5
Supersound

The Electron is capable of producing a wide range of sounds. If you have read Chapter 22 of the User Guide, you will know that these are programmed by using the BASIC keywords, SOUND and ENVELOPE. You can, of course, use SOUND on its own to produce simple tones or bursts of noise. The more exciting effects are produced by using SOUND together with ENVELOPE. SOUND makes the sound generator go into action, but ENVELOPE modifies its action to produce a more interesting sound.

The difficulty with these two commands is that they both need several numbers, or parameters, after them to specify what effect each shall have. SOUND needs four parameters, while ENVELOPE needs fourteen! Fortunately, the last six parameters ofthe ENVELOPE may all be zeros, so this simplifies matters a little. Actually it is better not to finish with six zeros but to use the parameters listed on page 121 of the User Manual instead. Then your Electron programs will also run on the BBC Microcomputer. SUPERSOUND uses these values when it works out what parameters are needed.

This brings us to the subject ofwhat SUPERSOUND does. It takes all the hard work out of designing sounds and sound effects. You can, of course, design your effects by typing in lines of an ordinary BASIC program. The trouble is that there are so many parameters. They can take so many different values, that working out which parameter should have which value is very confusing. It is hard enough to decide what values to use to begin with, but later, when you want to change one or two parameters to make the sound just right, you need to retype your program lines all over again. This takes a lot of time, and can lead to errors. SUPERSOUND lets you type in the essential parameters, then hear the sound, then change just those parameters that need to be changed and hear the sound again. Little by little you can find the best values to use. At every stage all the parameters are displayed on the screen. When you have got the sound just as you want it to be, its parameters are there on the screen, ready for you to copy into your own BASIC programs.

Using the program

The program begins by asking you 'Envelopes or Sounds?'. If you choose 'Envelope', by typing 'E', you next key in the parameters of one or more envelopes, as explained in a moment. If you type 'S' you go straight to the part of the program in which you design and hear the sounds. If you only want to produce straightforward tones, you need not bother with envelopes. If this is the first time you are using the program it might be better to start with sounds and come back to envelopes later.

So let us suppose that you have selected 'Sounds'. The screen clears and now a table is displayed (Fig. 5.1). Actually there are two tables on the screen, the bottom one having to do with envelopes. This contains all zeros and we will not concern ourselves with this at present.

```
        1   2   3   4   5   6   7   8


CH.0 A
     P
     D
CH.1 A 1
     P 80
     D 14
_

            A   Pi1 Pi2 Pi3 Pr1 Pr2 Pr3

Envelope 1  130 5   -3  1   10  5   20

Envelope 2  0   0   0   0   0   0   0

Envelope 3  0   0   0   0   0   0   0

Envelope 4  0   0   0   0   0   0   0
```

Fig 5.1. The tables of SUPERSOUND, with a sound effect on Channel 1, and Envelope 1 in operation. The line below the upper table is the cursor, waiting for the next command.

The upper table has eight columns. These are to hold the parameters of up to eight different sounds, which the computer will produce one after the other. In this way you can build up a sequence of sounds to make a complicated sound effect. Or you can settle for just a single sound, using only column 1 of the table.

The left-hand side of the table refers to Channel 0, which is the noise channel, and to Channel 1, which is one of the tone channels. The Electron has two other tone channels (2 and 3) but since only one channel can be made to sound at a time, this program does not use these two channels. You cannot have channels 0 and I sounding at once, either, so your sound design must allow for this (see later). Beside each channel's name, there are 3 letters, A, P, and D, standing for the 3 parameters which you have to enter.

A-parameter: this can be
-15 = 'sound on'
0 = 'sound off'
1 = 'use envelope 1'
2 = 'use envelope 2'
3 = 'use envelope 3'
4 = 'use envelope 4'

The Electron would accept any negative number instead of -15, but the program accepts only -15, so as to make the sound command compatible with the BBC Micro. Using -15 gives a simple burst of sound, without the effect of any envelope.

P-parameter: this decides the pitch of the note. It can be any value in the range 0 to 255. The higher the value, the higher the pitch. Page 118 of the User Guide tells you which values correspond to notes of various pitch on the musical scale. On the noise channel (CH.0) only the values 0 to 7 have different effects, as listed on the same page of the User Guide.

D-paramerer: this sets the duration (length) of the tone. The value of D must be between 1 and 254. This gives a sound of D × 50 milliseconds. If D = 10, the sound lasts for half a second. If D = 20, it lasts a whole second, and so on. The User Guide mentions making D equal to -1, so that the sound continues indefinitely, but this is not allowed for in this program.

