P627

DICK SMITH ELECTRONICS

NSW	125 York Street,	SYDNEY.	Ph 290 3377
	147 Hume Highway,	CHULLORA.	Ph 642 8922
	162 Pacific Highway,	GORE HILL.	Ph 439 5311
	30 Grose Street,	PARRAMATTA	Ph 683 1133
	613 Princes Highway,	BLAKEHURST	Ph 546 7744
	263 Keira Street,	WOLLONGONG.	Ph 28 3800
VIC	399 Lonsdale Street,	MELBOURNE.	Ph 67 9834
	656 Bridge Road,	RICHMOND.	Ph 428 1614
QLD	166 Logan Road,	BURANDA.	Ph 391 6233
	842 Gympie Road,	CHERMSIDE.	Ph 59 6255
ACT	96 Gladstone Street,	FYSHWICK.	Ph 80 4944
SA	60 Wright Street,	ADELAIDE.	Ph 212 1962
WA	414 William Street,	PERTH.	Ph 328 6944

MAIL ORDER CENTRE: PO Box 321, NORTH RYDE NSW 2113. Ph 888 3200. PACK & POST EXTRA.

VISA

SHOPS OPEN 9AM to 5.30PM
(Saturday: 9am till 12 noon)
BRISBANE: Half hour earlier.
ANY TERMS OFFERED ARE TO APPROVED APPLICANTS ONLY
RE-SELLERS OF DICK SMITH PRODUCTS IN MOST AREAS OF AUSTRALIA.

- GIVE YOUR 16K SYSTEM 80 A 'VOICE'!
- GENERATES ALL SORTS OF SOUND EFFECTS!
- USE YOUR SYSTEM 80 FOR MUSIC SYNTHESIS!

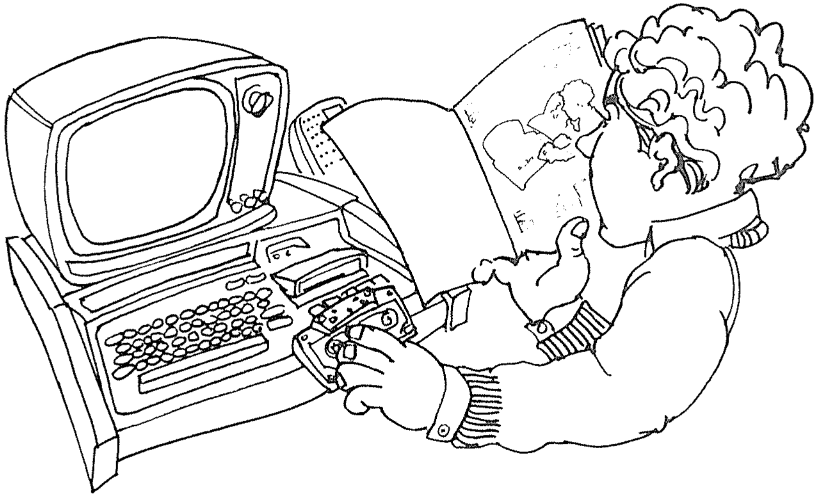
includes instruction booklet, amplifier & program cassette

"Also suits TRS-80

Dick Smith's
'SOUND OFF'
For the 16K System-80 Computer

(Also suitable for the Tandy TRS-80 Level II)

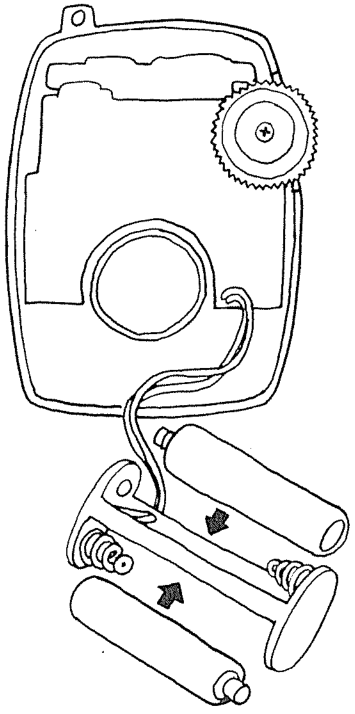
Welcome to the exciting world of computer sound effects and music synthesis! Your low cost Dick Smith "SOUND-OFF" hardware and software will let you explore this new world, and add a whole new dimension to your personal computing.



