

Chapter 7: Super Game Boy Sound.....	182
1. SGB Sound Program Overview.....	182
2. Memory Mapping (SUPER NES APU)	183
3. Creating and Transferring Score Data	184
3.1 Transferring Score Data	184
3.2 Summary of BGM Flags.....	184
3.3 Overview of Creating Score Data	185
3.4 Setting the NEWS System Working Environment.....	185
3.5 Setting the Working Environment When Using IS-SOUND	189
3.6 Score Data Format When Using Original Tools	190
3.7 Cautions Regarding Production of Musical Pieces.....	202
3.8 Format of Transferred Data.....	203
4. SGB Sound Program Source List	205
5. Transferring Audio Data to the Score Area	208
5.1 Required Data and Procedure for Audio Output.....	208
5.2 Transfer File Example.....	209

CHAPTER 7: SUPER GAME BOY SOUND

1. SGB SOUND PROGRAM OVERVIEW

The SGB sound program is a special SGB program built into the SGB system program. The sound program is automatically transferred to the SNES APU at system startup.

Using the SGB system commands, pre-loaded sound effects in the sound program can be used in Game Boy application programs that support SGB (SGB software).

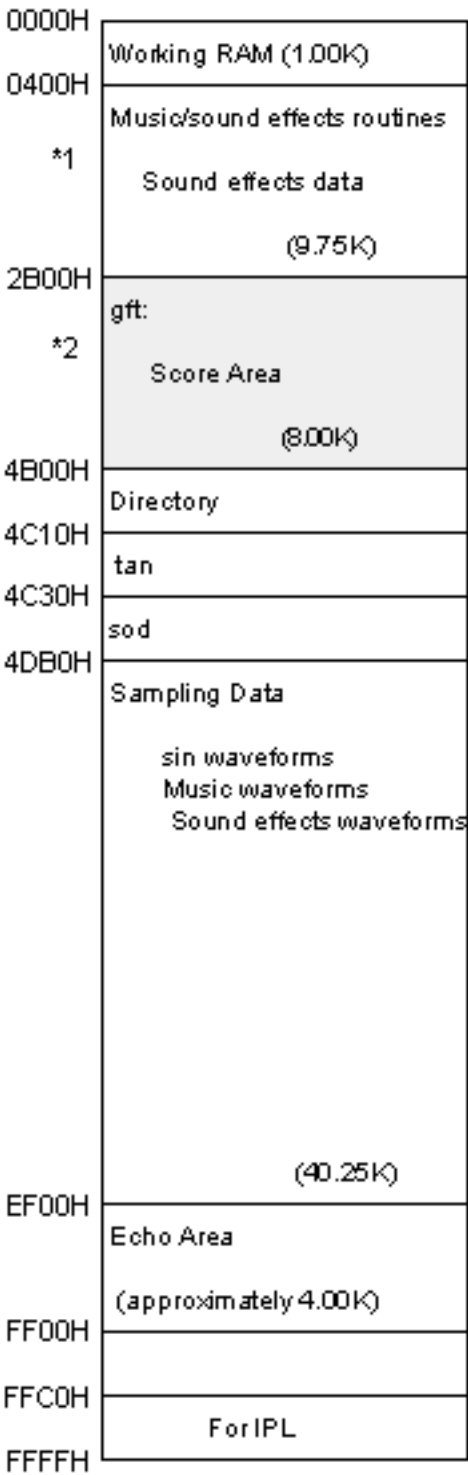
These commands can be used to set each of the 73 types of pre-loaded sound effects to 4 intervals (playback frequencies) and 3 volume levels.

Also preloaded are music data for BGM (instruments sound sampling data). This easily allows play of score data created with Kankichi-kun, the tool for creating SNES scores, and score data for KAN.ASM, the standard driver that is a software tool for IS-SOUND.

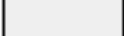
In addition, information on the SGB score data format has been made openly available, allowing those using tools other than the NEWS system or IS-SOUND to create score data in this format.

2. MEMORY MAPPING (SUPER NES APU)

[APU Addresses]



*1 Special SGB Sound Driver

*2  = Transfer area for usable score data

3. CREATING AND TRANSFERRING SCORE DATA

3.1 Transferring Score Data

BGM can be played with the APU by using the SOU_TRN command to transfer original score data to the prescribed area of APU RAM. The user area is the 8 Kbytes from 0x2B00 to 0x4AFF.

3.2 Summary of BGM Flags

SOUND Command	
0x04 d7-d0 [bit]	
Code	Flag Description
0x00	Dummy flag for retriggering
0x10 0xF	
0x80	BGM stop flag
0xFE	Use prohibited (used by system)
0xFF	Use prohibited (used by system)

Note *If 0x01-0x0F are set without score data being transferred, the BGM built into the system is played.*

This BGM is exclusively for use by the system, so 0x01-0x0F should not be written as a BGM flag without original score data being transferred.

Even if original score data is transferred, there is risk that the sound program will run uncontrolled if a non-designated code is written.

Muting is in effect when the system is initialized, so the BGM playback settings must be made after muting is canceled.

3.3 Overview of Creating Score Data

Original BGM can be played with the SGB sound program by transferring score data to the APU using system commands.

Fifty-seven sounds can be used in BGM, and the score data can be up to just under 8 Kbytes in size.

The method used to create a musical piece is nearly identical to that of the standard SNES.

