SOUND.pdf SEGA DTS

This document contains information on the following:

----

1. Overview/Release Notes of the Sega Saturn Sound Development Tools

----

The Saturn Sound Development Tools are all Macintosh applications. This DTS CD contains all of the tools necessary to create and incorporate sound into the Saturn. These are the four main development tools used for sound design on the Saturn. They are used in conjunction with the Saturn Sound Box which is directly connected to the Macintosh via SCSI. It is suggested that you add an in-line SCSI terminator in-between the sound box and the end of the SCSI cable.

Release Notes in English!
All the previously available Japanese release notes for the tools have been translated in to English for your reference purposes. Look for the release notes inside each tool’s folder. Updates to this information for the Sound Simulator 3.02, Tone Editor

2.07, and Sound Driver 2.10 are currently being translated and will be included on the DTS Website shortly.

Sound Simulator
This tool allows the sound designer to map out the sound memory and to simulate performance in game code. Refer to document ST-81 for more detailed instructions. The Sound drivers must be included in the same directory as the Sound Simulator. For more detailed information please refer to ST-168. To read more about MIDI conversion please refer to ST-66.

The **SndSim301** folder contains version 3.01 of SoundSimulator along with version 2.10 of the sound driver. **PLEASE NOTE:** Sound Driver 2.10 is **NOT** backwards compatible; if your tone bank is not created using at least 2.0 of the Tone Editor, then you will NOT get any sound out of it. A testbed for this will be included on the DTS Website shortly.

SoundSimulator, version 3.00 and above, offers a more user-friendly English interface along with improvements in the Song Collector and Function Key features. It is also capable of supporting both Japanese and English Macintosh Sytem environments. The differences between Sound Driver 2.04 and 2.10 are currently being translated. For your reference, the **Sound Driver Readme** details differences between previous versions of the sound driver (2.04 and below).

Tone Editor
This tool is a voice/patch editor. It is used to organize both sampled sounds and complex FM for sequential control and sound effect firing. Refer to ST-068 for a more detailed description of its operation. If you are new to patch editing, please read Section 7 and Section 8 of this document--Getting Started.

The TONE folder contains versions 2.07, 2.06 and 1.13E of the **Tone Editor**. For your reference, version 1.13E is still included since many users have found it to be more stable than 2.06.

Wave Editor
A sample editor, which should used as a final step before importing the sample into the Tone Editor. It is suggested that you use a Third party sample editor for more versatility and editing power. Refer to ST-99 for use of this application.
***note: make sure the loop start and end points are set on even numbers. The loop end must be set to the end of the sample even if it is a non-looping sample. ***

Linker
Object oriented DSP programming application. It allows to "click and drag" DSP routines and to direct those DSP algorithms to outputs of your choise. For 3-D sound DSP, like QSound™ or YSound (by Yamaha), you must use eLinker (not included on this disk). Contact your Third Party Account Executive to obtain the eLinker. Please refer to ST-69 for instructions on how to use this application. For more detail specifications on the DSP routines see ST-70.

----

2. CartDev Compatible Sound Tools for the Sega Saturn

11/11/1996

```
*	Sound Driver Version 1.30 and up
*	SoundSimulator Version 2.0 and up
	Program Debugger Version 1.0
*	68000 Assembler Environment 1.14

*	Tone Editor Version 2.0 and up
*	Wave Editor Version 1.13
*	DSP Linker Version 2.0
*	DSP Assembler Version 2.0
*	DSP Parameter Editor Version 2.0
```

All applications marked with an asterisk support the CartDev (the 68000 Assembler Environment 1.14 was modified to support CartDev since it is a target independent tool). These applications check the SCSI bus at startup to determine whether a Sound Box or a CartDev is connected to the Macintosh.

The program debugger (for the 68000) nor the Wave Editor support CartDev due to hardware limitations of the DevSaturn (modified Saturn). While editing functions such as cut and paste of waveforms are possible on the Wave Editor, digital input-based sampling or playback is not possible.

Important Notes

Two files, **sound.coff** and **satmon.bin**, need to be placed in the same folder as the SoundSimulator for CartDev support. Allow up to 20 seconds for SoundSimulator to start up.