When the table is first displayed, you see the cursor winking at the top of the first column, ready for you to type in the first parameter. If you want to use Channel 0, type -15. If you are using an envelope. type its number (1 to 4) instead. Press RETURN. The cursor immediately drops to the row below, ready for you to enter the pitch. If it happens that you have keyed in a number that is not allowed as an A-parameter, the cursor does not drop, but waits in the same row, for you to re-enter a correct number. The incorrect number remains on the screen, and you have to replace it by typing in the correct value. Key in the P-parameter and the D-parameter in the same way.

If you do not want to use Channel 0, just press RETURN at each stage. The cursor drops down line-by-line to the Channel 1 section. There you key in values for Channel 1. Or perhaps you have entered values for Channel 0, in which case you can by-pass Channel 1 by pressing RETURN at each line.

When you have pressed RETURN for the D-parameter of Channel 1 , you have reached the bottom of the column. The cursor then goes to the extreme left of the screen, just below the table. Here it waits for your next command. Issue this by keying in these letters:

S START - the cursor goes back to the beginning of the table, so that you can alter one or more of the parameters in any part of the table.
R REPEAT - the cursor goes back to the top of the column you have just finished, so that you can alter one or more of the parameters in that column.
N NEXT - the cursor goes to the top of the next column, for you to enter the parameters for the next sound in the sequence.
E ENVELOPE - go to the part of the program where envelopes are defined.
0 CHANNEL 0 - sound Channel 0.
1 CHANNEL 1 - sound Channel 1.

After sounding either of the channels, the cursor is still below the table; you can repeat the sound, or press any other of the keys listed above to continue designing your sounds. If you hold the '0' or '1' key down for a little longer than usual, the sound is repeated several times. Do not hold it down for too long, for the sound could be repeated for several minutes! If you want a sound effect to consist of a noise followed by a tone (or a tone followed by a noise), key in the two sets of parameters for Channels 0 and 1 . Press '0' or '1' for the first sound and while it is sounding press the '0' or '1' for the second sound. The second sound will follow immediately after the first, giving the effect you require.

If you use S or R to alter parameters, the keying in is as described earlier. You have to re-key every parameter, simply copying any that you do not wish to change. When you key E to go to 'Envelopes', the screen clears and your sound parameters are lost.

Designing Envelopes

This section of the program first asks 'Which number?' In response to this, key a number from 1 to 4. The Electron can have up to four envelopes. If you key a number ofan envelope that you have already defined, the old definition is replaced by the new one.

You are next asked to enter the following details:

Auto-repeat: If you key 'Y', the envelope is repeated for as long as the sound lasts. If you key 'N', the envelope acts only once, after which (if the sound is still continuing) there is a steady note.

Timing: This is the duration of every change of pitch, in tens of milliseconds (hundredths ofa second).

Change of pitch: This question is repeated for each of the three sections of the envelope (Fig. 5.2). The change decides by how much the pitch is to increase or decrease at each step.

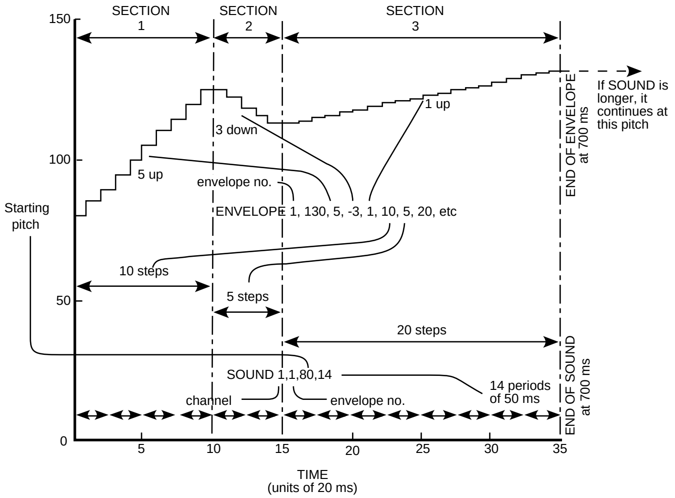

Fig. 5.2. A pitch envelope (see text). The value 130 in the envelope statement equals 128 + 2. The 128 means no auto-repeat; the 2 means that steps are 2 × 10 ms long.

Number of steps: This question is repeated for each of the three sections of the envelope.