When the NEWS system is used, score data is created using Kankichi-kun. When IS-SOUND is used, score data created by an external sequencer are processed through MIDI and converted to create score data supported by the standard sound driver KAN.ASM.

In addition the SGB score data format has been made openly available, allowing those using original tools to create score data in this format.

In creating musical pieces, please refer to Section 4, *SGB Sound Program Source List*, when selecting sounds. Please do not change the order of these source data.

3.4 Setting the NEWS System Working Environment

Working Environment Settings for the NEWS System

1. Rename the current sobox directory.

```
% mv sobox xxxxx
```

2. Create a new sobox directory.

```
% mkdir sobox
```

* SGB can use only specific sound objects. Thus, special SGB source data must be installed. A sobox directory for SGB use must be created to prevent loss of previously installed source data files with the same names as the data files to be installed.

3. Move to the sobox directory.

4. From the installation disk, install **soread** in this directory.

```
% tar xvf /dev/rfh0a soread
```

5. Next install the sampling data files (xxx.so . . .) in this directory.

```
% soread
```

Executing the above command causes the sampling data to be automatically installed.

6. Create a new SGB working directory at any location.

```
% mkdir #####
```

7. Move to the SGB working directory.

8. From the installation disk, install the following files in the working directory: **sgbt.asm**, **sample.kan**, **check.kan**, **kankichib.hex**, and **kan.equ**.

```
% tar xvf /dev/rfh0a sgbt.asm sample.kan check.kan kankichib.hex kan.equ
```

* The organization and address settings in **kankichib.hex** are as shown below.

* Use the installed **kankichib.hex** file when starting up **mapu**.

Item	Setting
Kan.equ	0x4c30
Kan.tan	0x04c10
Program start address	0x00400
DIR address	0x04b00
Echo end address	0x0ff00
Sound score start address	0x02b00

9. Make the following changes in the file **.cshrc** in the home directory.

--- Following are the Sound Generation Environments Settings ---

	Before Change	After Change
StartOfKan	0x800	0x400
StartOfDirectory	0x3c00	0x4b00
EndOfDirectory	0x3cff	0x4c0f
StartOfAttribute	0x3e00	0x4c30
StartOfTan	0x3f00	0x4c10
StartOfWave	0x4000	0x4db0
EndOfWave	0xcfff	0xeeff
StartOfFumen	0xd000	0x2b00

10. In the home directory, execute the following command: **source .cshrc**.

Cautions When Using Kankichi-kun

1. Copy **sample.kan** to a newly created score data file, **[score_name].kan**.

```
% cp sample.kan xxx.kan
```

* This avoids the task of creating a source list in source-list order when using **mapu**.

2. Start **mapu**.

```
% mapu -k
```

* When starting **mapu** for the first time, press the NICE reset button.

3. The usable sounds (sources) can be checked with **mapu**. Selecting **check.kan** allows the sounds to be checked in source-list order.

* If data in files such as **check.kan** are changed, the sounds cannot be checked.

4. To actually create a tune, select **xxx.kan**.

* Source data (sampling data) that SGB can use have been set in **xxx.kan**. The source list is shown in Section 4, *SGB Sound Program Source List*. Note that changing the order of the source list will result in sounds different from the intended sounds when BGM is played.

5. When producing a musical piece, see Section 3.7, *Cautions Regarding Production of Musical Pieces*. Refer to the Kankichi-kun Manual.

6. Finally, convert to the file format described in Section 3.8, *Format for Transferred Files*.

3.5 Setting the Working Environment Using IS-SOUND

Environment Required

- Hardware: IS-SOUND connected to a host computer
- Software: IS-SOUND software tools (installed)

Revisions

1. Portions of the IS-SOUND software tool KAN.EQU were revised as indicated below (older versions only).

Before Revisions	After Revisions
cut: equ 122+ 0x80	wav: equ 122+ 0x80
fft: equ 123+ 0x80	sel: equ 123+ 0x80
ply: equ 124+ 0x80	cut: equ 124+ 0x80
wav: equ 125+ 0x80	fft: equ 125+ 0x80
sel: equ 126+ 0x80	ply: equ 126+ 0x80

1. Set Gate Table data to 050 · 101 · 127 · 152 · 178 · 203 · 229 · 252.
2. Set Velocity Table data to 025 · 050 · 076 · 101 · 114 · 127 · 140 · 152 · 165 · 178 · 191 · 203 · 216 · 229 · 242 · 252.

Note ***Sound data (sampling data) are required to check music data using IS-SOUND. Consequently, a program equivalent to the sound program built into the SGB hardware (including sound-effect data) and sampling data (sound data) have been provided in a hex file for MS-DOS. The following briefly describes how to set up this program and data.***

Setting the Working Environment

1. Create an SGB working directory at any location, and move to that directory.
2. Copy **sgbsound.hex** from the disk to the working directory.
3. Start the debugger **shvc**.
4. Also start the sound debugger **ssnd**.
5. Execute **r sgbsound.hex** to load **sgbsound.hex**.
6. Execute **g400** to run the sound program.
7. Press the HOME button to switch to shvc mode.
8. Execute **s2140** to write 01 (from the main program, writes 01 to 0 of the sound port).

With this procedure, the pre-loaded source data (sampling data) are played in the order shown in Section 4 of this chapter, *SGB Sound Program Source List*.

After the data is transferred once, only the score data needs to be transferred to allow music to be checked again.