NOTE: The sound RAM size on the DevSaturn is 512KB (080000H). While the Sound Box is capable of handling sound data up to 1024KB (0100000H), it is not possible to download or play back sound data over 512KB on the DevSaturn. In the SOUND\TOOL DEMOS\CARTDEV folder is a demo that is compatible with the MIDI Cartdev.

It is necessary to have a DevSaturn equipped with a special MIDI interface module for CartDev and DevSaturn MIDI-based sound development. Only the SoundSimulator can be used for the playback of project files if the interface module is not connected to the DevSaturn. Contact Cross Products Ltd. or Sega Technical Support in your area for more details on the MIDI interface module.

The CartDev compatible tools (e.g., SoundSimulator, Tone Editor, and DSP Linker) are known to crash whenever high volume MIDI data input and CartDev<->Macintosh SCSI access occur at the same time. This problem also occurs with the Sophia Sound Box to a lesser extent. To solve this problem, avoid MIDI data input to the target while SCSI access occurs between the Macintosh and CartDev.

----

3. Conversion Tools

AIFF to RED Book(MAC)
This tool coverts AIFF sound files to Red book audio by stripping the header. It takes a file that contains audio data (in 16bit stereo at 44.1kHz) and creates a file whose data fork contains that audio data in raw Red Book format.

ADPCM Encoder Version 1.00E
The ADPCM Encoder is a tool for producing ADPCM data that is compatible with the Sega Basic Library's (SBL) ADPCM Playback Library. The encoder is capable of converting 8, 16, or 24 bit mono/stereo AIFF or Sound Designer II format files.

----

4. Tool Demos

You can quickly test your Saturn Sound Box by using this Tool Demo. Please refer to the Getting Started ReadMe. Also included is a demo that is compatible with the MIDI Cartdev.

----

5. Code Demos

This is a collection of example code demos for the programming box. The four directories contain:

SEQ directory - starts a MIDI sequence
ONESHOT directory - fires sound effects using MIDI direct command
PCM directory - plays a PCM file
CDDAPLAY directory - Plays and audio track of a CD.

----

6. New General MIDI and FM Sound Tone Banks

This is a collection of General MIDI and FM Sound Tone Banks ready to use with the Sound Tools.

IMPORTANT: Do NOT use the previously released InVision Tone Banks. If you have any questions regarding this, please contact SEGA DTS by emailing dts@sega.com.

----

7. Generation of sound data without use of Saturn Sound Box

Author: Thomas Szirtes / Sega Europe

Basic Principles:
=================

For one shot sound effects. What you need :
1. A Macintosh
2. Sega Sound Tools (on DTS disk) namely Tone Editor.
3. Sound Driver (SDDRVS.TSK)

- **To generate and playback:**

1. A map file (binary version) - contains information on where data will be stored in Sound RAM.
2. One (or more) tone file(s) (binary version) - contains the sample data together with information on how to play them back.

- **Structure of tone banks:**

They are divided into Voices and Layers. A Voice contains one or more Layers, each Layer contains a sample which is playable over a certain pitch range, has its own volume, pan and dsp effect. Therefore a Voice can contain several samples (Layers). Voices are switched between by sending "Program Change" Commands.

You can have (if you want) only one voice, and assign a seperate sample to a individual note value within that, therefore you do not have to send any Program Change commands.

- **Why you want a sound box:**

You cannot preview what your samples are going to sound like until you have incorporated the data into your code. There are two systems - the old large Sega

Sound Box (quite rare to get of now - but very nice), or the new Cross Products Midi Modified Saturns (not as nice, but cheaper and smaller).

You cannot run the Sound Simulator, or generate DSP .EXB files, you have to generate the map files by hand which is not very difficult.

Using Tone Editor 2.06:
=======================

In this example we will generate a very simple tone bank containing 1 voice, that in it contains 1 layer with a non-looping sample that triggers on only one note.

