# Sound Generator Board

## for Tangerine systems

BULLDOG
VIDEO
LIMITED

# CONTENTS

## 1. GETTING STARTED.

Slot the sound board into any socket on the system motherboard other than that used by Tanbug or Tanex.

Type on the keyboard

```
MBCØØ       <LF>
MBCØØ,ab,07 <LF>
MBCØ1,ab,FE <ESC>
MBCØØ,FE,Ø8 <LF>
MBCØ1,ab,ØF <ESC>
MBCØØ,ØF,ØØ <LF>
MBCØ1,ab,ØØ <ESC>
MBCØØ,ØØ,Ø1 <LF>
MBCØ1,ab,Ø4 <ESC>
```

```
where <LF> is the LINE FEED key
      <ESC> is the ESC key,
        the values ab are of no interest,
        and you type the values which are underlined.
```

You should now hear a sound from the speaker. If you do not, check through your screen listing with that above.

The sound you get is a single tone.

Vary it by typing

```
MBCØØ,Ø4,Ø1  <LF>
MBCØ1,ab,ØA  <ESC>
```

then

```
MBCØØ,ØA,Ø1  <LF>
MBCØ1,ab,Ø1  <ESC>
```

Add an envelope to the sound

```
MBCØØ,Ø1,Ø8  <LF>
MBCØ1,ab,1Ø  <ESC>          silence
```

```
MBCØØ,1Ø,ØC  <LF>
MBCØ1,ab,Ø8  <ESC>
MBCØØ,Ø8,ØD  <LF>
MBCØ1,ab,ØE  <ESC>
```

Add white noise

```
MBCØØ,ØE,Ø7  <LF>
MBCØ1,ab,F6  <ESC>
```

Add an additional tone

```
MBCØØ,ØE,Ø7  <LF>
MBCØ1,ab,F4  <ESC>
MBCØØ,F4,Ø9  <LF>
MBCØ1,ab,ØF  <ESC>
MBCØØ,ØF,Ø2  <LF>
MBCØ1,ab,FF  <ESC>
```

Reset

You have now tried out some of the sounds which can be produced. The next section describes how a sound is programmed.

## 2. PROGRAMMING.

A sound is programmed by setting a number of software 'switches'. Once set, these switches remain set and the sound continues. At power on of the Microtan, or after pressing reset, all the switches are 'off' and no sound is produced.

Now, these switches are in fact 14 registers on the sound generator chip, so changing the contents of these registers alters the sound.

Details of the function of each of these registers is given in the next section but here we are describing how the contents of any of the registers may be read or how those contents may be changed.

All the registers are accessed via two locations in the Microtan store map, 48128 (hex BCØØ) and 48129 (hex BCØ1). Suppose you wish to either read the contents of one of the registers or to alter those contents and thus change the sound. Then, the number of the register must be written to location 48128. However, location 48128 will then contain not the number of the register written but its contents. Thus by then reading location 48128 you can obtain the contents of the register whose number you just wrote.

To change those contents you must write the new value to location 48129.

Note here that BCØØ and BCØ1 are the locations used for controlling the first sound generator chip. If a second chip is fitted, this will be controlled from locations BCØ2 and BCØ3. However, all of these locations may be altered by use of wire links on the board. This is discussed in section 7.

Three different programming 'languages' can be used to write to and read from the two locations. They are TANBUG, BASIC and Assembler. The example in section 1 used TANBUG so we will start with this one.

a) <u>TANBUG</u> The two locations BCØØ and BCØ1 (hex, remember) may be accessed by using the M command.

Thus typing MBCØØ <LF> produces the display

```
MBCØØ, ab, ■
```

Type in the register number required, say it is register Ø, followed by <LF> to give

```
MBCØØ,ab,Ø
MBCØ1,ab,■                                 A
```

Now, to find the contents of register Ø type ESC to give

```
MBCØØ,xy,■
```

xy is the contents of register Ø in hex.

If you wish to change the contents of register Ø to, say, FF type FF <ESC> at Ⓐ above to give

```
MBCØ1,ab,FF
MBCØØ,FF,■
```

Now type in the number of the next register you wish to change and so on.

Looking through the beginning of the example in section 1, you can see that you change register 7 to FE,
register 8 to ØF,
register Ø to ØØ,
and register 1 to Ø4.

b) <u>BASIC</u>

In a BASIC program, the instruction to declare the register number required is

```
POKE 48128, register number (in decimal)
```

Then, to find the contents of the register

```
N=PEEK (48128)
```

N will contain the contents of the register (in decimal).

