



OUTPUT: MEMOTECH ASSEMBLY TUNE

Listen to DJ Miles' advice

on machine code sound and your

MTX will soon be warbling

'Thanks for the memory ...'

The Memotech MTX series produces sound using the Texas Instruments SN76489A chip — the integrated circuit used to great effect by BBC Basic. MTX Basic, unfortunately, does not stretch to 14 parameter music envelopes like the Beeb, but it does offer frequency and volume control through the SOUND statement. Continuous music may be played into a sound buffer, but the maximum number of notes per channel allowed is only 256, and this fills more than 12K of RAM.

Machine code, on the other hand, can be used for sound with the advantage of making efficient use of memory. Data must be sent to the chip through output port 6 and strobed in via input port 3. The destination of this data is one of eight registers which control the frequency and volume for the three tone generators and the noise generator (see table 1).

Volume is controlled by passing a nibble to the respective attenuators. A volume level of zero is the loudest, and 14 is the softest. If the bit pattern representing 15 is sent, the sound is switched off. It is not necessary to program the volume of the channel whenever the frequency is changed (as in Basic) but no sound is output if no level has been set.

Frequency is altogether different. The tone channels require ten bits of information to produce a sound. This data is related to the frequency produced by the formula: $\text{Frequency} = 4,000,000 / (32 * \text{Data})$.

A list of notes, frequencies and their equivalent data numbers is given in table 2. It's useful to remember that doubling these numbers produces the same notes one octave lower, and halving these numbers raises the scale by one octave. *Sound* is a handy subroutine which simplifies the task of transferring bytes to the sound processor. It must be entered with Z80 register C containing the chip's destination register and register pair HL containing the data to send. Program 1 demonstrates how it is used. It emits a simple

'laser-gun' noise by repeatedly changing the frequency of a tone channel.

Program 2 proves how simple it is to play tunes with this subroutine by playing a few bars of a familiar tune. It occupies only 200 bytes of memory, which is about 14 times more efficient than the equivalent Basic program using a sound buffer.

The data for the tune is held after label START in the format note, length, note, etc. Any suitable music could be placed at that address so long as it ends with the number 255 which tells the program to return to Basic. There is, of course, no limit to the length of the music, and the program could easily be adjusted to perform some other task during the delay between notes.

Table 2

| Note | Frequency (Hertz) | Data |
|------|-------------------|------|
| C | 264 | 475 |
| C# | 278 | 450 |
| D | 294 | 425 |
| Eb | 312 | 400 |
| E | 334 | 375 |
| F | 358 | 350 |
| F# | 370 | 338 |
| G | 400 | 313 |
| G# | 416 | 300 |
| A | 454 | 275 |
| Bb | 476 | 263 |
| B | 500 | 250 |

Program 1

```
0 CODE
8007 LD HL,0
800A LD C,1
800C CALL SOUND
800F LD C,0
8011 LOOP: DEC HL
8012 CALL SOUND
8015 JP LOOP
8018 SOUND: PUSH BC
8019 PUSH HL
801A LD A,L
801B AND 15
801D RRC C
801F RRC C
8021 RRC C
8023 RRC C
8025 ADD A,C
8026 SET 7,A
8028 BIT 4,A
802A OUT (6),A
802C IN A,(3)
802E JP NZ,BOT
8031 SRL H
8033 RR L
8035 SRL H
8037 RR L
8039 SRL L
803B SRL L
803D LD A,L
803E OUT (6),A
8040 IN A,(3)
8042 BOT: POP HL
8043 POP BC
8044 RET
```

Symbols:
SOUND8018BOT8042
LOOP8011

Program 2

```
0 CODE
8007 LD DE,START
800A LD HL,0
800D CALL VOLUME
8010 TOP: LD A,(DE)
8011 CP 255
8013 JP Z,END
8016 LD H,0
8018 LD L,A
8019 INC DE
801A LD A,(DE)
801B LD B,A
801C INC DE
801D LD C,0
801F CALL SOUND
8022 INC C
8023 INC C
8024 ADD HL,HL
8025 CALL SOUND
8028 INC C
8029 INC C
802A ADD HL,HL
802B CALL SOUND
802E DELAY: PUSH BC
802F LD B,0
8031 LP: PUSH BC
8032 PUSH BC
8033 POP BC
8034 POP BC
8035 DJNZ LP
8037 POP BC
8038 DJNZ DELAY
803A JP TOP
803D END: LD HL,15
8040 CALL VOLUME
8043 RET
8044 VOLUME: LD C,1
8046 CALL SOUND
8049 INC C
804A INC C
804B CALL SOUND
804E INC C
804F INC C
8050 CALL SOUND
8053 RET
8054 SOUND: PUSH BC
8055 PUSH HL
8056 LD A,L
8057 AND 15
8059 RRC C
805B RRC C
805D RRC C
805F RRC C
8061 ADD A,C
8062 SET 7,A
8064 BIT 4,A
8066 OUT (6),A
8068 IN A,(3)
806A JP NZ,BOT
806D SRL H
806F RR L
8071 SRL H
8073 RR L
8075 SRL L
8077 SRL L
8079 LD A,L
807A OUT (6),A
807C IN A,(3)
807E BOT: POP HL
807F POP BC
8080 RET
8081 START: DB 150,128,125,0,113,128,
100,192,89,64,100,128,113,0,131,128,
169,19 2,150,64,131,128,125,0,150,
128,1,5,150,192,169,64,150,128,131,0,
169,128,200
80A8 DB 0,150,128,125,0,113,
128,100,192,89,64,100,128,113,0,131,
128,169,192,150,64,131,128,125,192,
131,64,150,128,156,192,175,64,156,
128,150,0,1,5,150,0
,255
80D2 RET
```

Symbols:
START8081VOLUME8044 SOUND8054DELAY802E
TOP8010END803D LP8031BOT807E

Table 1

| Register | Contents |
|----------|----------------------|
| 0 | Channel 0 Frequency |
| 1 | Channel 0 Volume |
| 2 | Channel 1 Frequency |
| 3 | Channel 1 Volume |
| 4 | Channel 2 Frequency |
| 5 | Channel 2 Volume |
| 6 | Channel 3 Shift Rate |
| 7 | Channel 3 Volume |