1. Load up Tone Editor
2. Click OK on "Target Board Not Connect" (or any other dialogue box for that matter)
3. Click Cancel on File Requester that appears.
4. Select File : New menu
5. Requester appears asking for number of Voices. So enter the number of voices you want in the tone bank. For this example we will enter 1.
6. Window called "untitled" will appear. With No., Voice Name, Bend Range, Portamento, Vol Bias, Play Mode and Size. Ignore pratically all of these.
Enter a name in the Voice Name area (only for your own reference) say "Voice 1"
7. Double Click on "0" of No. area.
8. New requester appears asking for the number of Layers to be inside this voice. For this example enter 1.
9. New window opens called "Voice 1" containing No., Layer Name, Start, End, Direct Level, Effect Send etc...
Lets name our new Layer "Layer 1" (in the same way as our Voice)
10. Click on Start Box, here is the lowest note value that our layer will extend to. We will enter 60 (C4 - ie. Note C in octave 4).
11. Click on End Box, here is the highest note value that our layer will extend to. We will enter 60 again, ie. this sample will only sound (be valid) when note value 60 (C4) is sent on our voice.
12. Drag our Direct Level slider to 7 (max). This is the amount of sample signal that is routed straight to the output. Effect Send is the amount of sample signal that is routed via the DSP (more specifically the via the DSP input specified by Effect Select) - however this we will leave at 0. Direct Pan we leave.
13. Double Click on No. "0" box. Up appears a diagram - this is how our sample is processed.
14. Double Click on the empty rectangle box in the center of the window. A file requestor should appear.
15. Select the .AIF sample you want. Information on it will appear (take note of the sampling rate). Click OK.
16. The sample name will now appear in the rectangle. Yeah!!
17. Double Click on the Upside Triangle (the Inverter Symbol).

Here we can change the LOOP SELECT (either off for no loops, or forward for looping + others). We will leave this OFF.
We can also change the Base Note. This affects the playback rate that our sample is played at. A Base Note of 60 means that if a 44Khz Sample is triggered on note value 60 it is played back at 44Khz. Therefore, if we trigger a 44Khz on note value 61 (cause another sample is assigned already to note 60) we want to change this to base note 61 in order to still playback at 44Khz. Also if our sample is not recorded at 44Khz, ie. 22Khz we want to adjust this so that a note assigned to note 60 for example will be played back at 22Khz and not 44Khz then we change this to 72 (84 for 11Khz, and 96 for 8Khz).
NB >>>> The reason for this is due to sound theory - ie. in order to raise the pitch of a sound by one octave (12 tones) we need to double the playback frequency. Ie. from 44Khz to 88Khz, we need to raise the pitch by 1 octave.
Therefore to compensate for a 22Khz sample we also need to raise the base note 12 tones (1 octave).
18. Close the window after adjusting the base note.
19. Double click on Total Level box. But DON'T RUSH AND TOUCH ANYTHING YET in this window yet - an important note - the Total slider is there to fool you. Total Level actually = Attentuation. Ie. Total level of 0 means max volume, Total level 127 = no sound. Also of interest here is the SDIR - again don't touch. But if you are in trouble, checking this box will send all sample sound direct to output bypassing all envelope generators and effects, therefore, this is useful for debugging your banks if you cannot hear anything.
Velocity controls the ratio of velocity value to actual volume. We will leave this at "untitled" for now. This is a flat velocity curve (ie. it makes no difference) - if you want you can edit one to have a linear proportional one, but you need to learn to edit these (read the manual).
20. Close the window (you haven't changed anything have you???)
21. Don't touch any other boxes for the moment.
22. Okay Save your file. Also select File : SaveBinFile.

Yeah. We've created a tone bank.

Creating a Map File :
=====================

Reference - page 12, Sound Driver System Interface Manual

Map contains the following information:
1. Data Type - ie. Tone Bank / Sequence / DSP Prog / DSP WorkARea
2. Unique identifier within each data Type.
3. Start Address (actually an offset in sound Ram)
4. Data Size

See reference for bit register assign.

Note :
1. Lowest Start Address = B000H.
2. Highest data address = 7FFFFH.
3. That makes roughly 474K of data space. Remember that sound driver uses area from 0 - B000!
4. Finish map off with 0xffff, 0xffff.

Map for our example :

```
char map[] = { 0x00, 0x00, 0xB0, 0x00, 0x00, 0x0s, 0xss, 0xss, 0xff, 0xff, 0xff, 0xff };
               <>          <------>      <----------->    <------------------>
               \Tone Data   \ Start       \ Size of         \ Map Terminator
               #0 (first)    Address      Tone Bank
```

Triggering it from Code :
=========================

Principles :

1. MIDI CHANNEL - There are 32 logical channels, each channel can be configured to play back separate voices with its own volume/pan/pitch bend settings, and can be reconfigured at any point. Each channel can play many notes as long as they are of the same voice / configuration

2. SCSP SLOTS - you don't have to worry about these. These are the actually physical slots that the sound chip has, each one can generate one sound at a time - there are 32 of these. These slots are assigned by the sound driver you have very little control over this.

