



Imagine the fun an adventure game would be if, instead of printing those messages on the cramped (admit it) Vic screen, your computer actually spoke them.

The extra excitement in your lunar lander program if your Vic talked you through the landing. And the added realism in Star Trek when those communiques from Starfleet Command are announced over trans-space radio (your Vic again).

Sounds like fun? All you need is a speech

country was the Currah Chatterbox (for a full review see PCN, issue 2). It is fairly typical of the others in the way you can produce speech on the Vic, and costs around £70. (It is now being marketed as the Hales Speech Synthesiser.)

The immediate novelty of this unit is that when first switched on it turns the Vic into a talking typewriter (guaranteed shock effect if you sneak it past the rest of the family). This has obvious and immediate attractions for partially sighted or blind Vic owners,

# Just say the word....

synthesiser, and the Vic is particularly well catered for in this department. There are at least four companies offering the necessary hardware for under £100.

If your budget runs to it, for several hundred pounds you can buy the de luxe versions. The main difference, apart from cost, is the way in which the Vic (or any micro) synthesises the speech patterns.

The first is not only expensive in terms of cash but also in memory. Linear Predictive Coding (LPC) would require about 16K to store fewer than sixty words. It works by memorising an average word pattern and later repeating it, using the pattern as the model.

The great advantage is that the quality of the speech is very high.

The second, cheaper, version does not store whole words but uses segments of speech called allophones. As an analogy, imagine a normal typewriter and one that didn't use single letters to build up a word, but had to store a whole word on each key . . .

All of the cheap synthesisers use allophones. The advantages are great flexibility and economic memory usage. Against this it must be said that speech quality is not exceptional and it does have that cold, Dalek quality traditionally associated with computer-generated speech.

Allophones are sub-units of phonemes, the main building blocks of speech. A phoneme can be pronounced differently depending where it comes in a word and the context, eg the 's' sounds in 'start' and 'his'. These different versions of the same phoneme are allophones.

In speech synthesis systems there are about 60 to 70 allophones, and generating speech on your Vic is simply a question of stringing them together.

One of the first cheap speech synthesisers in this

but for the rest RUN/STOP and RESTORE will safeguard your hold on sanity.

The joy of the Hales unit is that once you have become accustomed to using the 64 allophones, creating the actual speech is very simple. Using a built-in pronunciation facility you can build up a string of words by trial and error. Then, using the Vic's wonderful screen editor, you can insert a line number and DATA statement at the front.

After that you incorporate in your program the code:  
READ SPS: AS=SPS: SYS 41000  
and the Vic speaks.

You can, of course, use the speech facility directly in your programs. The only requirement is that the string to be spoken must be set to AS immediately before the SYS call. Adding speech routines into your present programs is easy.

Some of the other units on the market vary slightly in the way speech is built up. For example, you may have to use numerical DATA statements to represent the allophones, and write a routine (or use a supplied program) to translate the data into allophones themselves.

The hardware will also vary in different systems. The Hales unit and others come as a plug-in cartridge with all the necessary software in ROM. Others come as separate boxes which connect to the user port.

The choice of system really comes down to personal preference. The cheaper units have a lower quality speech, and some require separate power supplies, which could be a problem given the spaghetti usually associated in running a basic Vic.

Whichever you choose, speech opens up a new area in your computing. There are serious applications; it's also tremendous fun.



## Schizo 64?

Commodore has announced a plug-in cartridge for the 64 which will take advantage of the machine's remarkable built-in sound synthesiser to generate speech in different voices.

Using this development, the 64 will, apparently, be able to generate the voices of a male, female, child . . . whatever you want.

**Hales Speech Synthesiser**, £69.95 inc VAT. Hales Ltd, PO Box 33, Harrowbrook Road, Hinckley, Leics. (0455) 634746.

**Wideband Speakeasy**, £89 plus VAT. Cyber Robotics Ltd, 61 Ditton Walk, Cambridge. (0223) 210675.

**Chatterbox**, £55.45 inc VAT. William Stuart Systems, 44 Bedford Gardens, London W8 7EH. (01) 221 1131.

**Voxbox**, £79.35 inc VAT. Mutek, Quarry Hill, Box, Wilts. Tel (0225) 743289.

**Speech Synthesiser**, £49.95 inc VAT. Adman Electronics, Ripon Way, Harrogate. (0621) 740972.

**Talk-Back**, £24.95 inc VAT. Maplins, PO Box 3, Rayleigh, Essex. (0702) 552911.



# A-Z OF SOUND ON THE BBC



The music capabilities of Acorn's BBC micro are among the best for a computer in its price range.

It can produce creditable versions of everything from Chariots of Fire to a Bach Fugue — providing you spend the time programming the tunes into it.

We'll show you how to build a small amplifier box in the later parts of the Micropaedia, and you'll also be able to send the BBC's sound signal to an outside amplifier. The amplifier will also let you route the sound to your home stereo so it can be recorded on tape and mixed with more conventional music.

The amplifier also has a volume and tone control to give you external control of the computer-generated sounds emanating from your Beeb. It can all be built for less than £10, and you'll soon learn how in upcoming parts of the Sound Micropaedia.

The BBC micro has very versatile sound generating capabilities. But because it can produce so many different types of sound, it is complicated to program, and it's difficult to get exactly the sound you want.

There are four sound channels. These can be programmed separately to produce different sounds, or can be played simultaneously, being added together in the speaker.

The actual sound generation is carried out by sound synthesis hardware independent of the 6502 microprocessor, so the sound production does not tie up the microprocessor and it can be doing other things.

Another feature that helps to prevent the microprocessor from becoming tied up with sound generation is the 'queues'. There are four sound queues, one for each channel, and each queue can hold up to four sound commands. If a sound is already playing on a particular channel and the program reaches a new sound command for that channel, the command is placed in the queue. When one sound finishes, the command at the front of the queue will start automatically, without requiring any further action from the program. If the queue is full when the program comes to a new sound command the program will, however, stop until there is a space.

The simplest form of sound command (on channels

1, 2 and 3 — channel 0 is a little different) is: SOUND C, A, P, D

In this command, C, A, P, and D are four numbers that determine the details of the note produced. C is the channel number, A is the amplitude or loudness, P is the pitch, and D is the duration. The amplitude has 16 levels, the loudest being produced if A is -15 and silence being produced if A is 0. P is a number from 0 to 255, and a change of 1 in P changes the note by one quarter of a semitone. The interval between notes on the chromatic scale is a change of 4 in the value of P, a full octave is a change of 48 in the value of P, and the full range of P from 0 to 255 covers just five octaves. Values of D from 0 to 254 set the duration in twentieths of a second, so putting 20 for D gives a note lasting for one second. If D is -1 or 255 the note will continue until you do something to stop it.

