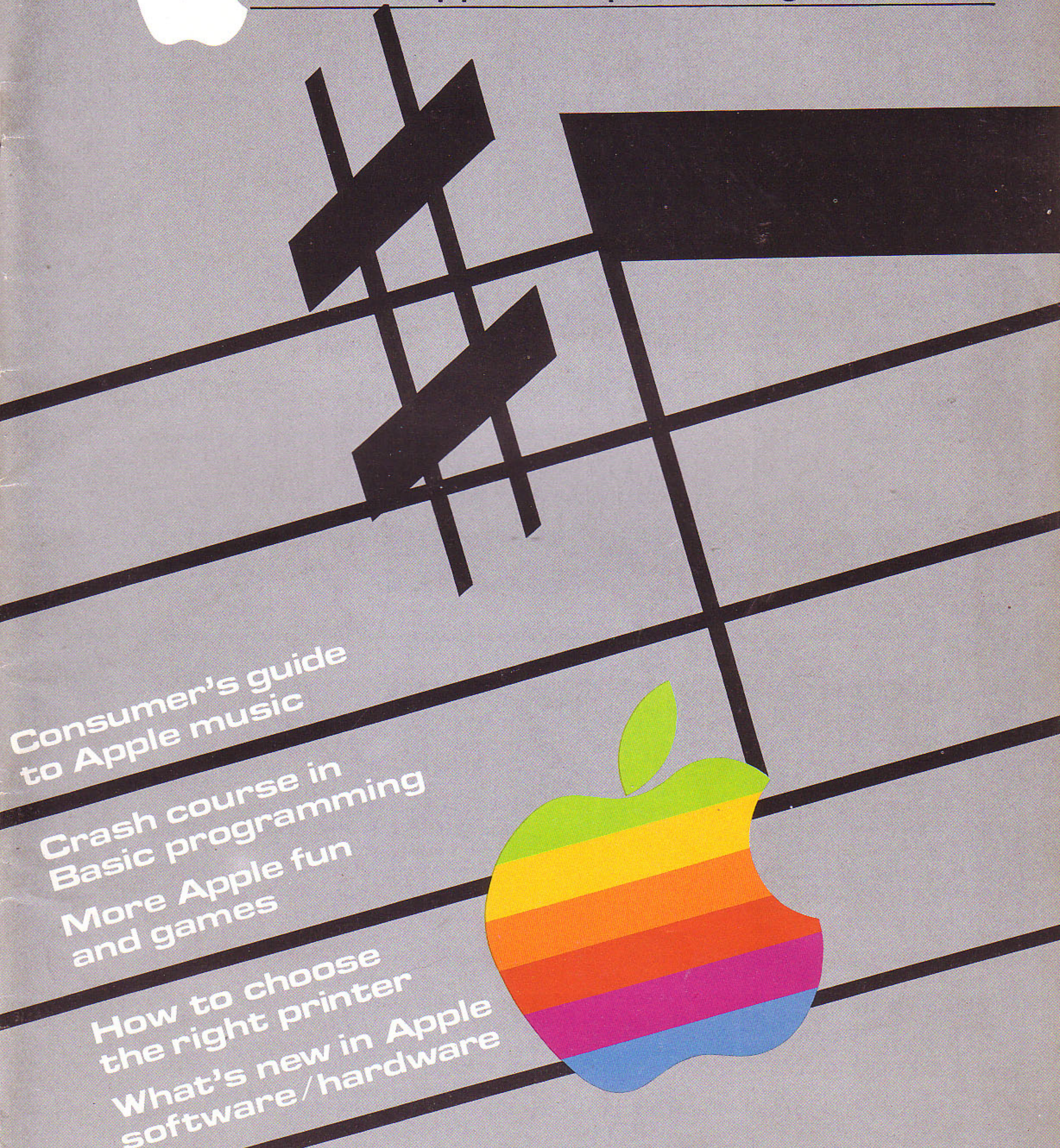
Windfall

No. 3 September 1981 £1

The Apple computer magazine



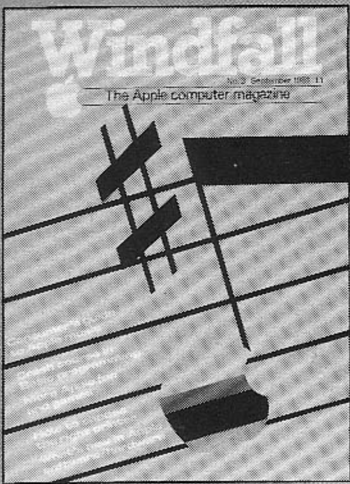
Consumer's guide to Apple music

Crash course in Basic programming

More Apple fun and games

How to choose the right printer

What's new in Apple software/hardware



No 3 September 1981

LISTING

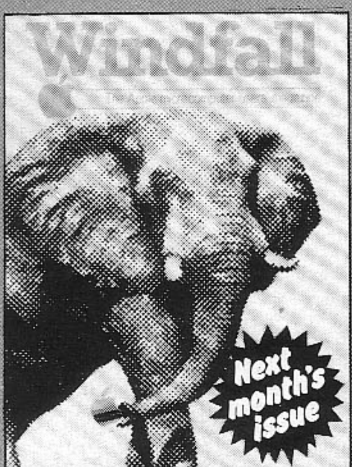
Managing Editor
Derek Meakin
Editor
David Chadwick
Art Editor
Peter Glover
Technical Editors
Peter Brameld
Max Parrott
John Watson
Advertisement Manager
John Riding

Tel: 061-456 8383 (Editorial)
061-456 8500 (Advertising)
061-456 8353 (24 hour answering service)
Telex: 665350 EUROPA-G

Published by Europress Ltd, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Subscription rates:
UK – £12 a year for 12 issues, post free.
Overseas – £20 a year.