Procedure :

1. Load sound data into HI-RAM + sound driver.
2. Initialize sound driver with sound map. (see below).
3. Change Map (see below)
4. Transfer sound data items to sound ram using SND library (see below)
5. For each midi channel you use, you need to : (see below)
Send a Tone Bank Select - this selects the tone bank that will be active on that channel
Send a Program Change - this selects which voice within that bank will be active on that channel
Send a Note On - Triggers the desired note on that voice. (Send a Note Off - switches off that note)
6. More advanced stuff like:
Pitch Bend - (midi command 6, data 1 - pitch up, data2 - pitch down)
Volume Change - (midi command 3, data 1 - 10, data 2 - volume)

Use the following code as a template (based on oneshot.c on DTS CD):

```
void sound(void)
{
   SndIniDt snd_init;

        /* External Reference to our data files in memory (main) */
   extern char          sddrvs[];
   extern char          sddrvs_end[];
   extern char          scsp_tone_bank[];
   extern char          scsp_tone_bank_end[];
   extern char scsp_map[];
   extern char scsp_map_end[];

/*
 *  process 1: initialize the sound driver (version 2.04 here)
 */

   SND_INI_PRG_ADR(snd_init) = (Uint16 *)sddrvs;              /* 68k snd drv
program */
   SND_INI_PRG_SZ(snd_init) = (Uint16 )(sddrvs_end - sddrvs);  /* sound driver
size */

   SND_INI_ARA_ADR(snd_init) = (Uint16 *)scsp_map;          /* sound area map
*/
   SND_INI_ARA_SZ(snd_init) = (Uint16)(scsp_map_end - scsp_map); /* snd area
map size */

        /* Initialize - this transfers map data from main to sound ram */
   SND_Init(&snd_init);

   SND_ChgMap(0);        /* change sound area map; start with map 0 */

        /* For each data item in your map transfer it to sound ram */
   SND_MoveData((Uint16 *)scsp_tone_bank,         /* source data address in HI-
ram */
                       (Uint32)(scsp_tone_bank_end - scsp_tone_bank),  /* source
data size */
                       SND_KD_TONE,                          /* type of data */
                       0);                           /* identifier (ie. as specified in map) */

   SND_ChgMix(0,0);               /* change mixer - default */

   SND_SetTlVl(15);               /* max volume */
/* Change tone bank 0 select */
    SND_CtrlDirMidi(0x02, 0x00,         /* sound control and priority number - not
important now*/
                                0x03,               /* midi command = control change */
                                0x01,               /* midi channel =1 */
                                0x20,               /* data 1 = "tone bank change" */
                                0x00);              /* data 2 = tone bank number */

/* Program change = 1 */
   SND_CtrlDirMidi(0x02, 0x00, /* as above */
                               0x04,                /* midi command = program change
*/
                               0x01,                /* midi channel = 1 */
                               0x01,                /* data 1 = program number (ie. voice
number) = 1 */
                               0x00);               /* data 2 = not used = 0 */

/* Play a simple note */
   SND_CtrlDirMidi(0x02, 0x00, /* as above */
                               0x01,                /* midi command = note on */
                               0x01,                /* midi channel = 1 */
                               60,                  /* data 1 = note value (pitch)=60 */
                               127);                /* data 2 = velocity =127 */

   while(1);
}
```

8. Getting Started

General Description

The Sound Simulator is an application which allows you to allocate sound RAM into individual "slots" in the form of a memory map. Each of these slots contains a certain type of data. The types are:

BANK	Data binary files for tone banks
SEQUENCE	Data converted MIDI sequences
DSP Program	program data for the DSP Linker
DSP Work RAM	buffer area to hold data to be processed by DSP algorithms

The Sound Simulator must be run first and a memory map loaded (or created) before any of the other applications can be used. It is also where you convert MIDI files into sequence data which the Saturn can read, and arrange it into sequence banks. After this has been done, you can play the sequences back with any DSP to hear how it will sound in the Saturn.