Cautions

1. Score data is the data defined in KAN.ASM Version 1.21 as being located from GFT onward. For information on all items related to converting data from other sequencers to score data, formats, and tool usage, see the IS-SOUND manual.
2. Set the source data number according to the source list.
3. Set the starting address of the score data to 0x2B00.
4. When producing a musical piece, do so in accordance with Section 3.7, *Cautions Regarding Production of Musical Pieces*.
5. Convert to the file format described in Section 3.8, *Format for Transferred Files*.

3.6 Score Data Format When Using Original Tools

The score data format has been made openly available for the benefit of those using original development tools.

Data that is not in this format will not operate on SGB.

Note that in some cases, program control may be lost.

Score Data

Glossary of Terms

gft	Location of tune table definitions (collection of tune label definitions). Up to 15 tunes can be defined. The order defined here corresponds to the flag set for port 0 (0x01-0x0F).
Tune label	A label name applied to each tune.
Block	A unit several bars long that each tune is divided into.
Parts	The channels that make up each block (maximum of 8 parts).
Performance data	The aggregate of the score data played by the parts. The parts in the channels must all be the same length (number of steps) in a given block.

Overall Format of Score Data

Example 1

* Area inside dotted frame = Data table for 1 tune

```

org 02b00H ; (a) Starting address of score data
gft: ; (b) Tune table
dw bgm1, bgm2, ... ; Indicate the tune labels

```

```

bgm1: ; (c) Tune label 1
dw bgm1_block1 ; (d) Block 01
bgm1_0:
dw bgm1_block2 ; (d) Block 02
dw bgm1_block3 ; (d) Block 03
dw 255 ; (e) Repetition code (endless)
dw bgm1_0 ; (e) Repetition starting address
dw 000 ; (f) Tune label end code
;
bgm1_block1: ; (g) Block 01
dw bgm1_block1_0 ; (g) Starting address of Part 0
dw bgm1_block1_1 ; (g) Starting address of Part 1
dw bgm1_block1_2 ; (g) Starting address of Part 2
dw bgm1_block1_3 ; (g) Starting address of Part 3
dw 00 ; (g) Part 4 unused
dw 00 ; (g) Part 5 unused
dw 00 ; (g) Part 6 unused
dw 00 ; (g) Part 7 unused
bgm1_block2:
. . ; (g) Same in Block 2
. .
bgm1_block3:
. . ; (g) Same in Block 3
. .
bgm1_block1_0: ; Block 01 (h) Part 0 performance data
db tp1,049,mv1,200,sno,$1a,pv1,180,pan,010
db ecv,255,040,040,edl,002,090,002,tun,050
db 012,P99+V99,c30,d30,e30,f30,024,g30,kyu
db 00 ; (h) Part end code
bgm1_block1_1: ; (h) Part 1 performance data
db sno,$1b,pv1,140,pan,008,tun,030
db 096,P99+V99,g20
bgm1_block1_2: ; (h) Part 2 performance data
db sno,$1b,pv1,140,pan,008,tun,030
db 096,P99+V99,e20
bgm1_block1_3: ; (h) Part 3 performance data
db sno,$1b,pv1,140,pan,008,tun,030
db 096,P99+V99,c20

```

Continued on next page

```
bgm1_block2_0:      Block 02  (h) Part 0 performance data
                    db  sno,$1a,pv1,200,pan,012,tun,050
                    db  . . . . .
                    db  00
                    .
bgm1_block3_0:      Block03  (h) Part 0 performance data
                    db  sno,$1a,pv1,200,pan,012,tun,050
                    db  . . . . .
                    db  00
                    .
```

Description of Example 1

(a) The score data map to memory addresses 0x2B00-0x4AFF in the APU. If this area is exceeded, a portion of the sound program will be destroyed.

(b) gft: is the starting address of the entire tune table.
dw, bgm1, and bgm2... are the tune labels and the starting addresses of the score data items.

(c) The tune label.
The order in which the blocks are played is defined following the tune label.
The dotted frame encloses the data for one tune, bgm1.

(d) Data for each block.

(e) 0x01-0x7F (01-127) is the number of loops (repetitions); 0x82-0xFF (130-255) is an endless loop. If repetition is not needed, set the end code (0x00) instead of a loop code.

(f) Block definition end code.

(g) Location where the parts of each block are indicated and the part labels are defined.

Defines the part labels for parts 0, 1, 2, . . . 7 in ascending order from top to bottom. 0x00 should be written for unused parts. Even if some parts are unused, always define 8 parts.

(h) The performance data for each part.

Play Data Overview

Parameters such as temp, volume, pan, source number, echo, velocity, interval, and sound length are set here.

For specific descriptions, see Section 3.6.4, *Code Summaries*.

First set are the effects parameters – such as main volume, ramp, and echo – for Part 0 of the first block. Once these are set, they need not be set again (for other blocks or parts) as long as they are not changed.

Next the parameters such as part volume, pan, source number, and tuning are set for each part.

Then the sound length, velocity + gate time, and interval are set in that order. Be careful to ensure that sound length is always set first, followed by velocity + gate item, then the interval.

If the next sound is the same as the previous sound, the sound length, velocity, and gate time need not be set again.

Finally, a data end code of 00 is set for Part 0 of each block.

Settings for parts 1-7 are not required.

The lower parts and blocks are set in the same manner.

Code Summaries

a) Length Data (step time)

This is the length (step time) to the subsequent sound; it corresponds to the length of the sound envelope.