If you wish to change the contents of the register, the instruction is

```
POKE 48129, new contents (in decimal)
```

As an example, type reset and then run the following program.

```
10  POKE 48128,7
20  POKE 48129,254
30  POKE 48128,8
40  POKE 48129,15
50  A  =  48
60  FOR I = 1 TO 144
70  POKE 48128,Ø
80  POKE 48129,A
90  A = A + 1
100 NEXT I
```

You should get a sweep effect.

c) <u>Assembler</u>.

In an assembler program, the instructions are

```
LDA #Ø register number (in hex)
STA  ØBCØØ
```

Then, if reading the contents of the register

```
LDA ØBCØØ
```

The accumulator will contain the contents of the register (in hex).

If you wish to change the contents of the register, the instruction is

```
    LDA #$ new contents (in hex)
    STA $BCØ1
```

As an example, the following program will enable you to play the keyboard.

```
      LDA #$Ø7
      STA $BCØØ
      LDA #$FE
      STA $BCØ1
      LDA #$Ø8
      STA $BCØØ
      LDA #$ØF
      STA $BCØ1
      LDA #$Ø1
      STA $BCØØ
      LDA #$ØØ
      STA $BCØ1
      STA $BCØØ
      CLI

POLL: JSR $FDFA              ; POLLKB
      LDA $Ø1
      STA $BCØ1
      JMP POLL
```

## 3. SOUNDS.

The sound produced from the speaker is a combination of the following independently controllable sounds. There are three sound channels each of which can produce a tone. Also there is a white noise source which can be added to one, two or all of the channels. The mixture of these sounds can be either set to a fixed amplitude or varied in a sound evelope. If the latter, any one of ten envelope shapes can be chosen.

The sound building schedule is -

Step 1. Select what mix of tone and noise is required for the finished sound.

Step 2. Decide which amplitude mode - fixed or envelope. If fixed, set the amplitude required.

Step 3. If amplitude has an envelope, set the envelope period and the envelope shape.

Step 4. Set frequency of tone for each channel selected.

Step 5. Set frequency of white noise if this has been selected on any channel.

This is shown in more detail overleaf.

| Step No. | Step Description | Register Number | | | Register Contents | | | Comment. |
|---|---|---|---|---|---|---|---|---|
| | | Octal | Decimal | Hex | Octal | Decimal | Hex | |
| 1 | Select the mix | 7 | 7 | 7 | 377 | 255 | FF | Nothing enabled |
| | | | | | 3ØØ | 192 | CØ | All channels enabled for noise and tone |
| | | | | | 376 | 254 | FE | Channel A tone enabled |
| | | | | | 367 | 247 | F7 | Channel A noise enabled |
| | | | | | 366 | 246 | F6 | Channel A noise and tone enabled |
| | | | | | | | | (Other values can be calculated from the GI documentation) |
| 2. | Amplitude mode for Channel A | 1Ø | 8 | 8 | Ø-37 | Ø-31 | ØØ-1F | Øx= fixed amplitude<br>x= Ø : off<br>x= F : loudest<br>1y= envelope mode<br>y= immaterial |
| | " " B | 11 | 9 | 9 | " | " | " | |
| | " " C | 12 | 10 | A | " | " | " | |
| 3. | Envelope period | 13 | 11 | B | Ø-377 | Ø-255 | ØØ-FF | ) ØØØ1= shortest |
| | | 14 | 12 | C | Ø-377 | Ø-255 | ØØ-FF | ) FFFF= longest |
| | Envelope shape | 15 | 13 | D | Ø-17 | Ø-15 | ØØ-OF | See GI documentation for different shapes. |

| Step No. | Step Description | Register Number | | | Register contents | | | Comment |
|---|---|---|---|---|---|---|---|---|
| | | Octal | Decimal | Hex | Octal | Decimal | Hex | |
| 4. | Tone frequency for Channel A | Ø<br>1 | Ø<br>1 | Ø<br>1 | Ø-377<br>Ø-17 | Ø-255<br>Ø-15 | ØØ-FF<br>ØØ-ØF | ØØØ1= highest tone<br>OFFF= lowest tone |
| | " " B | 2<br>3 | 2<br>3 | 2<br>3 | " | " | " | |
| | " " C | 4<br>5 | 4<br>5 | 4<br>5 | " | " | " | |
| 5. | Noise frequency | 6 | 6 | 6 | Ø-37 | Ø-31 | ØØ-1F | |

## 4. NOTES ON SOUND.

a) A pure tone is a sine wave with a fixed frequency.

b) White noise is a random mixture of sine waves with frequencies varying from ØHz to infinity.