Channel 0 produces noise instead of musical notes. The A and D parameters in a SOUND command for channel 0 work the same way as for channels 1, 2 and 3, but P works differently. If P is 0, 1, or 2 the computer produces high, medium or low frequency 'pulsed' noise, giving a rasping sound, and if P is 3 the pulsed noise has the frequency that has been set on channel 1. If P is 4, 5 or 6 the computer produces high, medium or low frequency 'white' noise, giving a hissing sound,



and if P is 7 the frequency from channel 1 is used.

The command SOUND C, A, P, D is an abbreviated form of the full command, which is SOUND &HSFC, A, P, D. &HSFC is a hexadecimal number, but you do not have to understand hexadecimal to use the sound command — you can treat it as just being four digits strung together with '&' in front.

H can be 0 or 1; if it is 0 the sound will be played in the usual way, but if H is 1 the rest of the command is ignored and the previous sound continues.

S is used to synchronise sounds, so that two or more notes on different channels can be started at exactly the same time. Values of 1, 2 or 3 for S mean that the sound will wait for that number of sounds on other channels with the same value for S, and all the channels will start together when the last command is received. If S is 0 the sound will start as soon as it reaches the front of the queue for its channel.

Putting F as 1 in a SOUND command makes the sound start immediately. It stops the sound being produced on its channel and 'flushes' the channel queue (throws away any notes waiting to be played).

In order to use an ENVELOPE command you put the envelope number for the A in the SOUND command instead of the amplitude, and the amplitude is specified in the ENVELOPE command.

The second parameter in the ENVELOPE command, T, is actually two parameters packed together. If T is a number from 0 to 127 then it will give the length of each step in the envelope in hundredths of a second, and the envelope will auto-repeat (start again so long as a sound is playing that uses the envelope).

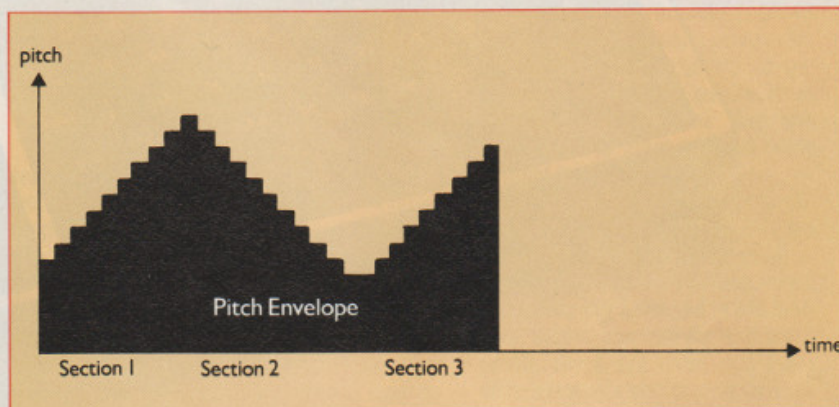
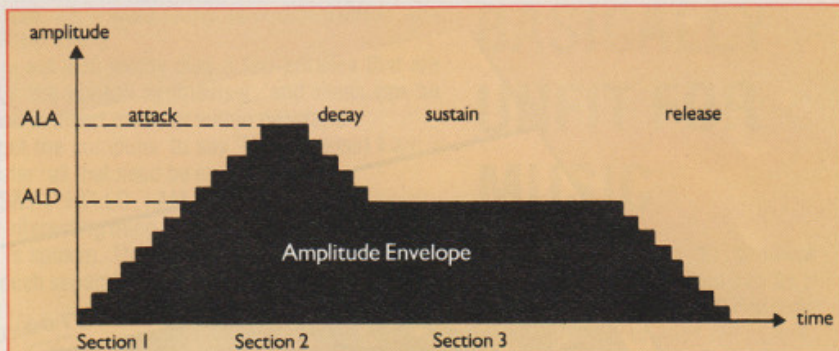
If T is a number from 128 to 255 the length of each step in the envelope will be T - 128 hundredths of a second, and the envelope will stop when it reaches the end of section 3. So if a sound is still playing it will continue as if there were no ENVELOPE command.

To set up a pitch envelope you use the six parameters PI1, PI2, PI3, PN1, PN2, PN3, but you also have to set the amplitude. If you want constant amplitude you set the parameters AA, ALA and ALD to the amplitude you want, AR to minus the amplitude, and AD and AS to 0.

A pitch envelope will change the pitch at a constant rate in each section. The change of pitch may be positive or negative, and is added to the pitch that has been set in the SOUND command. The total change of pitch in section 1 is  $PI1 \times PN1$ , the total change in section 2 is  $PI2 \times PN2$ , and the total change in section 3 is  $PI3 \times PN3$ , so the total change in the whole envelope is  $PI1 \times PN1 + PI2 \times PN2 + PI3 \times PN3$ . If you want an auto-repeat envelope to produce the same sound over and over again you have to make sure that this adds up to 0, so that each repeat starts at the same pitch.

If you want an amplitude envelope with constant pitch you have to set all of PI1, PI2, PI3, PN1, PN2 and PN3 to 0.

In a pitch envelope you have to set the number of steps and the change of pitch per step in each section. An amplitude envelope has to be set up differently. In the first two sections, the 'attack' phase and the 'decay' phase, you set the change of amplitude per step and the target level (the parameters AA and ALA for the attack phase and AD and ALD for the decay phase), and the amplitude changes until it reaches the target level. In the third section, the 'sustain' phase, the amplitude will change at the rate set in the AS parameter until the duration D set in the SOUND command has elapsed. There is a fourth section, the 'release' phase, in the amplitude envelope, and if there is no other sound waiting to be played at the end of the envelope the sound will die away at the rate set by the AR parameter instead of cutting off suddenly.



The two graphs show the BBC's amplitude envelope and pitch envelope, while the tables below — from the BBC user guide — show the implementation of both the SOUND and Envelope keywords and their respective parameters.

The ENVELOPE statement is followed by 14 parameters.

ENVELOPE N,T,PI1,PI2,PI3,PN1,PN2,PN3,AA,AD,AS,AR,ALA,ALD

Parameter	Range	Function
N	1 to 4	Envelope Number
T bits 0-6	0 to 127	Length of each step in hundredths of a second
bit 7	0 or 1	0 = auto-repeat the pitch envelope 1 = don't auto-repeat
PI1	-128 to 127	Change of pitch per step in section 1
PI2	-128 to 127	Change of pitch per step in section 2
PI3	-128 to 127	Change of pitch per step in section 3
PN1	0 to 255	Number of steps in section 1
PN2	0 to 255	Number of steps in section 2
PN3	0 to 255	Number of steps in section 3
AA	-127 to 127	Change of amplitude per step during attack phase
AD	-127 to 127	Change of amplitude per step during decay phase
AS	-127 to 0	Change of amplitude per step during sustain phase
AR	-127 to 0	Change of amplitude per step during release phase
ALA	0 to 126	Target of level at end of attack phase
ALD	0 to 126	Target of level at end of decay phase

The SOUND keyword is followed by four parameters, the first of which consists of 4 hexadecimal digits. Thus

SOUND &HSFC,A,P,D

	Range	Function
H	0 or 1	Continuation
S	0 to 3	Synchronization
F	0 or 1	Flush
C	0 to 3	Channel number
A	-15 to 4	Amplitude or envelope number
P	0 to 255	Pitch
D	1 to 255	Duration





As far as sound, music and speech are concerned, the TI99/4A must be one of the most flexible micros around.