"SOUND-OFF" consists of a small amplifier unit which connects to your 16K System 80 (or Tandy TRS-80 Level II) computer via the normal cassette recorder cable. Along with this amplifier comes a tape cassette with software which tells your computer how to manipulate its cassette interface, to produce a wide variety of sound effects. On one side of the cassette there is a sound effects demonstration program, designed to give you an idea of the different sounds that can be produced; on the other there is a "patch" program, designed so that you can combine it with programs of your own. This allows you to add sound effects to all sorts of programs, with a minimum of effort.

Connecting up the "SOUND-OFF" amplifier.

The "SOUND-OFF" amplifier unit is powered by two AA-size "penlight" batteries. These are not supplied with the unit, as batteries have a limited shelf life and can release a corrosive liquid when they deteriorate. Any standard penlight cells may be used; alkaline cells are not required as the current drain is quite modest. The Hi Watt AA cells (Cat. S-3003) available from Dick Smith stores are very suitable. Two cells are required.



To fit the cells, remove the back of the case by prising it off gently at the bottom using a coin or small screwdriver in the slot provided. Then insert the cells in the small plastic battery holder, with the flat (negative) end of each cell at the end of its compartment having the spring contact. Finally replace the rear of the case, pushing it gently until it clicks into place.

The amplifier connects to your computer via the normal cassette recorder cable. The 3.5mm plug on the "MIC" or "AUX" leads (the yellow lead for the System 80, or the grey lead for the TRS-80) is simply plugged into the corresponding jack socket on the top of the "SOUND-OFF" case. All that is then required is to turn on the "SOUND-OFF" amplifier using the volume control/switch, and set it to a convenient volume setting.

Important note for System 80 owners.

Please make sure that your System 80's "F1" key is in the UP position before attempting to run any programs using the "SOUND-OFF". Although this key is nominally provided to allow local control of the inbuilt cassette deck, it also disables the external cassette port when in the DOWN position. As "SOUND-OFF" uses the external port, this would prevent it from operating.

When you have finished a session using "SOUND-OFF", don't forget to switch off the amplifier unit by turning the volume control down until it "clicks". Although the amplifier draws very little current when it is not amplifying signals, the batteries would still be slowly drained. If you are not likely to be using the amplifier for a considerable time, it is also a good idea to remove the batteries in case they corrode.

About the "SOUND-OFF" software.

The "SOUND-OFF" software supplied on cassette tape consists of a sound effects demonstration program and a "patch" program which lets you add sound effects easily to your own programs. Both programs incorporate a machine language subroutine which is used to generate the desired audio output signals via the computer's cassette interface circuitry. A machine language subroutine is used in order to achieve the necessary speed.

The rest of the programs are written in BASIC, to allow you to analyse their operation and modify them as required. The programs are arranged to automatically load the machine language routine themselves, as soon as you tell them to "RUN". This is done by a BASIC subroutine which loads the machine language routine via a series of POKE statements. So when the programs begin running they both print out "SETTING UP" and then appear to pause for about 15 seconds while the machine language routine is loaded.

Note that it is necessary to ensure that there will be space available in the computer's RAM memory to fit the machine language routine, above the BASIC program area. In fact you must reserve at least 127 bytes at the top of memory before the programs are loaded from cassette.

You do this by keying in a memory size figure of 127 bytes less than the actual memory size of the machine, when it is first turned on and asks you READY? (in the case of the System 80) or MEMORY SIZE? (in the case of the TRS-80). Instead of simply hitting the NEWLINE (or ENTER) key, type in 32640 (or the corresponding number for larger machines). Then hit the NEWLINE or ENTER key and proceed.

Both the sound effects demonstration program and the "patch" program are loaded using the normal CLOAD command. With the System 80 computer, you do this using the internal tape deck; with the TRS-80 you must use an external recorder with its volume control set for the usual level for Level II tapes.

In both cases the programs ask you to indicate whether you are running them on a System 80 or a TRS-80 machine. This is necessary because the TRS-80 has only one cassette port in the basic machine, while the System 80 has two – of which the external one must be used with "SOUND-OFF".

The sound demonstration program.

This program gives you a "menu" on the screen, listing no less than 12 different sound effects. You can choose any of the sounds by keying in the corresponding alphabetic character. The program runs continuously, letting you try the sounds over and over again. However you can terminate the program by typing a "Q". Letters other than A–L inclusive and Q are ignored.

