| TEMPO CHANGE<br>(S - -) | 07 | P1 0-7: Sound control number<br>P2 -: dummy<br>P3-P4 +32767 to -32768 Given the standard tempo (000h), 1000h (4096) equals double the tempo. The tempo is 1/2 when the value is negative |
| MAP CHANGE<br>(S P - -) | 08 | P1 0-255: Area number of sound area map that is changed. |
| MIDI DIRECT CONTROL<br>(S - -) | 09 | P1 00h-FFh: MIDI command word (see bit image below for details)<br>P2 00h-FFh: MIDI channel word<br>P3 00h-7Fh: MIDI data 1<br>P4 00h-7Fh: MIDI data 2 |
| VOLUME ANALYZE START<br>(- - C) | 0A | Nothing (start of volume acquisition) |
| VOLUME ANALYZE STOP<br>(- - C) | 0B | Nothing (end of volume acquisition) |
| DSP STOP<br>(S P C) | 0C | Nothing (halts DSP) |
| SOUND ALL OFF<br>(S P C) | 0D | Nothing (stops all play slots) |
| SEQUENCE PAN<br>(S - -) | 0E | P1 07: Sound control number<br>P2 bit 7: 0: Control OFF 1: Control ON<br>bits 6-0: MIDI pan data (00h-7Fh)<br>00h: left <- -> 40h: center <- -> 7Fh: right (MIDI) pan is capable of 128 steps, but since SCSP pan only supports 32 steps, the lower 2 bits of the MIDI pan data are ignored.) |

(Continued on next page)

## MIDI Direct Control Bit Image

| | | |
|---|---|---|
| CD-DA LEVEL<br>(- - C) | 80 | P1 00h-E0h: Sound control number<br>8 levels: 00h (off), 20h, 40h, 60h, 80h, A0h, C0h, E0h (max)<br>P2 00h-E0h: CD-DA level Right<br>8 levels: 00h (off), 20h, 40h, 60h, 80h, A0h, C0h, E0h (max) |
| CD-DA PAN<br>(--C) | 81 | P1 0-31: CD-DA pan left; 32 levels<br>P2 0-31: CD-DA pan right; 32 levels |
| TOTAL VOLUME<br>(S P C) | 82 | P1 0-15: 16 levels, 0 if OFF |
| EFFECT CHANGE<br>(S P C) | 83 | P1 0-15: Effect bank number |
| PCM START<br>(- P -) | 85 | P1 bit 7 0: Mono 1: Stereo<br>bits 6-5 Not used<br>bit 4 0: 16 bit PCM 1:8 bit PCM<br>bits 3-0 0-7: PCM stream playback number<br>P2 bits 7-5 0-7: Direct sound level, 8 levels<br>bits 4-0 0-31: Direct sound pan, 32 levels (ignored in stereo)<br>P3-P4 0000h-FFFFh: PCM stream buffer start address<br>(upper 16 bits of 20 bit data)<br>P5-P6 0000h-FFFFh: PCM stream buffer size<br>(number of samples in 1 channel)<br>P7-P8 0000h-FFFFh: Pitch word (SCSP pitch register data: Oct&FNS)<br>P9 bits 7-3 0-15: Effect in select (P9 = Rch or MONO)<br>bits 2-0 0-7: Effect send level, 8 levels<br>P10 bits 7-3 0-15: Effect in select (P10 = Lch)<br>bits 2-0 0-7: Effect send level, 8 levels |
| PCM STOP<br>(- P -) | 86 | P1 0-7: Stop playback PCM stream playback number |
| MIXER CHANGE<br>(S P C) | 87 | P1 0-15: Tone bank number<br>P2 0-127: Mixer Number |
| MIXER<br>PARAMETER CHANGE<br>(S P C) | 88 | P1 0-17: Effect out select<br>P2 bits 7-5 0-7: Effect return level, 8 levels<br>bits 4-0 0-31: Effect pan, 32 levels |
| HARD CHECK<br>(- - -) | 89 | P1 0-5: check items 0 - DRAM 4Mbit read/write<br>1 - DRAM 8Mbit read/write<br>2 - SCSP MIDI<br>3 - Sound generator output (L/R)<br>4 - Sound generator output (L)<br>5 - Sound generator output (R) |
| PCM PARAMETER<br>CHANGE<br>(- P -) | 8A | P1 0-7: PCM stream playback number<br>P2 bits 7-5 0-7: Direct sound level, 8 levels<br>bits 4-0 0-31: Direct sound pan, 32 levels<br>P3-P4 0000h-FFFFh: Pitch word<br>P5 bits 7-3 0-15: Effect in select (P5 = Rch or MONO)<br>bits 2-0 0-7: Effect send level, 8 levels<br>P6 bits 7-3 0-15: Effect in select (P6 = Lch)<br>bits 2-0 0-7: Effect send level, 8 levels |
| Reserved | 8B-FF | Nothing |