The code corresponding to each sound envelope is shown in the following table. Please use the appropriate code in the settings.

Note Length	Code	Note Length	Code	Note Length	Code
Sixteenth note	6	Dotted eighth note	18	Half note	48
Dotted sixteenth note	9	Quarter note	24	Dotted half note	72
Eighth note	12	Dotted quarter note	36	Whole note	96

Note *For triplets and thirty-second notes, convert using the above values.*

(b) Velocity (volume) + gate time

Velocity expresses the volume as a percentage. Here it can be set to 16 levels using the lower-order 4 bits (d0 – d3).

Gate time expresses as a percentage the length that the sound is actually emitted. It can be set to 8 levels using the higher-order 3 bits (d4 – d6).

Changing these values provides legato and staccato effects.

The following table lists the values defined by the SGB sound driver.

The settings are designated using the codes for the listed velocities (VELOCITY) and gate times (GATE_TIME).

Symbol	Code	d7	d6	d5	d4	d3	d2	d1	d0	Rate
V10	0x00	x	x	x	x	0	0	0		VELOCITY=010%
V20	0x01	0								VELOCITY=020%
V30	0x02	x	x	x	x	0	0	0		VELOCITY=030%
V40	0x03	1								VELOCITY=040%
V45	0x04	x	x	x	x	0	0	1		VELOCITY=045%
V50	0x05	0								VELOCITY=050%
V55	0x06	x	x	x	x	0	0	1		VELOCITY=055%
V60	0x07	1								VELOCITY=060%
V65	0x08	x	x	x	x	0	1	0		VELOCITY=065%
V70	0x09	0								VELOCITY=070%
V75	0x0A	x	x	x	x	0	1	0		VELOCITY=075%
V80	0x0B	1								VELOCITY=080%
V85	0x0C	x	x	x	x	0	1	1		VELOCITY=085%
V90	0x0D	0								VELOCITY=090%
V95	0x0E	x	x	x	x	0	1	1		VELOCITY=095%
V99	0x0F	1								VELOCITY=099%
P20	0x00	x	0	0	0	x	x	x		GATE_TIME=020%
P40	0x10	x								GATE_TIME=040%
P50	0x20	x	0	0	1	x	x	x		GATE_TIME=050%
P60	0x30	x								GATE_TIME=060%
P70	0x40	x	0	1	0	x	x	x		GATE_TIME=070%
P80	0x50	x								GATE_TIME=080%
P90	0x60	x	0	1	1	x	x	x		GATE_TIME=090%
P99	0x70	x								GATE_TIME=099%

Symbol input example: P99+V99 Code input example: 0x70+ 0x0F

*When setting score data using symbols, assemble after defining the equal statement according to the table above.

c) Interval Data

Intervals for 6 octaves can be set here.

Depending on the sound, however, high sounds may not be heard.

The following table shows the correspondence between code settings and intervals. Please refer to this table when setting an interval.

*Interval symbols: 0x01- 0xB50

Codes: 0x81-0xC7 (tie = 0xC8 · rest = 0xC9)

*When score data is set using symbols, assemble after defining the equals statement.

Octave 0		Octave 1		Octave 2		Octave 3	
Interval	Code	Interval	Code	Interval	Code	Interval	Code
		C10	0x8C	C20	0x98	C30	0xA4
C01	0x81	C11	0x8D	C21	0x99	C31	0xA5
D00	0x82	D10	0x8E	D20	0x9A	D30	0xA6
D01	0x83	D11	0x8F	D21	0x9B	D31	0xA7
E00	0x84	E10	0x90	E20	0x9C	E30	0xA8
F00	0x85	F10	0x91	F20	0x9D	F30	0xA9
F01	0x86	F11	0x92	F21	0x9E	F31	0xAA
G00	0x87	G10	0x93	G20	0x9F	G30	0xAB
G01	0x88	G11	0x94	G21	0xA0	G31	0xAC
A00	0x89	A10	0x95	A20	0xA1	A30	0xAD
A01	0x8A	A11	0x96	A21	0xA2	A31	0xAE
B00	0x8B	B10	0x97	B20	0xA3	B30	0xAF

Octave 4		Octave 5		Misc.		
Interval	Code	Interval	Code	Tie	TIE	0xC8
C40	0xB0	C50	0xBC	Rest	KYU	0xC9
C41	0xB1	C51	0xBD			
D40	0xB2	D50	0xBE			
D41	0xB3	D51	0xBF			
E40	0xB4	E50	0xC0			
F40	0xB5	F50	0xC1			
F41	0xB6	F51	0xC2			
G40	0xB7	G50	0xC3			
G41	0xB8	G51	0xC4			
A40	0xB9	A50	0xC5			
A41	0xBA	A51	0xC6			
B40	0xBB	B50	0xC7			

Note 1: A value of 1 in the right-most position of the interval symbol indicates a \sharp .

A \flat is represented as the \sharp of one interval lower.

Example: C01=C \sharp for interval 0.

Note 2: When specifying a tie, first set the step time (length) and velocity + gate time. (This can be skipped if unchanged from the previous sound.)

A tie cannot be used at the start of a block.

Note 3: When specifying KYU (a rest), first set the step time (length). (This can be skipped if unchanged from the previous sound.)