The total duration of the envelope may be found by adding together the three 'step' values and multiplying the total by the time for a single step. For example, in Fig. 5.2 there are 10 + 5 + 20 steps, making a total of 35 steps. Each lasts 20 milliseconds, so the total length of the envelope is 35 × 20 = 700 milliseconds, or 0.7 seconds. If you had answered 'N' for Auto-repeat, it will be better if the sound is set to last no more than 0.7 seconds. The envelope then lasts exactly as long as the sound. Since sounds are always timed in units of 50 milliseconds, the duration of the sound should be set at no more than 35 (700/50). A suitable sound to use on Channel 1 with the envelope in Fig 5. 2 is A = 1 , P = 80, D = 1 4. Eventually, you will have finished developing a sound, or a series of sounds, and will want to know how to implement this sound in your own program. Simply copy SOUND parameters and ENVELOPE parameters (if any) from the screen. The SOUND parameter has the format:

```
SOUND n, A, P, D
```

where n is the channel number (0 or 1) and A, P and D are the values copied from the sound table on the screen.

The ENVELOPE command is the keyword 'ENVELOPE' followed by the numbers copied from the envelope table on the screen. This includes the envelope number. Six more parameters must follow this: 126, 0, 0, -126, 126, 126. For example, ifthe envelope shown in Fig. 5.2 is envelope number 1, the BASIC statement for it is:

```
ENVELOPE 1,130,5,-3,1,10,5,20,126,0,0,-126,126
```

When you use ENVELOPE in a program it is not essential to place it in the program before the SOUND statements which use it. You could put all the envelopes at the end of the program, like DATA statements, where they are easy to find.

Keying in

Check lines 330, 780 and 790 carefully after you have keyed them in, otherwise you may get odd sounds or no sound at all. The spacing in line 430 needs to be copied exactly so that the headings of the envelope table are properly aligned.

Program Design

20-50 Initialising mode, arrays and colours.
60-100 Envelopes or sound?
110-140 Getting the number of the envelope.
150-180 Auto-repeat or not?
190-200 Timing?
210-250 Change in pitch?
260-300 Number of steps?
310-320 Confirm that all entries are OK.
330 Define the envelope.
340 Return to selection.
350-420 Display empty sounds table.
430-470 Display envelope table.
480-690 Input column of parameters, then move cursor to below table.
700-720 Loop to input acceptable key-press and repeats input for next column if key is ' N' .
730 Ends main input loop if other keys pressed.
740-760 To other parts of program.
770-800 SOUND statements to produce sounds, in order.
810-880 To next stage of program.
890-940 PROCqueries for inputting a value w1th1n given limits.
950-1010 PROCclear for clearing array of old sound parameters.

