

SECTION VII  
SOUND GENERATION SOFTWARE

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

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

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

7.1 LST\_OF\_SND\_ADDRS and PTR\_TO\_LST\_OF\_SND\_ADDRS

All the music notes for an application program starts at the address called LST\_OF\_SND\_ADDRS in cartridge ROM. There is another dedicated CRAM pointer located at address PTR\_TO\_LST\_OF\_SND\_ADDRS which points to the LST\_OF\_SND\_ADDRS. It is the user's responsibility to set up the pointer before passing control to any sound generation software.

7.2 SOUND\_INIT

This routine should be called immediately after power on, before any sound processing can occur. It turns off the sound generators, initializes the CRAM locations to be used as sound data areas, sets up the four channel data area pointers and initializes PTR\_TO\_LST\_OF\_SND\_ADDRS.

INPUT: n

TYPE: 8-bit constant

PASSED: in B

DESCRIPTION: Number of sound data areas used by the game.

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  
23  
24  
25  
26

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

7.4 SOUND\_MAN

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

After the operations upon a data area have been performed, if necessary, the channel data area pointers (PTR\_TO\_S\_ON\_x) are updated. The following data areas are processed in the same fashion, in order of occurrence, until the end of data area code, 00, is reached.

SOUND\_MAN does not output the modified frequency and attenuation data. PLAY\_SONGS is called just before SOUND\_MAN to do this.

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

255: Data area inactive, do no processing;

62: A special effect is to be played; SOUND\_MAN calls the effect routine;

0: End of sound data areas (SOUND\_MAN processes data areas until it sees 0 in byte 0).

NOTE: Sound number 1 MUST use the first area in the block of sound data areas. SOUND\_INIT uses this address to find the sound data area.

INPUT: None.

CALLS: PT\_IX\_TO\_SxDATA,  
PROCESS\_DATA\_AREA.

OUTPUT: Calls routines which:  
1. Decrement sound duration and sweep timers.

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  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26

each tone generator, if sound  
on that channel is active; if  
sound on that channel is  
inactive, that generator is  
turned off.

2. Noise generator is sent  
current attenuation data and  
control data, if new.
3. Modifies SAVE\_CTRL if  
necessary.

#### 7.6 Application

These four routines would normally be called as follows:

Begin

Power on inits done by OS

Cartridge program receives control

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

LD HL, address where LST\_OF\_SND\_ADDRS is  
store in ROM.

---

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	