(S P C) S: Command for sequence playback or command related to sequence playback.
P: Command for PCM stream playback or command related to PCM stream playback.
C: Command for CD-DA playback or command related to CD-DA playback.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Priority level | | | | | CMD | | |
| 1 | KNo | | | MIDI Ch | | | | |
| 2 | MIDI Data #1 | | | | | | | |
| 3 | MIDI Data #2 | | | | | | | |

| | | |
|---|---|---|
| Priority level | : 0-31 | Priority of sound playback at sequence start |
| CMD | : 0-7 | MIDI command 8-F (8-F --> 0-7) |
| KNo | : 0-7 | Sound control number |
| MIDI Ch | : 0-31 | MIDI channel |
| MIDI Data #1 | : 0-127 | MIDI data byte #1 |
| MIDI Data #2 | : 0-127 | MIDI data byte #2 |

## Fade In Method

Issue a **SEQUENCE VOLUME** command before issuing a **START** command. The volume will fade in according to the fade rate starting from volume = 0 until the specified sequence volume is reached. The fade-in curve can be freely controlled by using the **SEQUENCE VOLUME** command two or more times.

## Fade Out Method

Issue a **SEQUENCE VOLUME** command at volume = 0. The volume will fade out according to the fade rate from the current sequence volume until volume = 0. The fade-out curve can be freely controlled by using the **SEQUENCE VOLUME** command two or more times.

**Note 1**

Since commands such as **MIDI DIRECT CONTROL** and **PCM START** are closely related to the MIDI channel and tone, as well as the DSP program and mixer settings, plan ahead with the sound developer regarding the sound content.

**Note 2**

Since a special DSP program is required for the frequency band based volume analysis, down- load the special DSP program using the **EFFECT CHANGE** command before issuing the **VOLUME ANALYZE** command. A DSP program is not required for the main volume. Also, since data is updated in 16 ms intervals in volume analysis, volume data should be read in 16 ms or longer intervals.

**Note 3**

When stereo is specified by PCM start, the first half beginning from the top is processed as the right channel and the second half is processed as the left channel of the area specified by `P3-P4`. The PCM stream buffer must start from an even address and must have an even size. Since the start address of the PCM stream buffer is the upper 16 bits of the 20-bit data (`00000h-FFFFFh`), the lower four bits are always 0. The SCSP pitch register word's octave + F number is specified as the pitch word of `P7-P8`. Refer to the SCSP manual for details on pitch.

**Note 4**

The sound CPU cannot operate during the transfer (host to sound memory) of PCM stream data, and therefore its operation cannot be guaranteed when data is transferred continuously over long periods while a song or sound effect is being played at the same time. Use DMA burst write or transfer data in intervals.

**Note 5**

PCM stream playback can place a considerable burden on data transfer depending on the playback rate, and therefore all eight voices may not be played. Refer to the *Overhead During Data Transfer* section for guidelines.

## Parameters

### Sound Control Number
Up to eight sequences can be controlled when sequences are played, and are specified by the sound control numbers 0-7 when the sequences are started. Stop, pause, continue and other commands are executed according to these sound control numbers.