Settings Example:

	<u>Length</u>	<u>Gt & Vel</u>	<u>Interval</u>	<u>Code</u>
db	024,	P99+V99,	C30	;(0x0A4) for specifying an interval
	<u>Length</u>	<u>Gt & Vel</u>	<u>Tie</u>	<u>Code</u>
db	048,	P90+V95,	TIE	;(0x0C8) for specifying a tie
	<u>Length</u>	<u>Rest</u>		<u>Code</u>
db	096	KYU		;(0x0C9) for specifying a rest

d) Special Symbols

The special symbols represent special data for implementing a variety of special effects. These include sound change, crescendo, panpot change, vibrato, tremolo, and echo. Each symbol has its own parameters.

The following table lists these special symbols, their parameters, and the valid values for these parameters.

☆ Special Symbols

Summary No. 1

Symbol Code	First Argument (range)	Second Argument(range)	Third Argument (range)	Function
sno (\$E0)	SOURCE NAME $0 \leq X \leq 127$			Sound change
pan (\$E1)	Pan value $0 \leq X \leq 20$			Panpot (0=L/20=R/10=C) (10 = default)
pam (\$E2)	No. of steps $1 \leq X \leq 255$	Pan value $0 \leq Y \leq 20$		Move panpot (Y takes effect after X steps)
vib (\$E3)	No. of hold steps $0 \leq X \leq 255$	Rate $1 \leq Y \leq 255$	Depth $1 \leq Z \leq 255$	Vibrato (no. of hold steps is the time till vibrato takes effect)
vof (\$E4)				Vibrato off
mv1 (\$E5)	Volume $0 \leq X \leq 255$			Main volume (192 = Default value)
mv2 (\$E6)	No. of Steps $1 \leq X \leq 255$	Volume $0 \leq Y \leq 255$		Move main volume (used for crescendo/decrescendo) (Y takes effect after X steps)
tp1 (\$E7)	Rate $1 \leq X \leq 82$			Tempo See Note 1.
tp2 (\$E8)	No. of steps $1 \leq X \leq 255$	Rate $1 \leq Y \leq 82$		Move tempo (Used for retardando/accelerando) (Y takes effect after X steps)
ktp (\$E9)	Transposition level $\$E8 \leq F \leq \FF (- value) $\$00 \leq X \leq \18			Main key transpose (1= semitone up/-1= semitone down) – is the two's complement
ptp (\$EA)	Transposition level $\$E8 \leq X \leq \FF (- value) $\$00 \leq X \leq \18			Part key transpose (1= semitone up/-1= semitone down) – is the two's complement

☆ Special Symbols

Summary No. 2

Symbol Code	First Argument (range)	Second Argument (range)	Third Argument (range)	Function
tre (\$EB)	No. of hold steps $0 \leq X \leq 255$	Rate $1 \leq Y \leq 255$	Depth $1 \leq Z \leq 255$	Tremelo (no. of hold steps is the time till tremelo takes effect)
tof (\$EC)				Tremelo off
pv1 (\$ED)	Volume $0 \leq X \leq 255$			Part volume (192=Default value)
pv2 (\$EE)	No. of steps $1 \leq X \leq 255$	Volume $0 \leq Y \leq 255$		Move part volume (Used for crescendo/decrescendo) (Y takes effect after X steps.)
pat (\$EF)	PAT ADRS(L) $\$00 \leq X \leq \FF	PAT ADRS(H) $\$00 \leq Y \leq \FF	REPEAT PAT $1 \leq Z \leq 255$	Pattern data subroutine Seen Note 2.
vch (\$F0)	No. of steps $1 \leq X \leq 255$			Vibrato deepens gradually over X number of steps
swk (\$F1)	No. of hold steps $0 \leq X \leq 255$	No. of steps $1 \leq Y \leq 255$	Amount of change $\$DC \leq Z \leq \FF (– value) $\$00 \leq Z \leq \24	Start sweep from next sound – is the two's complement
sws (\$F2)	No. of hold steps $0 \leq X \leq 255$	No. of steps $1 \leq Y \leq 255$	Amount of change $\$DC \leq Z \leq \FF (– value) $\$00 \leq Z \leq \24	Start sweep heading into next sound – is the two's complement
sof (\$F3)				Sweep off
tun (\$F4)	Amount of change $0 \leq X \leq 255$			Tune (Semitone up with 255)
ecv (\$F5)	ECHO CHANNEL $0 \leq X \leq 255$	ECHO-VOL(L) $0 \leq Y \leq 255$	ECHO-VOL(R) $0 \leq Z \leq 255$	Echo volume Seen Note 3.
eof (\$F6)				Echo off
edl (\$F7)	ECHO TIME $1 \leq X \leq 15$	FEED BACK $\$9D \leq Y \leq \FF (– value) $\$00 \leq Y \leq \$7F$	FILTER No. $0 \leq Z \leq 10$	Echo delay See Note 4. – is the two's complement
ev2 (\$F8)	No. of steps $1 \leq X \leq 255$	ECHO-VOL(L) $0 \leq Y \leq 255$	ECHO-VOL(R) $0 \leq Z \leq 255$	Move echo volume (YZ values take effect after X steps)

Symbol Code	First Argument (range)	Second Argument (range)	Third Argument (range)	Function
swp (\$F9)	No. of hold steps $0 \leq X \leq 255$	No. of steps $1 \leq Y \leq 255$	SWEEP value interval	Sweep (once) The interval takes effect after the specified number of hold steps.

Note 1: The tempo values set by the program data and the actual (musical piece) tempos that correspond to those values are as follows.