The program

```
   10 REM ** SUPERSOUND **
   30 DIM E%(4,6),S%(1,3,8)
   40 VDU 19,0,4,0,0,0
   50 COLOUR 2:COLOUR 128
   60 CLS:PRINT''TAB(12)"!! SUPERSOUND !
!"
   70 REPEAT
   80 PRINT TAB(0,7)"Envelopes or Sounds
? (E/S)  ";:KEY$=GET$
   90 UNTIL KEY$="E" OR KEY$="S"
  100 IF KEY$="S" THEN 350
  110 REPEAT
  120 CLS:PRINT'TAB(15)"ENVELOPES"
  130 PROCqueries("Which number? (1-4)
",1,4,3)
  140 NE=M
  150 REPEAT
  160 INPUT TAB(0,5)"Auto-repeat? (Y/N)
    "A$
  170 UNTIL A$="Y" OR A$="N"
  180 E%(NE,0)=0:IF A$="N" THEN E%(NE,0)
=128
  190 PROCqueries("Timing? (0 to 127)  "
,0,127,7)
  200 E%(NE,0)=E%(NE,0)+M
  210 FOR J=1 TO 3
  220 COLOUR 129:PRINT TAB(0,10+J*2)"SEC
TION ";J:COLOUR 128
  230 PROCqueries("Change in pitch? (-12
8 to 127) ",-128,127,11+J*2)
  240 E%(NE,J)=M
  250 NEXT
  260 FOR J=1 TO 3
  270 COLOUR 129:PRINT TAB(0,19+J*2)"SEC
TION ";J:COLOUR 128
  280 PROCqueries("Number of steps? (0 t
o 255)    ",0,255,20+J*2)
  290 E%(NE,J+3)=M
  300 NEXT
  310 PRINT'"ALL OK? (Y/N)  ";:KEY$=GET$
  320 UNTIL KEY$="Y"
  330 ENVELOPE NE,E%(NE,0),E%(NE,1),E%(N
E,2),E%(NE,3),E%(NE,4),E%(NE,5),E%(NE,6)
,126,0,0,-126,126,126
  340 GOTO 60
  350 CLS:PRINT''TAB(17)"SOUNDS"
  360 FOR J=1 TO 8:PRINT TAB(4+4*J,6);J;
:NEXT
  370 FOR J=0 TO 1
  380 COLOUR J:COLOUR 131-J
  390 PRINT TAB(0,9+3*J)"CH.";J;" A "
  400 PRINT TAB(4,10+3*J)" P "
  410 PRINT TAB(4,11+3*J)" D "
  420 NEXT
  430 PRINT TAB(12,21)"A   Pi1 Pi2 Pi3 P
r1 Pr2 Pr3"
  440 FOR J=1 TO 4
  450 PRINT TAB(0,21+J*2)"Envelope ";J;
  460 FOR K=0 TO 6:PRINT TAB(12+K*4);E%(
J,K);:NEXT
  470 PRINT"":NEXT
  480 J=0
  490 REPEAT:J=J+1:IF J=9 THEN J=1
  500 FOR K=0 TO 1
  510 COLOUR K:COLOUR 131-K
  520 REPEAT
  530 VDU 31,4+J*4,9+K*3
  540 INPUT""X$
  550 X=VAL(X$):UNTIL X>-1 AND X<5 OR X=
-15 OR X$="R"
  560 IF X$="R" THEN J=J-1:IF J=0 THEN J
=1
  570 IF X$="R" THEN 520
  580 S%(K,1,J)=X
  590 REPEAT
  600 VDU 31,4+J*4,10+K*3
  610 INPUT""X$
  620 X=VAL(X$):UNTIL X>-1 AND X<256
  630 S%(K,2,J)=X
  640 REPEAT
  650 VDU 31,4+J*4,11+K*3
  660 INPUT""X$
  670 X=VAL(X$):UNTIL X>-1 AND X<255
  680 S%(K,3,J)=X
  690 NEXT
  700 REPEAT
  710 KEY$=GET$
  720 UNTIL INSTR("01SNER",KEY$)
  730 UNTIL INSTR("01SER",KEY$)
  740 IF KEY$="S" THEN 480
  750 IF KEY$="R" THEN J=J-1:GOTO 490
  760 IF KEY$="E"THEN COLOUR 2:COLOUR 12
8:PROCclear:GOTO 110
  770 FOR L=1 TO 8
  780 IF KEY$="0" THEN SOUND 0,S%(0,1,L)
,S%(0,2,L),S%(0,3,L)
  790 IF KEY$="1" THEN SOUND 1,S%(1,1,L)
,S%(1,2,L),S%(1,3,L)
  800 NEXT
  810 REPEAT
  820 KEY$=GET$
  830 UNTIL INSTR("01SNER",KEY$)
  840 IF KEY$="S" THEN 480
  850 IF KEY$="N" THEN 490
  860 IF KEY$="R" THEN J=J-1:GOTO 490
  870 IF KEY$="E"THEN COLOUR 2:COLOUR 12
8:PROCclear:GOTO 110
  880  GOTO 770
  890 DEF PROCqueries(M$,LL,UL,TB)
  900 REPEAT
  910 PRINT TAB(0,TB)M$;:INPUT"  "A$
  920 M=VAL(A$)
  930 UNTIL M>LL-1 AND M<UL+1
  940 ENDPROC
  950 DEF PROCclear
  960 FOR channel=0 TO 1
  970 FOR parameter=1 TO 3
  980 FOR sound=1 TO 8
  990 S%(channel,parameter,sound)=0
 1000 NEXT:NEXT:NEXT
 1010 ENDPROC
```

Variations

When the program was being designed, it was thought that most readers would prefer to have a clear sound table on the screen when returning from envelope designing. This gives an uncluttered space in which to design new sounds. Some readers may prefer to retain the sounds and the displayed details, by amending the program as explained below.

The sound parameters are stored in an array S%(). This is a three dimensional array:

S%(channel,parameter,note)

Channels run from 0 to 1, parameters from 1 to 3 (A,P,D) and notes from 1 to 8. PROCclear clears S%() before returning to the envelope section of the program. This PROC could be omitted (delete 'PROCclear' on lines 760 and 870) and you could then add routines to lines 390-410 to redisplay the contents of S%(). Any 'old' sounds that are not wanted to will then have to be cancelled by typing '0' for amplitude (A).