It is also one of the first commercially available machines to offer independent chips for sound generation.

The basic machine has a CALL SOUND command which allows you to specify up to three tones and one noise.

You can also attach a speech synthesiser, which allows you to include speech in programs using TI Extended Basic. On top of this you can buy plug-in ROMpacks specifically geared to the production of sound.

For example:

- The TI Speech Editor can be used in direct mode or as part of a program, but only in conjunction with the speech synthesiser.

- Music Maker allows you to enter music either with traditional notes or by sound graphs.

- Terminal Emulator II allows you to enter speech as entire words or as allophone data. This pack has been released instead of the projected vocabulary packs for the speech synthesiser.

When any of these three packs is used, you cannot use the TI Extended Basic module, but this doesn't present any problems, as the ROMpacks contain all the necessary software.

But you can still experiment with sound even if you don't have one of these ROMpacks.

## MICRO WITH A TEXAS DRAWL

**T**he TI99/4A contains special chips which allow you to produce a wide range of sounds and noises either in direct mode or from programs.

The SOUND command in TI Basic is really a sub-program, and therefore you need to use CALL to get at it. The full syntax for the sound command is: CALL SOUND (d,f1,v1,f2,v2,f3,v3,f4,v4)

In the above, d stands for duration, and is measured in milliseconds, with a range from 1 to 4250. Frequency is controlled by f1-4, as you can see, up to four—three tones and one noise—can be specified at once. If you replace f1-4 with values of 110 to 44733 your machine will produce tones, whereas values of -1 to -8 produce noise.

The volume is controlled by v1-4, with 0 being the loudest and 30 the quietest. For example:

```
CALL SOUND (100, 156, 5)
```

plays a single note (d sharp) of 1/10th second duration, at a volume level of 5. By adding further values for f2, v2 etc you can play up to four notes at once.

You could also use a repeat routine to play a series of notes. But you can specify only one noise within a CALL SOUND command.

So the TI99/4 can play chords over a wide range of frequencies. You can also, by entering minus values for the duration, play notes immediately after one another. The following two programs illustrate this:

```
10 FOR F=20000 TO 25000
20 CALL SOUND (1,F,5)
30 NEXT F
```

```
10 FOR F=20000 TO -2500 STEP 1
20 CALL SOUND (1,F,5)
30 NEXT F
```

When run, you will hear the tone rise in steps à la Sinclair's Spectrum. When the second program is run, the tone will rise smoothly. This is because you have used a minus value.

Frequency values of -1 to -3 produce periodic noises, -5 to -7 produce white noises and -4 and -8 produce periodic white noise that varies when used with a third tone. It is difficult to describe the noises, so try the following program, which prints the noise number and produces the noise for one second.

```
10 FOR N=-1 TO -8 STEP -1
20 PRINT "NOISE NUMBER"; N
30 CALL SOUND (1000,N,5)
40 NEXT N
```

Mixing noises with tones can give some odd results, especially when -4 or -8 are used with other tones.

But there are drawbacks for the sound command. It is not possible to specify individual tone durations within a single CALL SOUND command—only the volume may be specified. And there is no direct control over the finer points of the tones—it is not possible to control the attack or decay of a tone from the CALL SOUND statement. But you can control the sound by supporting Basic commands. Here is a small program which will produce two tones (one high and one low) which will 'overlap' each other.

```
10 LET H=25000
20 LET L=20000
```



```

30 IF H=20000 OR L=25000 THEN STOP
40 CALL SOUND (-10,H,5,L,5)
50 LET H=H-1
60 LET L=L+1
70 GOTO 30

```

Finally, to show how more than one tone may be specified, the next program plays the notes of the tempered scale for the first octave. The data is the appropriate frequency for the note to be played. The same method can be used for chords where up to three values would be specified for each chord.

```

10 FOR N=1 TO 13
20 READ F
30 CALL SOUND (1000,F,5)
40 NEXT N
50 STOP
60 Data 110,117,123,131,139,147,156,165,175,
185,196,208,220

```

## Music

It is theoretically possible to produce good music using the CALL SOUND command in TI Basic but entering different frequency values in large DATA statements is boring. Fortunately, TI provides a solid state software module called Music Maker for entering and playing music.

Music maker makes it possible to create, edit and play music in either traditional or graph modes. The software is menu-driven, with high-resolution graphic displays to help you along. Notes can be entered just as you do on sheet music, and there you have considerable control over what you can enter.

Initially flats, time signatures and speed may be specified. When this is done, two musical staves are displayed (treble and bass) and you may enter the notes by using the keyboard or joystick. The displays are colour coded, which helps the screen look less cluttered.

You can also edit what you have already entered. The editor allows you to repeat parts of a tune without entering all the notes twice. This is done in a similar way to a word processor, where a marker is placed before and after the piece to be copied. Once the section has been copied, it can be amended.

With sound graphs, entering music is quite different. Music is created by entering a number of horizontal lines on a graph, where each line represents a note, and the higher the line on the graph the higher the note to be played. Each note may have a different volume and frequency. Using this method to enter music it is possible to create special effects of different kinds.

You can also edit the music produced by sound graphs, and you can save and load the music onto tape or disk in both modes. Although Music Maker allows you to play music when it is in the TI itself, it doesn't currently have the facility to incorporate the music into Basic programs.

## Speech

Texas Instruments produce a speech synthesiser which is attached to the right of the main TI99/4. To use it within Basic programs it is necessary to use the TI Extended Basic module, which contains extra commands to address the words that are stored into the speech unit.

The syntax for its two sound commands is:

```

CALL SAY (ws,ds)
and
CALL SPGET (ws,rs)

```

In these, ws = word string, rs = return string, and ds = direct string.

There are 400 words plus a few phrases that are used by the speech synthesiser, and these can be operated in direct mode or within programs.

To get the computer to say 'Hello (pause) I am a Computer' all that need be entered is this:

```
CALL SAY ("HELLO, I AM A COMPUTER")
```

The interesting thing here is the delay after the word 'hello' is spoken. The speech synthesiser recognises the speech separator characters listed below.