The patch program.

This program is designed to make it easy to provide your own programs with sound effects. Written in BASIC, it consists purely of the initialising routine together with the routine to "POKE" the machine language "SOUND-OFF" routine into the top of memory. This provides a "skeleton" into which you can fit your own programs. Once you get a program working with the "SOUND-OFF" patch, you can then CSAVE the combination on a new cassette tape, as a complete working program.This will still leave you with the original "patch" program on your main cassette, available for use with further programs.

If you LIST the "SOUND-OFF" patch program you will see that it consists of two parts. The smaller part is an initialising sequence (lines 0, 1 and 2), which must go at the start of your program. The main function of this sequence is to call the second part of the program, which is the loader for the machine language routine. This occupies lines 50000 – 52060 inclusive, which logically go at the top of your program. So your own program fits in between the two, and can use any line numbers between 3 – 49998 inclusive.

Your BASIC program "calls" the "SOUND-OFF" machine language routine by using the NAME function. Before doing so, however, you must arrange for it to set the various sound control parameters used by the machine language routine. There are five of these parameters, controlling the sound's (a) pitch; (b) duration; (c) number of pitch changes; and (d,e) the direction and magnitude of the pitch changes. Table 1 summarises the parameters and where they are stored in memory (actually the first seven locations of the machine language routine).

The various parameters will now be described in more detail.

Note: The address label "TM" is predefined, in line 1 of the patch program, as the start of the reserved area in memory. Hence all parameter values may simply be poked into memory using the address labels shown in Table 1.

TABLE 1

Poke Address	Sound Parameter	Default Value
TM	pitch (lo)	50
TM+1	pitch (hi)	0
TM+2	duration (lo)	15
TM+3	duration (hi)	0
TM+4	number of steps	0
TM+5	step size 1	0
TM+6	step size 2	0

PITCH: This parameter determines the primary or starting frequency of the sound. It is stored as a 2-byte number, with the less significant byte in address TM and the more significant byte in address TM+1. The numbers are actually used to control inverse frequency, or period, so small numbers produce high frequencies while larger numbers give lower frequencies. The useful range for this parameter is from about 560 down to about 16, a range of five octaves. Middle C corresponds to a value of 140. Table 2 gives the values of the pitch parameter for five musical octaves of the even-tempered musical scale, to simplify composing tunes.

Note that because the pitch parameter must be stored in memory as two bytes, you must work out the two-byte equivalent of the desired number, and arrange for your program to store the less significant byte in TM, and the more significant byte in TM+1. To work out the two-byte equivalent of the number, divide it first by 256. The integral (whole number) part of the answer becomes the more significant byte (store in TM+1), while the remainder becomes the less significant byte (store in TM).

If the parameter value is less than 256, things are really simple: the more significant byte becomes zero and all you need to do is store the whole number in TM.

TABLE 2

PITCH PARAMETER VALUES FOR MUSICAL NOTES

Note	Octave 1	Octave 2	Octave 3	Octave 4	Octave 5
C	560	280	140	70	34
C-sharp	524	262	131	66	32
D	500	250	125	62	30
D-sharp	476	238	119	59	28
E	442	221	111	55	26
F	420	210	105	52	25
F-sharp	396	198	99	49	24
G	376	188	94	46	22
G-sharp	352	176	88	44	21
A	336	168	83	41	19
A-sharp	316	158	79	39	18
B	298	149	74	36	17
C	280	140	70	34	16

DURATION: This parameter controls how long the sound lasts. Here the number corresponds to the actual number of sound cycles – so a particular number will give a shorter sound at a higher pitch than at a lower pitch. In order to produce sounds of the same absolute length at different pitches, you will need to keep the product of the pitch and duration parameters constant so that there are more cycles produced at higher pitches.

Again the duration parameter must be stored as a two-byte number, in memory addresses TM+2 (less significant byte) and TM+3 (more significant byte). The available range is thus from 1 cycle to 65,535 cycles – but in practice you'll rarely need to use numbers larger than 255 unless you want sounds lasting quite a long time. So for most purposes, you only need to put the duration number (<255) into address TM+2.