- **Loading a map and running a demo sequence:**

The folder containing the SndSim application should also have the following files:

```
CHOICE MAP
CHOICE MAP.bin
NEW CHOICE TONE
NEW CHOICE TONE.bin
CHOICE TUNE more.EXB
CHOICE TUNE more.EXL
CHOICE TUNE more.LRI
CHOICE TUNE more.YLI
CHOICE COLLECT
```

Make sure these files are in the SndSim's folder before proceeding, because the above files are located in the SOUND\TOOL DEMOS\CHOICE directory.

- **System Startup:**

1. Double click on the Saturn SndSim-E1.28 icon.
   A blank map should appear.
2. Select OPEN from the File menu.
3. Choose DEMO MAP, click on OPEN.
   If this is the first time you are opening the file, read the following.
   Otherwise, go on to 4.

You will get a series of dialog boxes indicating that certain files can't be found. Don't worry, just keep clicking OK until they go away. The reason this happens is because a Sound Simulator map stores the paths of the files it contains. Whenever the files are moved, they must be re-loaded individually. Then the map can be saved again along with the new paths.

At this point the blank map should be replaced by the new map. Each line in the table indicates an area in memory used by a file or DSP work RAM. The area is shown by the start - end addresses in the third column. (These are in hex) The type of data is shown in the fifth column, and the file name (if any) is shown in the sixth. At this point, memory has been allocated for these files but they haven't been loaded.

Double click on the line for the file CHOICE COLLECT. You will get a dialog box with the name of the currently selected file, along with the start and end addresses currently allocated. Click once on the oval containing the name of the file. A file selection box will appear open to the folder that the SndSim application is in. Find the file CHOICE COLLECT and double click on it's name. The file selection box will go away, leaving you with the dialog box. Nothing has changed in the box, but by re-loading the file you have established it's new path. Close the dialog box, highlight the same line in the map, go to the File menu and choose DOWNLOAD. After a few seconds an X should appear in the first column. This means the file is now loaded. Repeat the steps in this paragraph for the other files in the map. (NEW CHOICE TONE.bin and CHOICE TUNE more.EXB) Don't do anything with the space for DSP work RAM. Choose SAVE from the File menu. The map is now saved with it's new paths. From now on when you open it, the files will automatically be loaded.

4. Select DOWNLOAD SYSTEM from the File menu.
5. Select STARTUP SYSTEM from the File menu.
6. Select MAKE CURRENT from the File menu.
7. Select SOUND SIMULATOR from the Option menu.
   The Sound Simulator window should open.
8. Click on the blank box to the right of EFFECT CHANGE.
   A file name with the extension .EXB should appear in the box.
9. Once the name of the .EXB file is in the box, click on EFFECT CHANGE.
10. Click on the blank box to the right of MIXER CHANGE.
    A file name with the extension .bin should appear in the box.
11. Once the name of the .bin file is in the box, click on MIXER CHANGE.
12. Click on SEQUENCE 0 so that it is highlighted.
13. Click on START.
    The sequence should start playing.
14. To stop the sequence, click on STOP while SEQUENCE 0 is highlighted.

Running a MIDI sequencer

At this point you should be able to boot up a MIDI sequencing program and with MIDI thru enabled, play the sounds from the Saturn with your midi controller.

- **The Tone Application**

General Description
This is where you construct voices using AIFF samples, map them to MIDI note numbers, set output levels, set DSP effects sends and returns . It is similar to patch editors for multi-timbral synthesizers.

Getting Started
When you launch the Tone Application the only thing that appears is the Master Volume window. This sets the overall output level for the tone bank. To open up a tone bank, select OPEN from the File menu, and click on a file name to open it. (or select NEW from the file menu to create a tone bank from scratch) If you select a different file from the one loaded in the Sound Simulator, it will become the active tone bank.

Voice Architecture
A tone bank may contain up to 128 different voices numbered 0-127, which correspond to standard MIDI program change numbers. Each of these voices is made up of up to 255 layers. A layer is an AIFF sample, along with key mapping, output and panning information; as well as velocity mapping and modulation data.

Voice Parameters
Voices contain the following parameters:
Number: Standard MIDI program change number (0-127).
Name: Double click on to edit.
Bend Range: Pitch bend range in semitones from center to either extreme; 0-12 = 0-12 semitones 13 = 2 octaves.
Portamento: As of version 1.13, this does not work.
Vol Bias: Determines how the output level of the voice responds to key velocity and volume (ctrl 7) information. A value of 0 can be thought of as "neutral", producing (roughly) a linear response. As Volume Bias is set higher (positive), lower velocities and volumes are output at higher levels. In addition, the maximum output level is reached with lower velocities and volumes. The value may also be set to negative numbers, with the reverse effect.
Play Mode: As of version 1.13, this does not work.