Symbol	Pause (seconds)
+	0
space	.1
-	.2
,	.3
:	.5
.	.8
'	1.0

The separators can be used to delay speech and are very effective. Multiple delays may be entered into a string of speech:

```
CALL SAY ("HELP,,,GIVE ME
SOME HELP,,PLEASE")
```

One separator character you'll find very useful is the '+', which has a delay of zero seconds. If there is a word that is not in the vocabulary, then it may be possible to string the word together by using two words with the zero delay character. There is no 'therefore' in the TI vocabulary, so entering: CALL SAY ("THERE+FOUR") will solve the problem.

It is not possible to address words in the vocabulary by an ordinal number. The only way to do so is when the word is a number itself. This program shows this:

```

10 FOR N=0 TO 9
20 LET NS=STR$(N)
30 CALL SAY (NS) :: PRINT N
40 NEXT N

```

When run, the words one to nine will be spoken and displayed. We can also see that literals do not have to be used within CALL SAYs. To assign words to strings, we can use CALL SPGET. This doesn't simply move a word to a string variable, but assigns that string variable with the speech pattern of the desired word. To see this look at the following program:

```

10 CALL SPGET ("HELLO",A$)
20 CALL SPGET ("THEIR",B$)
30 CALL SAY A$B$

```

In the call say command, the variables A\$ and B\$ don't contain 'HELLO' and 'THEIR', but instead hold the speech pattern that is used to make up the sounds to produce the required word. To see this yourself, after the program has run enter as a direct command PRINT A\$. You will see a large number of characters printed on the screen.

You don't have to buy TI's Extended Basic module to use the speech synthesiser. Instead you can get the Speech Editor. This lets you do everything mentioned so far and some more.

The speech synthesiser has a limited vocabulary, but with enough experimentation additional words can be made up from existing ones.

Besides being a terminal emulator, the Editor can be used with the speech synthesiser to produce any word. This is achieved by translating text into its allophone equivalent. A little practice is needed to get used to it, but once mastered, text is limitless.

● The next part of the Micropaedia will deal with the Editor in more depth.

# BASICALLY UNFIT FOR MUSIC

Getting your micro to produce something near to music can be a harrowing business. It all depends on what features your micro has, and how easy they are to use.

The Texas Instruments TI-99/4A provides a Basic music command, with which it is possible to get music. The biggest hurdle is to gather a few routines that will enable you to input musical notes without too much bother.

Here you come across a problem. Basic (any dialect), was not designed for the programming of music. Individual manufacturers (including TI), have had to add their own music commands and as a result, there are more and more different music commands for each micro.

To achieve truly easy musical programming, a dedicated programming language is needed. Some of the sophisticated micro/synthesisers do have their own language, but again, the manufacturers have not come up with a standard.

The TI owner has to put up with the CALL SOUND command. This comes in the format of:

```
CALL SOUND (duration, tone1,
vol1, tone2, vol2, tone3, vol3,
noise1, vol4).
```

Looking at the syntax, we can see that up to three tones and one of eight different sounds may be specified.

The disadvantage of this command is the duration. The duration in each CALL SOUND applies to all the tones (or noise). This means that it is not possible to have one tone longer or shorter than the next within the same CALL SOUND.

Each note on the tempered scale has a corresponding frequency. For example, the frequency for middle C is 261.63Hz.

Like most other micros that can produce sound, the TI will not play the frequency to floating point accuracy. What happens is that the argument is rounded up to the nearest integer. This is no problem, however as you will not be able to detect such a small difference.

Next week we'll give you a chart containing musical scales and their corresponding notes on the TI 99/4A, and give you further music programming tips on the Texas.



# SOUND: PART 2

## 'SOUND' Command

```

7000          2      PRT
7000          10 @START EQU *
7000 108E7E67      20 LDY #32359
7004 8E0833        20 LDX #32819
7007 108F0121      20 STY 289
7008 A688          30 @LOOP LDA ,X+
7008 A7A0          30 STA ,Y+
700F 8C81CA        30 CMPX #33226
7012 25F7          30 BLO @LOOP
7014 8E7F88        40 LDX #32648
7017 8F0123        40 STX 291
701A 8E7D21        50 LDX #8SOUND
701D 8F7FCC        50 STX 32716
7020 39           60 RTS
7021          70 @SOUND EQU *
7021 3436          80 PSHS A,B,X,Y
7023 8D8E93        90 JSR 36483
7026 2715          90 BEQ @ERR
7028 8F0144        100 STX 324
702B 3536          110 PULS A,B,X,Y
702D 8089AA        110 JSR 35242
7030 3436          110 PSHS A,B,X,Y
7032 8D8E93        120 JSR 36483
7035 8F0146        120 STX 326
7038 8D00A5        130 JSR 165
703B 2707          130 BEQ @NDERROR
703D 3536          140 @ERR PULS A,B,X,Y
703F C082          140 LDB #2
7041 7E8344        140 JMP 33604
7044 86FF03        150 @NDERR LDA 65283
7047 84FE          150 ANDA #254
7049 87FF03        150 STA 65283
704C 86FF23        160 LDA 65315
704F 8A08          160 ORA #8
7051 87FF23        160 STA 65315
7054 86FF01        170 LDA 65281
7057 84F7          170 ANDA #247
7059 87FF01        170 STA 65281
705C 10E0146       180 LDY 326
7060 86FD          190 @AGAIN LDA #253
7062 87FF20        190 STA 65312
7065 8E0144        200 LDX 324
7068 301F          210 @WAIT LEAX -1,X
706A 26FC          210 BNE @WAIT
706C 7FFF20        220 CLR 65312
706F 8E0144        230 LDX 324
7072 301F          240 @WAIT1 LEAX -1,X
7074 26FC          240 BNE @WAIT1
7076 313F          250 LEAY -1,Y
7078 26E6          250 BNE @AGAIN
707A 86FF23        260 LDA 65315
707D 84F7          260 ANDA #247
707F 87FF23        260 STA 65315
7082 86FF03        270 LDA 65283
7085 8A01          270 ORA #1
7087 87FF03        270 STA 65283
708A 3536          280 PULS A,B,X,Y
708C 39           280 RTS
708D          290 END @START

```

The new version of the Dragon's sound command. Once loaded it will operate in the same way as any other Basic statement and will remain in memory, even after a NEW command, until the machine is switched off. It offers more flexibility than the built-in Sound command although the syntax is different. See main text for details.

Following the first venture in to Dragon Sound, this week we can move on to machine code. The chip used is called an 'analog multiplexer', and this controls which sound source is selected at any time. The options are: 6 BIT (Sound/Play), CASSETTE (Audio on/off) or CARTRIDGE (Sound chip in cartridge).

There are three connections to this chip that are of interest: SEL 1, SEL 2, and BDA. These are addressed to bit 3 of addresses 65281, 65283, and 65315 respectively. The table shows the arrangement for any particular source.