c) The output from the sound board is square waves but these can be considered as sine waves.

d) A tone will sound "low" or "high" according to its frequency

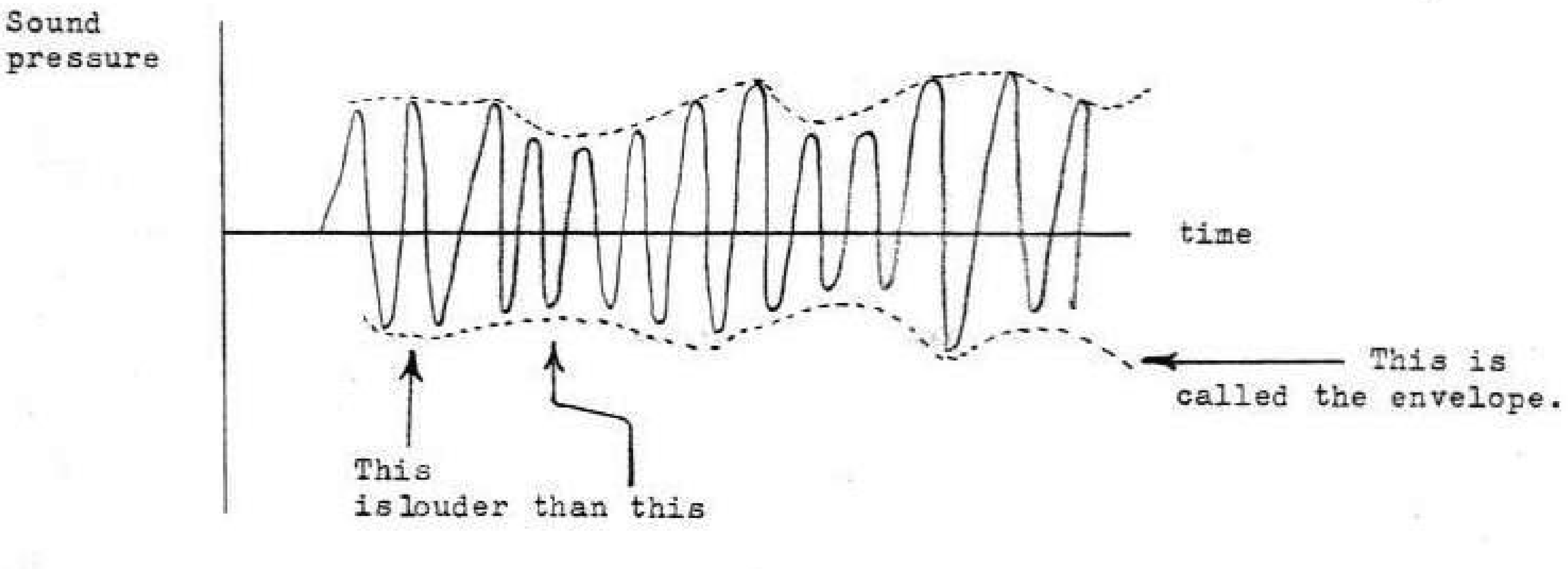
e.g. 50Hz is a low hum (mains hum),
an octave above, 100 Hz, sounds quite similar
440 Hz corresponds to the musical note A
5 KHz is a high pitched whistle.

e) White noise sounds like a waterfall (a shushing noise). The components of white noise above audible cut-off (say 15 KHz) cannot be heard, so removing them will not alter the sound.

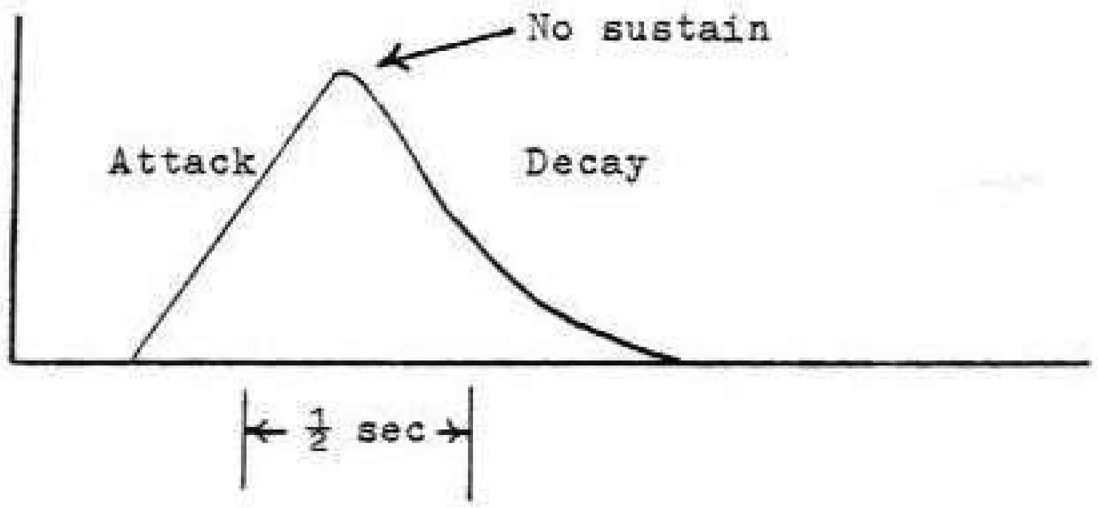
f) A violin playing note A will sound different to an organ playing note A (and both will sound different to the sound board producing that note) due to the differing amounts of harmonies in each particular note. But each fundamental frequency will be 440 Hz.