Please refer to this table to make the conversions.

Music Tempo	Driver Tempo	Music Tempo	Driver Tempo
Quarter note = 400	82	Quarter note = 120	25
Quarter note = 30	62	Quarter note = 60	12
Quarter note = 24	49	Quarter note = 30	6

Note 2: Used when the same performance data is repeated (for data compression). Following the pat code, the L and H addresses and the repetition frequency for the performance data is set. The performance data at the addresses specified by pat are then read. The data is played the number of times specified by the repetition frequency. The performance data at the locations specified by pat require an end code of 0x00.

Note 3: When applying echo, **ecv** and **edl** are required. The value entered for the echo channel is 1 for echo used in Part 0, 2 for Part 1, 4 for Part 2, 8 for Part 3, 16 for Part 4, 32 for Part 5, 64 for Part 6, and 128 for Part 7. When echo is used for multiple parts, enter the sum of the channel number values.

Examples

When echo is used for parts 0 and 1, the value entered is 3.

When echo is used for all parts, the value entered is 255.

Note 4: Echo time is the delay duration. It uses RAM area equal to twice the echo time value, expressed in Kbytes. The echo area in SGB is 4 Kbytes, so a value of 2 or less should be entered. Feedback indicates the amount of delay returned. Filter No. indicates the type of filter applied to the delayed sound.

0 = no filter; 1 = high-pass filter; 2 = low-pass filter; 3 = band-pass filter

*The symbols marked with a ☆ in the *Special Symbols* table are applied to all parts. These should be set in the first part.

*When using a symbol to set a special symbol for score data, assemble after defining the equals statement according to the *Special Symbols* table.

*The special symbols and the arguments that follow should be set in the order shown in the tables.

*If using IS-SOUND, load **sgbsound.hex** according to the steps in Section 3.5, *Setting the Working Environment for IS-SOUND*. Transferring the subsequently created score data allows the tunes and sounds to be checked.

Cautions

1. The starting address for score data should be set to 0x2B00.
2. Source numbers should be set according to the source list.
3. Musical pieces should be produced according to the instructions in Section 3.7, *Cautions Regarding Production of Musical Pieces*.
4. Convert to the file format described in Section 3.8, *Format for Transferred Files*.

Summary of Play Data Codes

0x00	Part end code
0x10-0x7F	Note/rest length data & VELOCITY (volume) + GATE_TIME
0x80-0xC7	Interval (sound length) data (C00-B50) * C01-B50 in SGB
0xC8	Tie (TIE)
0xC9	Rest (KYU)
0xCA-0xDF	Use prohibited
0xE0-0xF9	Special symbols
0xFA-0xFF	Use prohibited

3.7 Cautions Regarding Production of Musical Pieces

The echo parameters set in BGM are applied in the same manner for the A and B sound effects. This is because echo is applied equally to all 8 channels. The parameters have been tuned so that they can also be used with BGM, so please note this when resetting the parameters.

Score Data Settings			
Special Symbol	Echo Channel	Echo Volume L	Echo Volume R
ecv	<u>000</u> (Note 1)	40	40
Special Symbol	Echo Time	Feed Back	Filter No.
edl	<u>2</u> (Note 2)	90	2

If echo is not used, specify **e o f** (special symbol) instead of **e c v**.

If a value greater than 2 is specified for **Echo Time**, the sampling data will be destroyed. Up to 15 tunes can be registered (0x01-0x0F). Channels 2 and 3 are allocated for BGM, so these channels should be used for regular playback of BGM parts.

Microtuning of source data used for notes should be specified using the **tun** code with the score data. For tuning values, refer to the recommended tunings in Section 4 of this chapter, *SGB Sound Program Source List* (except for percussion instruments).

The recommended tuning values for this source list are based on an interval of C30 (See Section 3.6.4, *Interval Data*).

Also indicated for each source data item is the score data setting (interval code) for producing sounds with a C30 interval. Please refer to these settings in inputting score data.

In high and low areas, the tuning of some source data may be somewhat off. Whenever this occurs, the tuning value must be modified.

For SGB, all tunings are set 50 cents higher than the standard value (A = 440 Hz).

3.8 Format of Transferred Data

When Using NEWS

1. Copy **s g b t. a s m** to a new transfer file, *filename.asm*.

```
% cp sgbt.asm yyy.asm
```

* When making transfer files, create them based on **sgbt.asm**.

2. Open **yyy.asm** and modify it as follows.

Line No.	Before Changed	After Changed
113	gft : 02b00H	gft : yyy\$,
115	; include xxx.dat	; include yyy.dat

* When adding multiple tunes, add them beginning from line 113. Also increase the number of 'include OOO.dat' statements after line 115 by the number of tunes.

3. Execute the following command: `asm700 yyy.asm`.

The above completes creation of the **yyy.hex** transfer file.

4. Convert the **yyy.hex** file completed in Step 3 to the format used by the SNES sound generator.

Converting to binary data:

```
% cat | h2b -start 400 -b > yyy.bin
```

Converting to hexadecimal data:

```
% cat | h2b -start 400 > OOO.asm
```

When Using IS-SOUND or Original Tools

The score data file to be transferred is converted to the format used by the sound boot program.