SEL 1	SEL 2	BDA	Sound source selected
0	0	1	6 BIT D/A
1	0	1	CASSETTE
0	1	0	CARTRIDGE (ext. chip)
0	0	0	SILENT

As SEL 2 is normally zero on the Dragon we can neglect it. The most interesting is 6 BIT. For this option, the sound comes from a D/A converter (the same D/A converter used for the joysticks). This is addressed as the six most significant bits of location 65312. The upshot of all this is that with the multiplexer set up for 6 BIT, storing a number at location 65312 produces a click to the TV speaker — the bigger the number, the higher the click.

Program Sound Demo One demonstrates how to produce sound. Those of you with assemblers can enter the listing directly, those without can use the Basic loader program to enter the hexadecimal codes on the left. The first column is the address — all demo programs start at 32000. Type EXEC 32000 to hear the demo.

As well as setting up the sound chip, the program disables interrupts during sound production. In the last article I explained the difficulties of producing high notes without interference — disabling interrupts prevents this interference.

The program is very simple. In essence, it produces

# DRAGON SOUND CODE

## WIERD INTERRUPT SOUND

```

7000          20      PRT
7000 86FF23        30 @START LDA 65315
7003 8A08          30 ORA #8
7005 87FF23        30 STA 65315
7008 FC0112        40 @LOOP LDD 274
700B 30           40 MUL
700C FBFF20        40 ADDB 65312
700F 77FF20        40 STB 65312
7012 86FF00        50 LDA 65280
7015 8A08          50 ORA #128
7017 81FF          50 CMPA #255
7019 27ED          50 BEQ @LOOP
701B 86FF23        60 LDA 65315
701E 84F7          60 ANDA #247
7020 87FF23        60 STA 65315
7023 39           70 RTS
7024          100 END @START

```

```

5 'MACHINE CODE ENTRY FOR USERS WITHOUT ASSEMBLERS
10 CLEAR200,31999
20 X=32000:POKE 157,125:POKE 158,0
30 PRINTHEX(X); " "; LINE INPUT A$
40 IF A$="END" THEN 90
50 FOR I=1 TO LEN(A$) STEP 2
60 Z=MID$(A$,I,2):A=VAL("&H"+Z$)
70 POKE X,A:X=X+1
80 NEXT I:GOTO30
90 PRINT:PRINT"TYPE 'EXEC' TO TEST":END

```

## SOUND DEMO ONE

```

7000          10      PRT
7000 3432          20 @START PSHS A,X,Y
7002 86FF23        20 LDA 65315
7005 8A08          20 ORA #8
7007 87FF23        20 STA 65315
700A 86FF03        30 LDA 65283
700D 84FE          30 ANDA #254
700F 87FF03        30 STA 65283
7012 8E05DC        40 LDX #1500
7015 86FD          50 @LOOP LDA #253
7017 87FF20        50 STA 65312
701A 8D1C          60 BSR @DELAY
701C 7FFF20        60 CLR 65312
701F 8D17          60 BSR @DELAY
7021 301F          70 LEAX -1,X
7023 26F0          80 BNE @LOOP
7025 86FF03        90 LDA 65283
7028 8A01          90 ORA #1
702A 87FF03        90 STA 65283
702D 86FF23        100 LDA 65315
7030 84F7          100 ANDA #247
7032 87FF23        100 STA 65315
7035 3532          110 PULS A,X,Y
7037 39           110 RTS
7038 1F12          120 @DELAY TFR X,Y
703A 313F          130 @LB1 LEAY -1,Y
703C 26FC          140 BNE @LB1
703E 39           150 RTS
703F          160 END @START

```

At bottom left is the Basic loader program if you don't have a monitor or assembler. Above it, the sound program demonstrates the use of the interrupt routine in sound effects. Sound Demo One includes the technique of disabling the interrupt preventing interference in high pitched sounds.



C, waits a certain time and then loops round to do  
click, so producing a note. The delay gets  
larger and so the note rises.

Using machine code means that much better sound  
effects can be obtained. The Laser Sound program  
copies the idea of program one, changes it so  
the note starts very high and comes down in pitch  
in a fraction of a second.

The term 'white noise' is often used when  
describing a particular kind of sound and here we  
will apply the term for sound effects such as  
explosions or gunshots.

On the Dragon it is not easy to produce white noise.  
What is needed is continuously varying volumes to be  
fed into D/A. A good source of continuously  
varying numbers is the BASIC ROM memory which  
starts at 32768.

The Gunshot example program takes numbers from  
the ROM and puts them in the D/A. By fading down the  
volume (by logically ANDing the data) a surprisingly  
realistic gunshot sound is obtained. A suitable place to  
store it is location 47000. Location 327 stores the  
pitch to be used.

When writing relocatable machine code programs it  
is useful to know that locations 310 and 327 can be  
used as data stores without fear of over-writing. These  
locations are used by Basic to store the entry  
addresses for USR1 to USR9, but due to a ROM bug,  
USR0 works.

## The Sound Command

With the Sound command the interference on high  
pitch and short range of frequency is not as good as it  
could be.

The program called Sound Command rewrites  
the Sound command so that there is no interference  
so that the first number can be between 1-65535  
(instead of 1-255). The second number can also be in  
this range to give very short or long notes.

This second number is no longer a set time but now

refers to the number of cycles the command will take.  
Therefore, for a high-pitched note, the length should  
be high, but for a low note the length should be short.

## Tailored programming

The way in which this program is written should  
interest anyone who would like to change any of the  
Basic commands to their own taste. Address 289  
contains the start of the Basic command table and  
location 291 contains the start of the command vector  
table. By moving these down into RAM and changing  
the vector for the Sound command we can effectively  
rewrite the command.

Lines 10 to 60 do this by first copying the ROM to  
RAM and then changing the necessary locations to  
point to the new sound command which starts at line  
70.

The new routine makes some subroutine calls to the  
ROM which should be explained. Line 90 gets the first  
number following the command in the 'X' register.  
Line 110 searches for a comma which separates the  
two numbers, and line 120 gets the second number for  
the command. If the first number (the pitch) is zero, a  
syntax error message is generated by @ERR in line  
140, the 'B' register holding the code of the error.

The rest of the routine is just a sound generating  
loop similar to that in the first program.

When you have typed in the program (either with the  
loader program or an assembler) type EXEC 32000. OK  
should appear immediately and the new sound  
command will now work. You can use it as normal in  
Basic.

Note that as the Basic vectors are now in RAM no  
machine code should be loaded above 32000 or the  
Basic will crash. To return to normal type POKE 113,0  
and press RESET, this will perform a cold start.

## Speaking up

The program listed in Basic is an attempt at speech

synthesis on the Dragon. Type in the program and RUN  
it. After a short delay — while the machine code is  
loaded from the DATA statements — a message will  
appear asking you whether you wish to Playback or  
Listen. Place a cassette in the recorder set up as if to  
load a program and press play.