g) Sound can be modulated by altering the amplitude (loudness). A "good" female vocalist will be able to put a "tremor" into her voice by altering the loudness about 5 times a second.

This is shown graphically as follows



h) The envelope of a piano note is



i) White noise can have an envelope too

e.g. a train going chuff-chuff

## 5. FREQUENCIES.

The sound board clock runs at 0.75 MHz

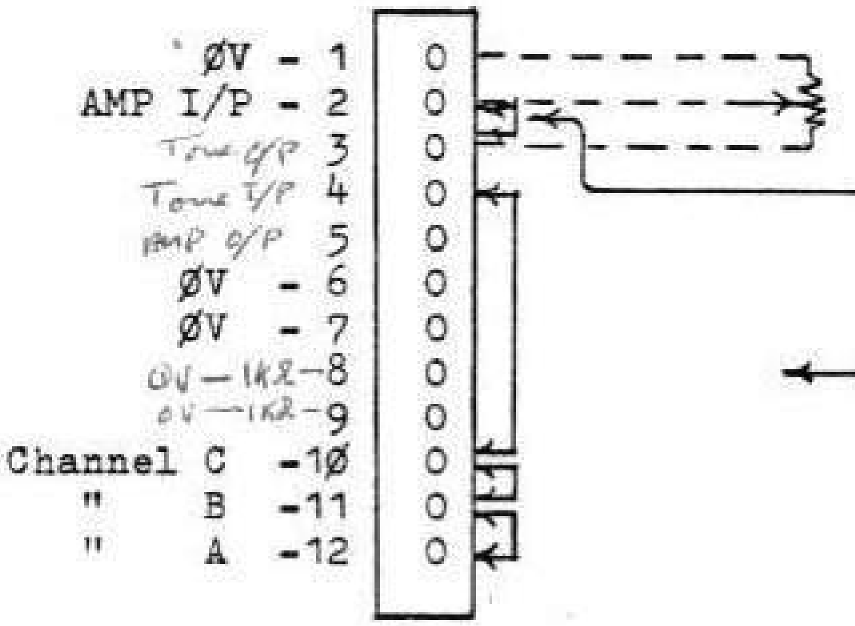
Tone frequency is calculated by dividing the clock frequency by 16 and then dividing the result by the values in the two tone registers (registers Ø and 1 for channel A)

For example

| Hex<br>Register 1 | Register Ø | Frequency | Comment |
|---|---|---|---|
| ØØ | Ø4 | 11.7 KHz | This may be inaudible for you |
| ØØ | Ø5 | 9.3 KHz | |
| ØØ | Ø6 | 7.8 KHz | |
| ØØ | Ø7 | 6.7 KHz | |
| ØØ | Ø8 | 5.8 KHz | |
| ØØ | 5A 90 | 523 Hz | C ) |
| ØØ | 5F 95 | 494 | B ) |
| ØØ | 65 101 | 466 | A# ) |
| ØØ | 6B 107 | 440 | A ) |
| ØØ | 71 113 | 415 | G# ) |
| ØØ | 78 120 | 392 | G ) |
| ØØ | 7F 127 | 370 | F# ) Musical |
| ØØ | 86 134 | 349 | F ) Notes |
| ØØ | 8E 142 | 330 | E ) |
| ØØ | 96 150 | 311 | D# ) |
| ØØ | 9F 159 | 294 | D ) |
| ØØ | A9 169 | 277 | C# ) |
| ØØ | B3 179 | 262 | C ) |
| Ø1 | ØØ | 183 | |
| Ø2 | ØØ | 91.5 | |
| Ø4 | ØØ | 46.3 | |
| Ø8 | ØØ | 22.9 | |