Example:

```
dw $0030 ; Number of data items to transfer
dw $2b00 ; Transfer destination address
db $00,$01,$02,$03,$04,$05,$06,$07 ; Score data
db $08,$09,$0a,$0b,$0c,$0d,$0e,$0f ; Score data
db $00,$01,$02,$03,$04,$05,$06,$07 ; Score data
db $08,$09,$0a,$0b,$0c,$0d,$0e,$0f ; Score data
db $00,$01,$02,$03,$04,$05,$06,$07 ; Score data
db $00,$01,$02,$03,$04,$05,$06,$07 ; Score data
dw $0000 ; Transfer end code
dw $0400 ; Program start address
```

The number of data items to transfer (2 bytes) and the transfer destination address (2 bytes) are placed at the starting address of the score data. (Be careful to ensure that the data is in this order.) Finally, the transfer end code (2 bytes) and the program starting address are added. (Be careful to ensure that the data is in this order.) The transfer end code is \$0000.

Cautions Regarding Data Transfer

In SGB, the transfer destination address is \$2b00, and the program starting address is \$0400. Please be sure to use the correct addresses, or program control will be lost.

The area used for the transferred score data is approximately 8 Kbytes. A data overflow will destroy the directory.

If the data exceed 4 Kbytes, divide them into 2 files.

Transfer of score data is completely executed using system commands.

4. SGB SOUND PROGRAM SOURCE LIST

so No.	Kankichi-kun so	so Name	Sound Family	Envelope Type / Specific Sound	Recommended Tuning	Interval
0x000	sn0	+d0.so	Sine Family	Normal envelope		
0x001	sn1	+Dch.so		Envelope with extremely short decay		
0x002	sn2	+d1.so		Electric keyboard envelope		
0x003	sn3	+d2.so		Brass envelope		
0x004	sn4	+d3.so		Pedal organ envelope		
0x005	sn5	+d5.so		Banjo envelope		
0x006	sn6	+d9.so		'Soft' envelope		
0x007	sn7	sin.so		Normal sine wave		
0x008	sn8	+d5.so	Bass Family 1	Banjo envelope	t u n, 0 1 3	
0x009	sn9	+d6.so		Bass envelope	t u n, 0 1 3	
0x00a	s10	+d8.so		Fretless bass envelope	t u n, 0 1 3	
0x00b	s11	B1.so		Bass 1	t u n, 0 1 3	
0x00c	s12	+d5.so	Bass Family 2	Banjo envelope	t u n, 0 2 0	
0x00d	s13	+d6.so		Bass envelope	t u n, 0 2 0	
0x00e	s14	+d9.so		'Soft' envelope	t u n, 0 2 0	
0x00f	s15	B2.so		Bass 2	t u n, 0 2 0	
0x010	s16	+d3.so	Guitar Family	Pedal organ envelope	t u n, 0 4 0	
0x011	s17	+d5.so		Banjo envelope	t u n, 0 4 0	
0x012	s18	+Dch.so		Envelope with extremely short decay	t u n, 0 4 0	
0x013	s19	acg.so		Guitar	t u n, 0 4 0	
0x014	s20	+d1.so	Electric Keyboard Family 1	Electric keyboard envelope		
0x015	s21	+d3.so		Pedal organ envelope		
0x016	s22	ep.so		Electric keyboard 1		
0x017	s23	ep2.so		Electric keyboard 1	t u n, 0 0 3	C 2 0

so No.	Kankichi-kun so No.	so Name	Sound Family	Envelope Type/ Specific Sound	Recommended Tuning	Interval
0x18	s24	+d1.so	Electric Keyboard Family 2	Electric keyboard envelope		
0x019	s25	+d3.so		Pedal organ envelope		
0x01a	s26	epf.so		Electric keyboard, soft type		
0x01b	s27	pipe.so	Organ Family	Pipe organ		
0x01c	s28	+d8.so	Strings Family	Fretless bass envelope	t u n, 0 8 0	C 2 0
0x01d	s29	+d4.so		Strings envelope	t u n, 0 8 0	C 2 0
0x01e	s30	S1.so		Strings	t u n, 0 8 0	C 2 0
0x01f	s31	+d9.so	Chorus Family 1	'Soft' envelope	t u n, 1 7 0	B 0 0
0x020	s32	cho1.so		Chorus 1	t u n, 1 7 0	B 0 0
0x021	s33	+d3.so	Chorus Family 2	Pedal organ envelope	t u n, 1 6 5	B 1 0
0x022	s34	cho2.so		Chorus 2	t u n, 1 6 5	B 1 0
0x023	s35	+Dch.so	Xylophone Family	Xylophone	t u n, 0 5 5	
0x024	s36	+d1.so		Electric keyboard envelope	t u n, 0 5 5	
0x025	s37	+d9.so		'Soft' envelope	t u n, 0 5 5	
0x026	s38	Dxlp.so		Xylophone + looping sound	t u n, 0 5 5	
0x027	s39	+d1.so	Brass Family 1	Electric keyboard envelope		
0x028	s40	brs.so		Brass 1		
0x029	s41	brs8.so	Brass Family 2	Brass 2	t u n, 0 2 0	C 2 0
0x02a	s42	+Dch.so	Trumpet Family	Envelope with extremely short decay	t u n, 0 4 0	C 2 0
0x02b	s43	+d5.so		Banjo envelope	t u n, 0 4 0	C 2 0
0x02c	s44	+d9.so		'Soft' envelope	t u n, 0 4 0	C 2 0
0x02d	s45	tp3.so		Trumpet	t u n, 0 4 0	C 2 0
0x02e	s46	+d4.so	Bassoon Family	Strings envelope		
0x02f	s47	fg.so		Bassoon		
0x030	s48	fl.so	Flute Family	Flute	t u n, 0 5 3	C 2 0