### Sequence Bank Number
When multiple sequence data banks are mapped in the currently active map, the sequence bank number identifies the sequence data bank in the active map. Since up to 16 banks can be held in one map, this number ranges from 0 to 15. If there is only one sequence data bank, then the number is always 0.

### Sequence Song Number
Since multiple sets of sequence data can be stored in the sequence data bank, the sequence song number identifies the sequence data set in the sequence data bank. Up to 256 sequences can be stored as long as the area size allows.

### Priority Level
This specifies the priority level in 32 levels (0-31) when sequences are played. The highest priority is assigned 0, and the lowest, 31. When all 32 of the sound slots are full, subsequent sequences are given last-note priority and played accordingly. Voice allocation for playing these sounds is controlled by this priority level.

### Sequence Volume
The sequence volume is specified from 0-127. The volume is lowest at 0 and highest (the original sound) at 127. The fade rate specifies the time until the specified volume is reached. The final volume after a fade (specified sequence volume) is retained as the current sequence volume. If a fade is not required, it is possible to set only the sequence volume by setting the fade rate to 0.

### Fade Rate
The time it takes to reach the specified sequence volume from the current sequence volume is specified by the rate (0-255). The larger the value, the faster the change, with 255 being the shortest time and 1 being the longest. If 0 is specified, the sequence volume is set without a fade. A fade is performed from the currently set sequence volume to the specified sequence volume, and therefore a fade in or fade out occurs depending on whether the specified sequence volume is larger or smaller than the current sequence volume.

### Effect Bank Number
When multiple DSP program banks that store DSP microprograms are in the current active map, the effect bank number identifies each DSP program bank in the active map. Since up to 16 banks can be held in one map, the number ranges from 0 to 15. This number is 0 if there is only one sequence data bank.

### PCM Stream Playback Number

A maximum of eight sounds can be played during PCM stream playback, and are specified by the PCM stream playback numbers 0-7. Caution must be paid when playing in stereo as two voices are played for each PCM stream play number. The maximum total of voices for mono and stereo is eight. If eight voices are exceeded, the command is ignored.

### Tone Bank Number

When multiple tone data banks are mapped in the current active map, the tone data bank number identifies each bank. Since up to 16 banks can be held in one map, the number ranges from 0 to 15. When there is only one tone data bank, this number is always 0.

### Mixer Number

Specify mixer numbers when multiple mixers are contained in a tone bank. Since a maximum of 128 mixer data settings can be held in one tone bank, a value from 0 to 127 can be specified.

### Effect In Select

This is the DSP effect input channel. There are 16 input channels for the DSP, so a value from 0 to 15 must be specified. The signal input to the effect input channel undergoes signal processing in the DSP and is output from the effect output channel of the same number. Any value from 0 to 15 can be used when a DSP effect is not used, but the effect send level must be set to 0.

### Effect Out Select

This is the DSP effect output channel. Since there are 16 output channels for the DSP effect, a value from 0 to 15 may be specified. In addition, the two output channels (Lch/Rch) for CD-DA are output from the DSP effect output channels. Specify 16 and 17 when CD-DA is selected.

### Effect Send Level

The level of the direct signal input to the effect input channel (0-15) selected by Effect In Select is specified as a value from 0 to 7. Nothing is input when 0 is specified, and input is set at maximum volume when 7 is specified. Specify 0 when the DSP is not used.

### Effect Return Level

When selected via Effect Out Select, the effect volume output level from the effect output channel (0-15) is specified as a value from 0 to 7. Nothing is output when 0 is specified, and output is set at maximum volume when 7 is specified.