## 6. CONNECTING EXTERNAL AUDIO EQUIPMENT.

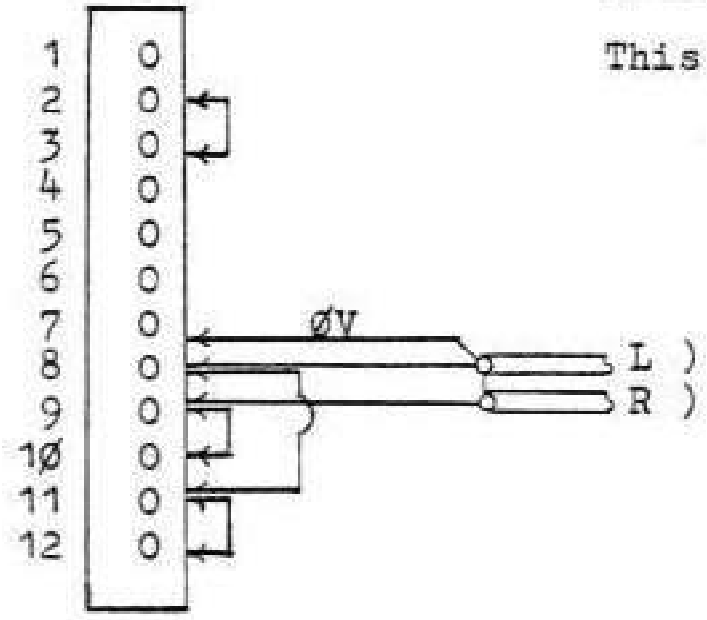
At the edge of the board are three terminal blocks making up a tagstrip. Wire links should be used to give any required configuration of audio equipment



1ØK LOG volume control may be fitted here

If vol' control fitted, remove this link

Board is supplied with these links which connect the three sound channels into the on-board amplifier

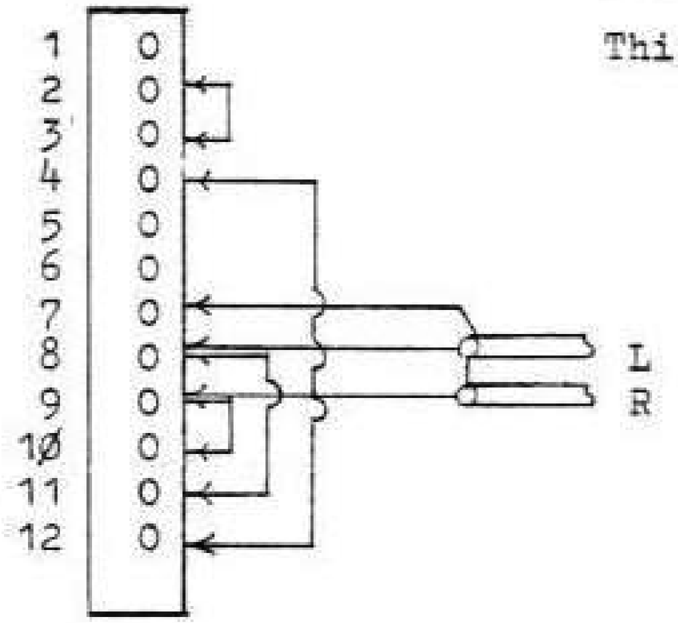


Example 1. connecting to a pre amp

This connects :-

Channel A and B to Left
Channel C to Right
On-board amp is OFF

To AUX I/P of preamp



Example 2.

This connects :-

Channel A to on-board amp.
Channel B to Left
Channel C to Right.

## 7. MEMORY MAPPING

The board as supplied will respond to addresses BCØØ and BCØ1. If the optional second AY-3-8910 is fitted then this will respond to BCØ2 and BCØ3.

Wire links LK1 and LK2 are provided to allow this address range to be altered to suit system requirements as follows -

| ADDS BITS | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | Ø | R/$\overline{W}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Ø | Selected by LK1 supplied = Ø | Ø | Ø | Ø | Selected by LK2<br>Supplied linked to ØØØ | | | AY-3-891Ø select (Ø for basic, 1 for option) | Ø | Ø | - Set Reg' Address |
| | | | | | | | | | | Ø | 1 | - Read register |
| | | | | | | | | | | 1 | Ø | - Write register |
| | | | | | | | | | | 1 | 1 | - Inactive |

NOTE. Upper address bits 1Ø - 15 are decoded by IOE = 1Ø1111