Layers
Double click on the number on the left side of the voice window to open the layer window. If you are creating a new voice, you will be asked how many layers to use. You can always change this number later. Each layer has the following parameters:
Name: Double click on to edit.
Start-End: Two MIDI note numbers to determine the range of the layer. (These may overlap with other layers).
Direct Level: Output level of the direct signal.
Effect Send: Amount of signal sent to a DSP module.
Effect Select: Number of the DSP module.
Direct Pan: Panning position of the direct signal.

In addition to these parameters, each layer has a slot window which contains settings for modulation and envelope data, as well as the area to load the sample itself. Double click on thenumber to the left of the layer name to open the slot window.

Loading an AIFF File
Double click on the rectangle in the middle of the slot window. You can then select an AIFF file to load. Once you have selected a file, it's name should appear in the rectangle. Double click on the triangle below the filename. This is the Inverter window, and is where you can set the tuning of the sample, along with the loop type.

----

9. Notes on Development

Hardware Requirements
- Macintosh IIFX or better
- 16 MB RAM (more is needed to simulate PCM streaming)
- 300 MB Fast Hard Disk (for sample storage)
- MIDI interface
- MIDI Controller
- Saturn Sound Box (Supplied by SEGA)
- SCSI Cable (Supplied by SEGA)
- InLine SCSI terminator

Tools - Supplied by SEGA
- Sound Simulator
- MAP Editor
- Tone Editor
- Wave Editor
- Linker (DSP Editor)
- eLinker (by request) includes 3-D sound modules QSound, Ysound
- SDSS (68000) Assembler (optional DSP programming tool)
- SSBug (the debugging tool that works with SDSS)

Other Tools needed
- Midi Sequencer (MIDI file format 1)
- It is suggested to work with a more sophisticated digital audio editor than the one that is provided.

Specs
- For a more detail description of the specs see ST-081
- 32 note polyphony
- 8 sound control numbers 1 sequence or 127 direct midi commands per control number)
- up to 32 op FM note each carrier or modulator takes up one note of polyphony)
- 4 channels of PCM (although, it hogs a lot of bus time)

Functional Overview
- Two CPU interfaces; 68000 / data transfer, can be operated in parallel
- Built-in MIDI interface (2 x 16 channels)
- 128-step DSP
- Built-in ring filter buffer for filtering
- Memory control functions allow program in sound memory to operate the CPU - 16 channel Digital Mixer
- DMA transfer between SCSP Control Registers + Sound Memory (3812 bytes)

Sound Development Documentation:

Sound Development Manual ver, 1.1	ST-081-R5-062894
Saturn Sound Simulator Manual	ST-168-R3-011895
Wave Editor User's Manual	ST-099-R1-042594
SCSP Waveform Editor Technical Specifications	ST-067-121593
Tone Editor User's Manual	ST-068-R1-042594
Tone Editor User's Manual Addendum: File Format	ST-235-030795
SCSP / DSP Effect Module Specifications	ST-069-121693
DSP Linker User's Manual	ST-070-R1-031094
dAsms User's Manual	ST-228-R1-030595
Parameter Editor User's Manual	ST-227-R1-030595
Saturn Sound Tools Manual Supplement	ST-198-R1-121594
SATURN Sound Driver System Interface Version 3.03	ST-166-R4-012395
Standard MIDI File: Converter Specifications	ST-066-121593
Sound Programming Debugger User's Manual	ST-065-R1-031494
Microcomputing Developing Int. Environment for Macintosh	ST-080-R2-050994
NEW: Saturn Sound Driver Implementation Manual	ST-241-042795

Sound for the Saturn is best dealt with by using the Saturn Custom Sound Processor (SCSP) because it has the versatility of standard midi commands. The tools allow the sound designer to set up the memory map and how it is utilized. It uses Standard MIDI files to trigger AIFF files that have been collected by the Tone Editor. It can also trigger these samples.

For further assistance, contact SEGA DTS by emailing: dts@sega.com