The cassette should have some speech or simple  
music on it. Then enter 'L' and press enter. The  
cassette will play into the computer for about eight  
seconds and then the program will try to synthesise  
what it has just heard. Press P to hear it again. Don't  
expect miracles, the voice sounds like it is coming  
from the moon, but should be understandable. This  
should have given you a basic idea of what your  
micro is capable of, but you can still go on to greater  
things.

If you just get interference try adjusting the volume  
on the cassette; it will need to be higher than normal.  
As the program is so short, the quality will not be that of  
something like 'Computavoice', the synthesiser  
available from Dragon, but the system is entirely  
different.

## Digital dialects

Whereas Computavoice has a set number of sounds  
which can be put together to form words like a Speak  
'n' Spell machine, this short program records the  
sound digitally in the memory. 27K is used for eight  
seconds to get this quality which means that people's  
accents should be clearly audible. Also music,  
although somewhat distorted, can be synthesised.

The final program is just to prove that interrupts can  
be a good thing in sound, as well as causing  
interference. The program takes the value of the  
TIMER, does a bit of maths to it and then stores in  
location 65312. The result is a very weird piece of  
'music' (which incidentally will not repeat for 20  
minutes) which you will either love or hate.

If you do hate it take it as an encouragement to do  
better yourself.

# MODIFIED

### GUNSHOT SOUND EFFECT

```

7000      10      PRT
7000      20      @START EQU *
7000      30      PSHS A,X
7000      40      LDA 65315
7000      40      ORA #8
7000      40      STA 65315
7000      40      LDX #1
7000      50      @LOOP LDA #253
7000      60      STA 65312
7000      70      BSR @DELAY
7000      80      CLR 65312
7000      90      BSR @DELAY
7000      100     LEAX 1,X
7000      110     CMPL #155
7000      120     BLO @LOOP
7000      130     LDA 65315
7000      140     ANDA #247
7000      150     STA 65315
7000      160     PULS A,X,Y
7000      170     RTS
7000      180     END @START
7000      190
7000      200
7000      210
7000      220
7000      230
7000      240
7000      250
7000      260
7000      270
7000      280
7000      290
7000      300
7000      310
7000      320
7000      330
7000      340
7000      350
7000      360
7000      370
7000      380
7000      390
7000      400
7000      410
7000      420
7000      430
7000      440
7000      450
7000      460
7000      470
7000      480
7000      490
7000      500
7000      510
7000      520
7000      530
7000      540
7000      550
7000      560
7000      570
7000      580
7000      590
7000      600
7000      610
7000      620
7000      630
7000      640
7000      650
7000      660
7000      670
7000      680
7000      690
7000      700
7000      710
7000      720
7000      730
7000      740
7000      750
7000      760
7000      770
7000      780
7000      790
7000      800
7000      810
7000      820
7000      830
7000      840
7000      850
7000      860
7000      870
7000      880
7000      890
7000      900
7000      910
7000      920
7000      930
7000      940
7000      950
7000      960
7000      970
7000      980
7000      990
7000      1000
7000      1010
7000      1020
7000      1030
7000      1040
7000      1050
7000      1060
7000      1070
7000      1080
7000      1090
7000      1100
7000      1110
7000      1120
7000      1130
7000      1140
7000      1150
7000      1160
7000      1170
7000      1180
7000      1190
7000      1200
7000      1210
7000      1220
7000      1230
7000      1240
7000      1250
7000      1260
7000      1270
7000      1280
7000      1290
7000      1300
7000      1310
7000      1320
7000      1330
7000      1340
7000      1350
7000      1360
7000      1370
7000      1380
7000      1390
7000      1400
7000      1410
7000      1420
7000      1430
7000      1440
7000      1450
7000      1460
7000      1470
7000      1480
7000      1490
7000      1500
7000      1510
7000      1520
7000      1530
7000      1540
7000      1550
7000      1560
7000      1570
7000      1580
7000      1590
7000      1600
7000      1610
7000      1620
7000      1630
7000      1640
7000      1650
7000      1660
7000      1670
7000      1680
7000      1690
7000      1700
7000      1710
7000      1720
7000      1730
7000      1740
7000      1750
7000      1760
7000      1770
7000      1780
7000      1790
7000      1800
7000      1810
7000      1820
7000      1830
7000      1840
7000      1850
7000      1860
7000      1870
7000      1880
7000      1890
7000      1900
7000      1910
7000      1920
7000      1930
7000      1940
7000      1950
7000      1960
7000      1970
7000      1980
7000      1990
7000      2000
7000      2010
7000      2020
7000      2030
7000      2040
7000      2050
7000      2060
7000      2070
7000      2080
7000      2090
7000      2100
7000      2110
7000      2120
7000      2130
7000      2140
7000      2150
7000      2160
7000      2170
7000      2180
7000      2190
7000      2200
7000      2210
7000      2220
7000      2230
7000      2240
7000      2250
7000      2260
7000      2270
7000      2280
7000      2290
7000      2300
7000      2310
7000      2320
7000      2330
7000      2340
7000      2350
7000      2360
7000      2370
7000      2380
7000      2390
7000      2400
7000      2410
7000      2420
7000      2430
7000      2440
7000      2450
7000      2460
7000      2470
7000      2480
7000      2490
7000      2500
7000      2510
7000      2520
7000      2530
7000      2540
7000      2550
7000      2560
7000      2570
7000      2580
7000      2590
7000      2600
7000      2610
7000      2620
7000      2630
7000      2640
7000      2650
7000      2660
7000      2670
7000      2680
7000      2690
7000      2700
7000      2710
7000      2720
7000      2730
7000      2740
7000      2750
7000      2760
7000      2770
7000      2780
7000      2790
7000      2800
7000      2810
7000      2820
7000      2830
7000      2840
7000      2850
7000      2860
7000      2870
7000      2880
7000      2890
7000      2900
7000      2910
7000      2920
7000      2930
7000      2940
7000      2950
7000      2960
7000      2970
7000      2980
7000      2990
7000      3000
7000      3010
7000      3020
7000      3030
7000      3040
7000      3050
7000      3060
7000      3070
7000      3080
7000      3090
7000      3100
7000      3110
7000      3120
7000      3130
7000      3140
7000      3150
7000      3160
7000      3170
7000      3180
7000      3190
7000      3200
7000      3210
7000      3220
7000      3230
7000      3240
7000      3250
7000      3260
7000      3270
7000      3280
7000      3290
7000      3300
7000      3310
7000      3320
7000      3330
7000      3340
7000      3350
7000      3360
7000      3370
7000      3380
7000      3390
7000      3400
7000      3410
7000      3420
7000      3430
7000      3440
7000      3450
7000      3460
7000      3470
7000      3480
7000      3490
7000      3500
7000      3510
7000      3520
7000      3530
7000      3540
7000      3550
7000      3560
7000      3570
7000      3580
7000      3590
7000      3600
7000      3610
7000      3620
7000      3630
7000      3640
7000      3650
7000      3660
7000      3670
7000      3680
7000      3690
7000      3700
7000      3710
7000      3720
7000      3730
7000      3740
7000      3750
7000      3760
7000      3770
7000      3780
7000      3790
7000      3800
7000      3810
7000      3820
7000      3830
7000      3840
7000      3850
7000      3860
7000      3870