**Details of Pan Effect (32 Steps)**

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

## Precautions for Using Sequence Pan (`Command 0Eh`)

- This is pan control for only the direct signal component and has no effect on the pan of the effect signal.
- When bit 7 of address `25a00483H` (byte) is `H:MONO`, all pan control is disabled (Stereo/Mono selector switch).
- When the control command `SEQUENCE PAN = ON` is issued, pan control [`BnH,0AH,xxH`] in the song as well as the pan control and the tone's layer-pan set by the **MIDI DIRECT CONTROL** command (`09H`) of the corresponding control number are disabled.
- When program change (`CnH,xxH`) in a song is issued, enabling/disabling of the tone's layer-pan data is dependent on the ON/OFF of the corresponding command. That is, once this command (= ON) is set, the pan set up to that point remains enabled even if a new song is requested.
- The conditions under which the tone's layer-pan data is enabled are as follows:
  1: Stereo mode is set
  2: Sequence pan control is OFF.
  3: MIDI pan control has not been issued after Sequence Pan Control has been set to OFF.
  4: Note-on is issued under the above conditions.
- Even if `SEQUENCE PAN = OFF` is issued, it has no effect on the pan position of the sound being played.

**Example** (must be in stereo mode):

| | |
|---|---|
| Control No. 3 SEQUENCE START —— | Pan setting is in the layer data. |
| ↓ | |
| Control No. 4 SEQUENCE PAN ON —— `0Eh, 00h, 04h, 7Fh` | Pan is set to `7Fh` (far right)<br>Has no effect on the pan setting of control No. 3 |
| ↓ | |
| Control No. 4 SEQUENCE START —— | The pan setting of sounds under control No. 4 are `7Fh` (far right) |
| ↓ | |
| Control No. 4 MIDI PAN CONTRL —— `Bnh, 0Ah, xxh` | No change |
| ↓ | |
| Control No. 3 MIDI PAN CONTRL —— `B7h, 0Ah, 00h` | The pan setting (affects sounds being played) of MIDI channel #7 of control No. 3 shifts to `00h` (far left). The pan position of other MIDI channels is set by the layer data. Has no effect on the pan setting of control No. 4. |
| ↓ | |
| Control No. 4 SEQUENCE PAN ON —— `0Eh, 00h, 04h, 00Fh` | The panned sound (including those being played) of all sound under control No. 4 moves to `00h` (far left). Has no effect on the pan setting of control No. 3. |

↓

Control No. 4 SEQUENCE PAN OFF —— There is no change in the sounds of control No. 4, and sounds played hereafter are dependent on the pan setting of the layer data.

↓

Control No. 4 MIDI PAN CONTRL —— The pan position (including sounds being played) of MIDI channel #6 of control No. 4 moves to `00h` (far left). The pan position of other MIDI channels is set by the pan setting of the layer data. The pan of control No. 3 is not affected.

`B6h, 0Ah, 00h`

↓

Control No. 4 SEQUENCE START —— The pan position (including sounds being played) of MIDI channel #6 of control No. 4 is determined by the pan setting of the layer data controlled by program changes [`C6h, xxh`] within the song.

↓

Control No. 4 SEQUENCE PAN ON —— The pan position (including sounds being played) of all control No. 4 sounds moves to `7Fh` (far right), and has no effect on the pan setting of control No. 3.

`0Eh, 00h, 04h, 7Fh`

↓

Control No. 4 SEQUENCE START —— No change in the pan setting of control No. 4.

## PCM Stream Playback

### Overview

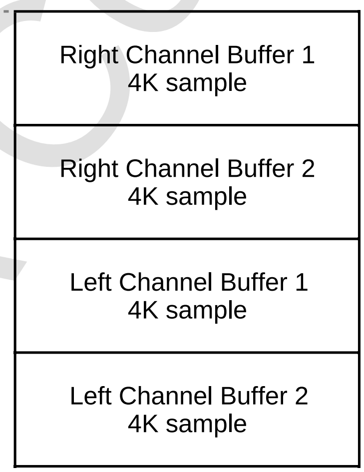
PCM Stream playback is used to play continuous, long PCM data by providing a ring buffer for PCM playback in the sound memory. By using the loop playback function of the sound chip to perform processing that writes new PCM data to buffer locations that have been played, long PCM data can be played back continuously. It is possible to write new data in areas that have been played by knowing the position of the data being played. This current playback position is determined by using 4K sample units. Therefore, by providing at least two 4K sample buffers (for an 8K sample), it becomes possible to swap data.

