

1
2
3 SECTION VII
4 SOUND GENERATION SOFTWARE
5

6 The OS provides sound generation routines that output frequency,
7 attenuation and control data to the TI SN76489 sound generator
8 controller. The "sound" described in this section can be repre-
9 sented as both music or noise.

0 There is at least one ten-byte block of CRAM called a "Sound Data
11 Area" reserved for each sound channel. This area contains a
12 record of the current values "playing" on that sound channel.
13 These values are the timing and descriptive information which
14 generate musical notes that are originally stored in cartridge
15 ROM. In total, there should be a minimum of four sound data
16 areas reserved by the user, one for each channel. More data
17 areas are needed if there are sounds to be played concurrently.
18 For an average video game, seven is the required number.

19
20 Basically, in order to generate sound effects, the user has to
21 prepare music notes and call the sound generation routines. The
22 notes table, pointer and four routines are described below. For
23 detailed information, refer to the Sound Users' Manual in
24 Appendix C.
25
26

1 7.1 LST_OF_SND_ADDRS and PTR_TO_LST_OF_SND_ADDRS
2

3 All the music notes for an application program starts at
4 the address called LST_OF_SND_ADDRS in cartridge ROM.
5 There is another dedicated CRAM pointer located at
6 address PTR_TO_LST_OF_SND_ADDRS which points to the
7 LST_OF_SND_ADDRS. It is the user's responsibility to
8 set up the pointer before passing control to any sound
9 generation software.

10
11 7.2 SOUND_INIT

12 This routine should be called immediately after power
13 on, before any sound processing can occur. It turns off
14 the sound generators, initializes the CRAM locations to
15 be used as sound data areas, sets up the four channel
16 data area pointers and initializes
17 PTR_TO_LST_OF_SND_ADDRS.

18 INPUT: n

19
20 TYPE: 8-bit constant

21 PASSED: in B

22 DESCRIPTION: Number of sound data areas used by
23 the game.
24
25
26

1 INPUT: LST_OF_SND_ADDRS
2 TYPE: 16-bit address
3 PASSED: in HL
4 DESCRIPTION: LST_OF_SND_ADDRS is the base
5 address of a list of the starting
6 addresses for each sound's note
7 list and data area.

8
9 OUTPUT: 1. Turns off all sound
10 generators.
11 2. Initializes
12 PTR_TO_LST_OF_SND_ADDRS.
13 3. Writes the inactive code
14 (OFFH) to byte 0 of the n
15 sound data areas.
16 4. Stores 00 at end of sound data
17 areas.
18 5. Sets the 4 channel sound
19 pointers to a dummy inactive
20 area.
21 6. Sets SAVE_CTRL to OFFH. (See
22 "Noise Notes" discussion in

3
4
5
6

ColecoVision Sound Users'
Manual in Appendix C).

7.3 PLAY_IT

PLAY_IT is called to start a sound. Using a sound number passed in B, PLAY_IT loads the data for the sound's first note into the appropriate sound data area, thereby truncating whatever sound had been "playing" in that data area. (The address of the appropriate area is found by using the sound number as an index into the LST OF_SND_ADDRS table). It also formats the data area's header and sets up the next note pointer. If the sound is a special sound effect, its next note pointer is set to the address of the special effect routine. The next time PLAY_SONGS is called, that sound's first note will be played.

If PLAY_IT is called with a sound number of a sound which is already in progress, it returns immediately (i.e., it doesn't restart the sound).

1 INPUT: Sound number to be started.
2 TYPE: 8-bit constant, 1 to 61.
3 PASSED: In B.
4 CALLS: PT_IX_TO_SxDATA,
5 LOAD_NEXT_NOTE PTR,
6 UP_CH_DATA_PTRS.
7
8 OUTPUT: 1. Moves the sound's first note
9 data to the appropriate sound
10 data area.
11 2. Formats byte 0 header of the
12 sound's data area.
13 3. Points next note pointer in
14 data area (bytes 1 & 2) to
15 address of first note in
16 sound, or address of special
17 sound effect routine.
18
19
20
21
22
23
24
25
26

1
2
3 7.4 SOUND_MAN

4 SOUND_MAN should be called every VDP interrupt. For
5 each data area, SOUND_MAN processes the appropriate
6 timer and sweep counters and modifies the frequency and
7 attenuation data accordingly. If the data area is
8 assigned to a special effect, SOUND_MAN calls that
9 effect. When a note is finished, SOUND_MAN, using the
10 data area's next note pointer, moves data for the next
11 note of the sound into the area. If SOUND_MAN reads a
12 header byte (in Cart ROM) that has bits 3 and 4 set,
13 indicating repeat sound, it will start the sound again
14 by reloading the first note in the sound.

15
16 After the operations upon a data area have been per-
17 formed, if necessary, the channel data area pointers
18 (PTR_TO_S_ON_x) are updated. The following data areas
19 are processed in the same fashion, in order of
20 occurrence, until the end of data area code, 00, is
21 reached.
22
23
24
25
26

1 SOUND_MAN does not output the modified frequency and
2 attenuation data. PLAY_SONGS is called just before
3 SOUND_MAN to do this.
4

5 Special codes in byte 0 of the sound data area indicate:
6

7 255: Data area inactive, do no processing;

8 62: A special effect is to be played; SOUND_MAN calls
9 the effect routine;

10 0: End of sound data areas (SOUND_MAN processes data
11 areas until it sees 0 in byte 0).

12 NOTE: Sound number 1 MUST use the first area in the
13 block of sound data areas. SOUND_INIT uses this
14 address to find the sound data area.
15

16 INPUT: None.

17 CALLS: PT_IX_TO_SxDATA,
18 PROCESS_DATA_AREA.
19

20 OUTPUT: Calls routines which:

- 21 1. Decrement sound duration and
22 sweep timers.
23
24
25
26

2. Modify swept frequency and attenuation values.
3. Call special effects routines where necessary.
4. Update the channel data area pointers if necessary.
5. Restart the sound if indicated.

7.5 PLAY_SONGS:

PLAY_SONGS takes the frequency and attenuation data pointed to by the four channel data area pointers (PTR_TO_S_ON_X) and outputs it to the four sound chip generators.

INPUT: None.

CALLS: TONE_OUT, UPATNCTRL.

OUTPUT: 1. Current frequency and attenuation data is output to

1 each tone generator, if sound
2 on that channel is active; if
3 sound on that channel is
4 inactive, that generator is
5 turned off.

- 6 2. Noise generator is sent
7 current attenuation data and
8 control data, if new.
9 3. Modifies SAVE_CTRL if
10 necessary.

11
12 7.6 Application

13
14 These four routines would normally be called as follows:
15

16 Begin

17 Power on inits done by OS

18 Cartridge program receives control

19 LD B, number of song data areas used in the
20 game

21 LD HL, address where LST_OF_SND_ADDRS is
22 store in ROM.
23
24
25
26

1 CALL SOUND_INIT to initialize sound data areas
2 Whatever other power on inits you want to do
3 Start game
4 .
5 .
6 .
7 LD B, number of sound you want to start
8 CALL PLAY_IT to set up for start of sound
9 .
10 .
11 VDP interrupt occurs:
12 CALL PLAY_SONGS to output data
13 CALL SOUND_MAN to process sound data
14 Whatever else you want to do during VDP
15 interrupt
16 RETN to game

17 End
18
19
20
21
22
23
24
25
26