7000      3880
7000      3890
7000      3900
7000      3910
7000      3920
7000      3930
7000      3940
7000      3950
7000      3960
7000      3970
7000      3980
7000      3990
7000      4000
7000      4010
7000      4020
7000      4030
7000      4040
7000      4050
7000      4060
7000      4070
7000      4080
7000      4090
7000      4100
7000      4110
7000      4120
7000      4130
7000      4140
7000      4150
7000      4160
7000      4170
7000      4180
7000      4190
7000      4200
7000      4210
7000      4220
7000      4230
7000      4240
7000      4250
7000      4260
7000      4270
7000      4280
7000      4290
7000      4300
7000      4310
7000      4320
7000      4330
7000      4340
7000      4350
7000      4360
7000      4370
7000      4380
7000      4390
7000      4400
7000      4410
7000      4420
7000      4430
7000      4440
7000      4450
7000      4460
7000      4470
7000      4480
7000      4490
7000      4500
7000      4510
7000      4520
7000      4530
7000      4540
7000      4550
7000      4560
7000      4570
7000      4580
7000      4590
7000      4600
7000      4610
7000      4620
7000      4630
7000      4640
7000      4650
7000      4660
7000      4670
7000      4680
7000      4690
7000      4700
7000      4710
7000      4720
7000      4730
7000      4740
7000      4750
7000      4760
7000      4770
7000      4780
7000      4790
7000      4800
7000      4810
7000      4820
7000      4830
7000      4840
7000      4850
7000      4860
7000      4870
7000      4880
7000      4890
7000      4900
7000      4910
7000      4920
7000      4930
7000      4940
7000      4950
7000      4960
7000      4970
7000      4980
7000      4990
7000      5000
7000      5010
7000      5020
7000      5030
7000      5040
7000      5050
7000      5060
7000      5070
7000      5080
7000      5090
7000      5100
7000      5110
7000      5120
7000      5130
7000      5140
7000      5150
7000      5160
7000      5170
7000      5180
7000      5190
7000      5200
7000      5210
7000      5220
7000      5230
7000      5240
7000      5250
7000      5260
7000      5270
7000      5280
7000      5290
7000      5300
7000      5310
7000      5320
7000      5330
7000      5340
7000      5350
7000      5360
7000      5370
7000      5380
7000      5390
7000      5400
7000      5410
7000      5420
7000      5430
7000      5440
7000      5450
7000      5460
7000      5470
7000      5480
7000      5490
7000      5500
7000      5510
7000      5520
7000      5530
7000      5540
7000      5550
7000      5560
7000      5570
7000      5580
7000      5590
7000      5600
7000      5610
7000      5620
7000      5630
7000      5640
7000      5650
7000      5660
7000      5670
7000      5680
7000      5690
7000      5700
7000      5710
7000      5720
7000      5730
7000      5740
7000      5750
7000      5760
7000      5770
7000      5780
7000      5790
7000      5800
7000      5810
7000      5820
7000      5830
7000      5840
7000      5850
7000      5860
7000      5870
7000      5880
7000      5890
7000      5900
7000      5910
7000      5920
7000      5930
7000      5940
7000      5950
7000      5960
7000      5970
7000      5980
7000      5990
7000      6000
7000      6010
7000      6020
7000      6030
7000      6040
7000      6050
7000      6060
7000      6070
7000      6080
7000      6090
7000      6100
7000      6110
7000      6120
7000      6130
7000      6140
7000      6150
7000      6160
7000      6170
7000      6180
7000      6190
7000      6200
7000      6210
7000      6220
7000      6230
7000      6240
7000      6250
7000      6260
7000      6270
7000      6280
7000      6290
7000      6300
7000      6310
7000      6320
7000      6330
7000      6340
7000      6350
7000      6360
7000      6370
7000      6380
7000      6390
7000      6400
7000      6410
7000      6420
7000      6430
7000      6440
7000      6450
7000      6460
7000      6470
7000      6480
7000      6490
7000      6500
7000      6510
7000      6520
7000      6530
7000      6540
7000      6550
7000      6560
7000      6570
7000      6580
7000      6590
7000      6600
7000      6610
7000      6620
7000      6630
7000      6640
7000      6650
7000      6660
7000      6670
7000      6680
7000      6690
7000      6700
7000      6710
7000      6720
7000      6730
7000      6740
7000      6750
7000      6760
7000      6770
7000      6780
7000      6790
7000      6800
7000      6810
7000      6820
7000      6830
7000      6840
7000      6850
7000      6860
7000      6870
7000      6880
7000      6890
7000      6900
7000      6910
7000      6920
7000      6930
7000      6940
7000      6950
7000      6960
7000      6970
7000      6980
7000      6990
7000      7000
7000      7010
7000      7020
7000      7030
7000      7040
7000      7050
7000      7060
7000      7070
7000      7080
7000      7090
7000      7100
7000      7110
7000      7120
7000      7130
7000      7140
7000      7150
7000      7160
7000      7170
7000      7180
7000      7190
7000      7200
7000      7210
7000      7220
7000      7230
7000      7240
7000      7250
7000      7260
7000      7270
7000      7280
7000      7290
7000      7300
7000      7310
7000      7320
7000      7330
7000      7340
7000      7350
7000      7360
7000      7370
7000      7380
7000      7390
7000      7400
7000      7410
7000      7420
7000      7430
7000      7440
7000      7450
7000      7460
7000      7470
7000      7480
7000      7490
7000      7500
7000      7510
7000      7520
7000      7530
7000      7540
7000      7550
7000      7560
7000      7570
7000      7580
7000      7590
7000      7600
7000      7610
7000      7620
7000      7630
7000      7640
7000      7650
7000      7660
7000      7670
7000      7680
7000      7690
7000      7700
7000      7710
7000      7720
7000      7730
7000      7740
7000      7750
7000      7760
7000      7770
7000      7780
7000      7790
7000      7800
7000      7810
7000      7820
7000      7830
7000      7840
7000      7850
7000      7860
7000      7870
7000      7880
7000      7890
7000      7900
7000      7910
7000      7920
7000      7930
7000      7940
7000      7950
7000      7960
7000      7970
7000      7980
7000      7990
7000      8000
7000      8010
7000      8020
7000      8030
7000      8040
7000      8050
7000      8060
7000      8070
7000      8080
7000      8090
7000      8100
7000      8110
7000      8120
7000      8130
7000      8140
7000      8150
7000      8160
7000      8170
7000      8180
7000      8190
7000      8200
7000      8210
7000      8220
7000      8230
7000      8240
7000      8250
7000      8260
7000      8270
7000      8280
7000      8290
7000      8300
7000      8310
7000      8320
7000      8330
7000      8340
7000      8350
7000      8360
7000      8370
7000      8380
7000      8390
7000      8400
7000      8410
7000      8420
7000      8430
7000      8440
7000      8450
7000      8460
7000      8470
7000      8480
7000      8490
7000      8500
7000      8510
7000      8520
7000      8530
7000      8540
7000      8550
7000      8560
7000      8570
7000      8580
7000      8590
7000      8600
7000      8610
7000      8620
7000      8630
7000      8640
7000      8650
7000      8660
7000      8670
7000      8680
7000      8690
7000      8700
7000      8710
7000      8720
7000      8730
7000      8740
7000      8750
7000      8760
7000      8770
7000      8780
7000      8790
7000      8800
7000      8810
7000      8820
7000      8830
7000      8840
7000      8850
7000      8860
7000      8870
7000      8880
7000      8890
7000      8900
7000      8910
7000      8920
7000      8930
7000      8940
7000      8950
7000      8960
7000      8970
7000      898
```