Generally speaking, you only need the pitch and duration parameters to sound steady musical notes or play tunes. For example to produce a steady note at middle C (261Hz) lasting for 0.5 seconds (130 cycles), all you need is the line:

```
100 POKE TM, 140: POKE TM+2,
    130:NAME
```

Or, to play the first 6 notes of "Happy Birthday", the sequence:

```
100 P=140:D=50:GOSUB 200
110 P=140:D=50:GOSUB 200
120 P=125:D=168:GOSUB 200
130 P=140:D=150:GOSUB 200
140 P=105:D=200:GOSUB 200
150 P=111:D=189:GOSUB 200
160 END
200 POKE TM,P:POKE TM+2, D:
     NAME:RETURN
```

THE REMAINING PARAMETERS:

These provide a means whereby sounds may be arranged to change up or down in pitch a number of times, before ending. There are three parameters available to do this, and they make it possible to create many complex sounds with a single call of the "SOUND-OFF" machine language subroutine.

The first of the additional parameters (NS) specifies how many changes in pitch or "steps" are to take place, after the primary note sounds for the specified pitch and duration. The second and third parameters (SS1 and SS2) specify the actual pitch changes that are to occur during those steps.

NS is a single-byte variable stored in the address TM+4. It may have any value up to 255 – but note that for values more than 2, the pitch changes for subsequent steps simply use the data specified by SS1 and SS2 alternately.

SS1 and SS2 are single-byte numbers which are used to modify the basic pitch parameter of the sound. SS1 is stored in TM+5, and SS2 in TM+6. SS1 is used to calculate the pitch of the first step and any subsequent odd-numbered steps, while SS2 is used to calculate the pitch of the second step and any subsequent even-numbered steps. SS1 and SS2 are added algebraically to the less significant byte of the main pitch parameter. This means that they are basically signed numbers: values from 1 to 127 will **increase** the effective pitch parameter (lowering the pitch), while values from 128 to 255 will **decrease** the pitch parameter and increase the pitch.

Note, however, that although SS1 and SS2 are used over and over for sounds with more than two steps, they are **cumulative.** So that if SS1=3, it will increase the pitch parameter (lowering the note) by 3 units for each odd-numbered step. Similarly if you make SS2=249 (equivalent to -7), it will decrease the pitch parameter (increasing the pitch) by 7 unit for each even-numbered step.

Note also that if you use NS, SS1 and SS2 to produce additional "steps" after the main sound, the steps use the same duration parameter figure as the main sound. However this does not mean that the steps will have the same absolute duration, as this will depend upon the pitch. Hence steps which are of higher pitch than the primary sound will be shorter, while those with lower pitch will be longer. So you can create sounds which either move up the scale and get faster and faster (like the "saucer" sound in the demonstration program), or move down the scale and get slower and slower (like a "saucer" landing). And by suitably arranging SS1 and SS2, you can create all sorts of "vibrato" and "warble" effects. The possibilities are almost endless!

The default values of NS, SS1 and SS2 are zero, so if your BASIC program doesn't specify these parameters you won't get any steps or changes in pitch for each sound.

Table 3 shows the values of all five parameters used for some of the sound effects provided on the

TABLE 3

0 1 2 3 4 5 6

SOUND	PITCH		DURATION		NS	SS1	SS2
	Lo	Hi	Lo	Hi			
Saucer	250	0	25	0	249	5	249 (-7)
Fast Laser	15	0	10	0	30	1	0
Slow Laser	5	0	5	0	100	2	0
Bomb	15	0	30	0	150	1	0
Pong-A	50	0	15	0	0	0	0
Pong-B	200	0	7	0	0	0	0
Siren-up	200	0	15	0	100	255	0
Siren-down	100	0	15	0	100	1	0

"SOUND-OFF" demonstration program. Trying these out for yourself should help you in getting familiar with the way they work. Then you can try the effect of varying them, and creating some new sounds of your own!

A final point: the cassette output circuitry of your System 80 computer (also that of a TRS-80) is actually capable of delivering a variety of waveforms. So theoretically it is possible to make your "SOUND-OFF" amplifier produce sounds with different waveforms or "tone colours". For simplicity, however, the machine language subroutine supplied in the cassette programs does not allow waveform to be varied.

Those of you who are experienced in machine language programming may care to write a more elaborate machine language subroutine of your own, to allow waveform to be specified as well. This will let you use your "SOUND-OFF" to generate an even wider range of sounds. . .

Have fun!