so No.	Kankichi-kun	so Name	Sound Family	Envelope Type Specific Sounds	Recommended Tuning	Interval
0x031	s49	Db.so	Percussion Instrument Family	Bass drum		
0x032	s50	+Dch.so		Closed high-hat		
0x033	s51	Doh.so		Open high-hat		
0x034	s52	sdr3.so		Snare 1		
0x035	s53	Ds.so	Percussion Family	Snare 2		
0x036	s54	Dt.so		Tom (for stepping down)	t u n, 0 1 0	
0x037	s55	clp.so	SE Family	Hand clap		
0x038	s56	jet2.so	SE Family	Jet		
* The following (0x39-0x3E) can be used with Kankichi-kun.						
0x039		jet1.so	Jet			
0x03a		noiz.so	Noise			
0x03b		glas.so	Glass breaking			
0x03c		shot.so	Shot			
0x03d		river.so	River flowing			
0x03e		wind.so	Wind blowing			
		_____	_____			

Settings for source data numbers 0x39-0x3E cannot be specified on Kankichi-kun. These source data can be used only with sound effects. However, they can be set using tools other than Kankichi-kun.

The shaded portions are the basic source data. The other source data items are the basic source data with modified envelopes.

The contents of the source list are also listed in the README file located in the **sobox** directory installed for NEWS.

The recommended tuning values in the source list are based on an interval of C30. (See Section 3.6.4, *Interval Data*.) With high- and low-pass filtering, the tuning of some source data may be somewhat off. Whenever this occurs, the tuning value must be modified.

The interval value is the score data setting (interval code) for producing sounds with a C30 interval. For SGB, all tunings are set 50 cents higher than the standard value (A = 440 Hz). The source data items in the empty areas do not require tuning. (In addition, they can be used without changing the interval).

5. TRANSFERRING AUDIO DATA TO THE SCORE AREA

In general, the score area (8 K) is provided for transferring only score data. However, audio data also can be transferred for output. Audio data can be transferred only if the following conditions are met.

- The data must not exceed the score area (8 K).
- The data is not transferred to areas other than the score area (except for the Directory and sod data).

If the data is transferred to other areas, the sound effects used by the system may no longer play or may be altered (strange sounds). Transferring data to other areas may also lead to a loss of program control. Therefore, please be certain to ensure that the above two conditions are met.

5.1 Required Data and Procedure for Audio Output

1. Sampling data (multiple data items permitted)
2. Score data (score used to play sampling data)

* 1 and 2 combined must occupy less than 8 Kbytes.

* The sound numbers (so No.) corresponding to the sampling data should be from among one of the following.

002H,003H,004H & 00CH,00DH,00EH & 02AH,02BH,02CH (hex No.)

Note *All numbers other than the above are used for system sound effects or music. Therefore, be careful to use only the above numbers.*

3. Directory and sod data corresponding to the sampling data:

* Directory and sod data are provided for each sound (so No.).

	Start Address	Data Structure	No. of Bytes
Directory	0x4B00	Source start address (L)/(H) · Source loop(end)address(L)/(H)	4 bytes
sod	0x4C30	so No./ adsr(1)/adsr(2)/gain/blk No.(2byte)	6 bytes

When the sound number is 0x000, the directory data comprise 4 bytes beginning at 0x4B00, and the sod data comprise 6 bytes beginning at 0x4C30 (0x000 cannot be used).

Please substitute the directory data and sod data values corresponding to the given sound number.

Note *For the sound number, however, be careful not to use any number other those shown in 2. Use of an incorrect number will cause a loss of program control.*

Transferring all of these data and issuing a BGB request will result in audio playback.

5.2 Transfer File Example

With sampling data consisting of a single sound with a sound number of 0x002, the **Directory** data would be the 4 bytes beginning at 0x4B08, and the **sod** data would occupy the 6 bytes beginning at 0x4C3C. In this case, ensure that the score data begin at 0x2B00. Starting these data at any location other than 0x2B00 would cause a loss of program control. The sampling data (audio data) should be transferred to the area between 0x2B00 and 0x3AFF.

```

dw      $0004                      ; No. of data items to transfer for Directory
dw      $4B08                      ; Directory transfer destination address
db      $00,$30,$3F,$30            ; Directory data (4 bytes)
;
dw      $0006                      ; No. of data items to transfer for sod
dw      $4C3C                      ; Sod transfer destination address
db      $02,$FF,$E0,$B8,$02,$B0    ; Sod data (6 bytes)
;
dw      $0020                      ; No. of score data items to transfer
dw      $2B00                      ; Score data transfer destination address
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Score data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Score data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Score data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Score data
;
dw      $0040                      ; No. of sampling data items to transfer
dw      $3000                      ; Sampling data transfer destination address
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
db      $00,$01,$02,$03,$04,$05,&06,$07 ; Sampling data
;
dw      $0000                      ; Transfer end code
dw      $0400                      ; Program start address

```

When using multiple sampling data items, also transfer the **Directory** and **sod** data specified for each item in Step 2.

Note *Be careful not to rewrite the **Directory** and **sod** data used by the system.*

THIS PAGE WAS INTENTIONALLY LEFT BLANK