## SOUND: PART 2

**W**hy the Apple? In its short lifetime, Stephen Wozniak's brainchild has become a primary focus for budding computer-musicians. Certainly, the Apple's natural capabilities for sound production are, shall we say, limited; but its capability for expansion seemingly not so.

The Apple's eight peripheral interface slots (seven on the IIe) would provide an adequate material answer, but let us expand the question: Why are some computers more artistic than others?

We are informed that programming is an art, although attempts to define art are usually considered imprecise. And yet a computer is a machine of great precision, so we must avoid wishy-washy 'intangibles' in our definition: Art is the pursuit of perfection. *Voila*. And the original Apple II was achieved in pursuit of perfection, which is why it has been rewarded with an inherent empathy on the part of artists — especially musicians.

It has, of course, also found favour in the world, as Bach would say, of cabbage and beet, where perfection is considered improved with the addition of each zero (and where, by all accounts, its straightforwardness must come as a refreshing blast).

Many musicians and composers bought Apples even before 'music peripherals' were available, so there is something about the machine itself which harmonises with the needs of the creative spirit.

The Apple's split-second timing has proved a positive advantage. It also proved that the requirements of musical performance are changing — to be made aware of microseconds may not be an emotional experience, but its power is undeniable to those who have felt it. Computer-generated sound holds many keys to the future, not least in the way in which we experience the nature of sound itself.

But what is computer-generated sound? Well, there are two main uses to which a musical computer can be put (leaving aside ancillary operations like multi-track mixing, delay/echo effects, etc) which are *a*) making the notes and *b*) playing the notes.

We will consider *b*) first, as it is easier. In its simplest form, the computer only has to decide when to play each note, and how long to continue it. It does this by comparing the time for each note to be played with the time 'now', and waiting, if necessary. Ideally, the computer would then be able to control the course of each note, its attack and decay characteristics, and any changes in pitch or loudness. We would like the computer to have some ability to change the quality, or timbre, of the sound, but this is more difficult (on analog synthesisers this is normally achieved by using a filter) and presupposes *a*), to which we pass, pausing only to suggest that the most important criterion for *b*) is — how does it respond to your creative needs?

Try, before choosing a system, programming in a short piece. See how the system copes with changes. With mistakes. With repeated sections. With rests.

Unless your system has a (musical) keyboard, this represents performance, and creativity demands flexibility.

The generation of sound itself is more complex, but a little theory should suffice. There are three main kinds of output from Apple-based systems, in which the Apple itself (or a card/cards inside it) generates the sound (in the other, the Apple is used to control existing synthesiser(s), such as those made by Arp, Roland and Yamaha). They are:

i) square waves: easy for computers to synthesise (a one and a zero draw the picture) and fairly pleasant on the ear, being not unlike the tone of a clarinet.

ii) Pre-shaped waveforms produced by special sound-generating chips. These waves may be square,



## GIVE US AN A FOR APPLE

triangular, and possibly sinusoidal (and usually there is also a white noise generator, for swoosh-bang). Neither of these represents any advance on the analog synthesisers of ten or 15 years ago, except perhaps in the area of tuning stability, but more importantly, none can claim to be truly digitally-generated sound.

iii) True digitally-generated sound — a sampling method, whereby the periodic level, or amplitude, of a wave is read out through a digital-to-analog converter (DAC), and thence to your amp, speakers, etc, at fixed intervals of time. Sampling is the technique used by digital tape recorders, already acclaimed for their high accuracy of reproduction.

There are two possibilities with sampling: you work it out 'now', sample by sample, or you work it out in advance, store it, and look it up.

Suffice to say that today's micros are not fast enough to do the real-time calculation necessary for, let's say, a digital filter; so they store the information for the shape of a wave in, say, 256 bytes of memory, which is called a look-up table. Who said machine-code was difficult!

Actually, there is only one music system available on a microcomputer (the Apple!) which uses this genuine digital synthesis technique. So this one (the Mountain Hardware music system) represents by far the most serious attempt to make the Apple into a true digital synthesiser, especially with the addition of the Alpha Syntauri keyboard/software which makes use of its hardware. It is, after all, something of a minor miracle that such a system should be available on any 8-bit micro, and it is fitting that the Apple should be its host.

The venerable Apple much loved by both micro and music enthusiasts. If it is too much to claim that musicians' perceptions have been altered by possession of the Apple, it is true that many have found it an extremely useful tool in the art. The Apple remains the only micro to offer true digital sound synthesis. Largely because of its eight versatile expansion slots, but also because of its inventive and imaginative owners, it has proved easy to design and implement flexible and sophisticated software/hardware systems dedicated to complex and demanding tasks such as real-time sound-generation. Such systems use advanced techniques such as Direct Memory Access (DMA), as well as state-of-the-art technology in the form of 12-bit digital to analog converters, or even 16-bit D to A chips.

Contributors: Ted Ball, Brian Cadge, David Janda, Francis Monkman and Peter Worlock  
Illustration: Adam Peters  
Design: Nigel Wingrove

## NEXT WEEK

In part three of our opus on sound, there'll be a close examination of speech on the TI99/4A, a look at the history of sound on the Apple, and what you can do with it today, programming examples for your Atari, a rundown of the sound commands on the BBC and yet more on making noise with the Dragon.

Looking further ahead, there'll be more for these micros, plus musical programming on the Vic 20.