**Note:** One 8 bit sample becomes one byte and one 16 bit sample becomes 2 bytes. For example, a 4K sample would be 4096 bytes when 8 bits and 8192 bytes when 16 bits.

### Setting Up a Buffer

A buffer can be specified at any location in memory. Normally, areas unused by tone data or song data are used, but memory areas that are not used simultaneously with PCM stream play can also be used. The maximum buffer size that can be set up for stream playback of one channel is a 64K sample. The buffer size specified by the command parameter is the number of samples for one channel, and therefore twice the memory of the specified size is used in stereo, so caution must be used.

**Example:** Using the PCM stream buffer for stereo playback (one channel is divided up into two parts):



### Playback Method

1. The first PCM data is transferred to the entire PCM playback buffer (`buff1, buff2`).
2. `[PCM start : 85h]` is issued by the **SOUND CONTROL** command.
3. When playback of `buff1` (4k sample) is finished, the next PCM data is transferred to `buff1`.
4. When playback of `buff2` (4k sample) is finished, the next PCM data is transferred to `buff2`.
5. Continuous PCM data can be played back by repeating this operation.
6. To stop PCM playback, `[PCM stop : 86h]` is issued by the **CONTROL** command.

**Note:** In the case of stereo, the above processing is performed in two areas for the right and left channels.

### Timing for Buffer Rewrites

Since the data position currently being played back is stored in the host interface work area, the next PCM data is written to the buffer that has completed playback. However, since the data position reference is restricted by hardware to only the upper 4 bits of the 16-bit sample (relative number of samples from the top of the PCM stream playback buffer), the monitoring precision is limited to every 4K sample. PCM playback processing is possible if there are at least two PCM stream buffers. However, multiple buffers can be used to provide a greater margin for timing. Since it is possible to generate an interrupt to the host according to the timing of the data position being updated, this feature can be used if necessary. The control information of this interrupt is in the system interface work.

### Overhead During Data Transfer

Since the PCM playback slot is fixed at 44.1 KHz, 1 sample is played back every 22.68 µs from the start of playback. Therefore when Vint is used, 735 samples (16,666 µs ÷ 22.68 µs) need to be rewritten (transferred) every 16 ms for one stream playback channel.

During DMA burst writes:

1 word transfer = 4 clock cycles (1 clock cycle = 35 ns)
Assuming 1 word transfer = 6 clock cycles to allow a margin of safety, then 6 clock cycles x 735 words = 4410 clock cycles (= 154.35 µs).
The SH2 requires approximately 154 µs to transfer 735 words (1,470 bytes).

However, if other sounds are being generated, or if the DSP is being used, the sound chip can only access the sound memory 20% to 30% of the time per one sound chip cycle (22 µs). This requires extra wait time, and since only 16 words can be transferred in one cycle (22 µs) of the sound chip, about 1 ms is required to transfer 735 words.

The 735 samples referred to by the equation above is for mono playback at 44.1 KHz. Overhead can be reduced to half if playback at 20 KHz is acceptable (equivalent to 368 samples).

When using DMA, avoid long, continuous transfers so that the sound CPU can operate. The sound CPU cannot operate during DMAs if data is transferred continuously.

### Supporting Playback Frequencies Other Than 44.1 KHz

The sound chip is normally set to play at 44.1 KHz, but the pitch parameters enable playback at other frequencies. For example, if a 22-KHz sound (half of 44.1 KHz) is played back, it is played at twice its original pitch. Therefore, the original pitch can be reproduced by halving the playback pitch. Since the pitch can be changed by ±8 octaves and one octave can be divided into 1024 steps, any frequency can be accommodated within this range. The calculation is a simple ratio calculation, and if the frequency used is known, it can be calculated in advance. This should be passed to the sound system during playback as the pitch parameter.

**Note:** Since mixer data is required when using effects, tone bank data must be provided even if there is no sequence playback. Mixer data includes output levels for each effect channel as well as pan data. This data is included in the tone bank data.