Writing for Windfall: Articles and programs relating to the Apple are welcome. Articles should preferably be typed or computer-printed, using double spacing. Unsolicited manuscripts, discs, etc, should be accompanied by a self-addressed stamped envelope, otherwise their return cannot be guaranteed.



A consumer's guide to Apple music

THE Apple II has attracted a quite remarkable amount of musical applications over the last year or so, both from manufacturers of hardware and software and from musicians tentatively dipping their feet into the techniques of digital synthesis.

Some of the eight music synthesizers now available (in the States, if not here) are close relatives to the bang and crash fraternity of sound boards, but others offer as much as any professional musician could realistically hope for with the present generation of 8-bit processors.

Music synthesis must be one of the most effective – in terms of enduring pleasure – and dramatic (soothing the savage beast, etc.) uses of microcomputers, and certainly seems likely to be an area of intensive development for any society that has access to such technology and follows the predicted course of increased leisure time.

Encoding music, whether on paper with traditional notation via alphanumeric keyboards using a music composition language or whatever, utilizes a great deal of labour-intensive human input, and it's this realisation that makes the use of number-crunching processors a logical development, freeing the composer to concentrate on the intuitive aspects of creativity.

If, as Sir Michael Tippett put it, "he (the composer) must activate the nervous system in order to produce the effect (to be transmitted in performance independent of himself) so continuously as to engineer periodic nervous collapse", then it seems perfectly fair to share some of the compositional work load with a computer!

Furthermore, for the person whose musical aspirations outstrip his performing ability, the use of computer-assisted music synthesis allows professional results to be produced, and even in some cases the training of the user to improve his basic musicianship.

This consumer's guide to Apple music will attempt to examine and dissect the five highly-viable music synthesis systems available now, or in the near future, for the Apple II in this country. First, a brief look at each system.

By DAVID ELLIS

Vista Music Machine 9

Composer and Edit programs allow the entry of up to nine parts of music using a combination of game paddles, Apple keyboard functions and a high resolution display of music graphics. The Play program enables the three on-board programmable sound generators to play note files with intermixing of white noise and square wave outputs, envelope shaping, amplitude specification and stereo mixing.

Soundchaser

The preliminary Music Operating Software comprises two principal programs. Edit allows the musician to preprogram the OSCs, VCFs and VCAs of analogue voice cards by using an interactive display to enter parameters in the form of "soft" switches, sliders and contours. The Sequencer assigns record or playback modes of four independent multi-event banks holding instrumental presets and polyphonic sequence data entered from the four-octave keyboard. Real-time playing can also be added on top of a sequence playback.

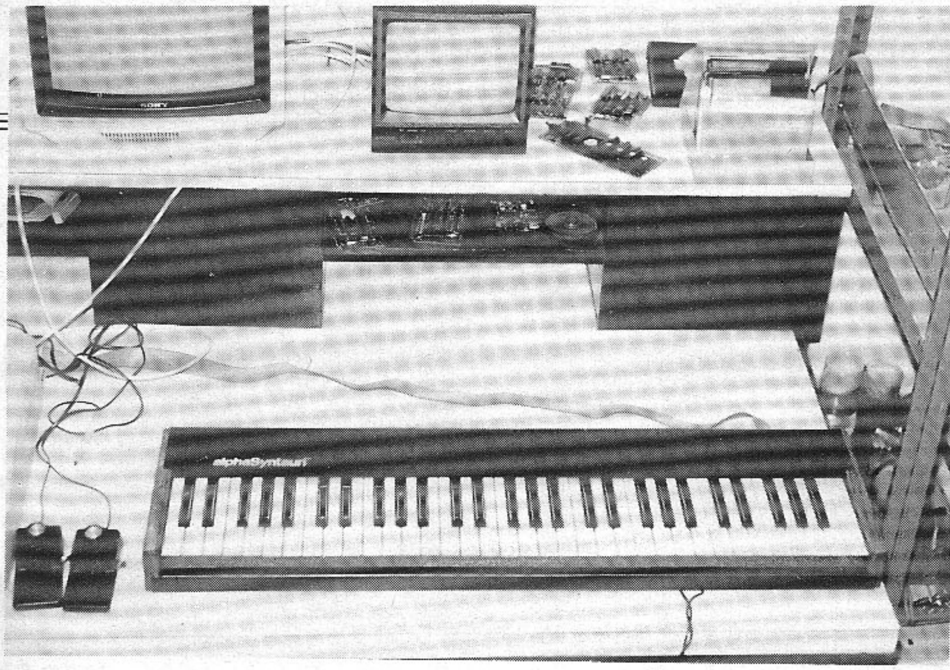
Software is written in Applesoft Basic, which is perhaps surprising for a program where speed of execution is usually of great importance. The future is likely to see much greater expansion of the Soundchaser's capabilities, with professional series software for teaching and musician series software offering more advanced musical utilities such as instant recall of eight presets and layered sequencing. Also a Music Sketcher offering real-time transcription onto the VDU of monophonic lines played on the keyboard is under development and is likely to retail for around $100.

MH MusicSystem

The Music Editor provides the means for music entry, with a theoretical capacity of 16 parts, either by using game paddles and light pen selection of options on four screen-based menus, or by the use of a totally keyboard-based music composition language. Music entered by either method can be printed out, part-by-part, on a Silentype printer.

The Music Merger enables small Comp files to be merged together to produce larger Comp files. The Music Player assigns instruments created by additive synthesis from sine wave components in the Instrument Definer to Comp files to create Play files. Waveform tables created by the Definer are then passed to the 16 digital oscillators, or D/A converters, on the MusicSystem boards, with further specification of each logical oscillator derived from amplitude envelopes and frequency/amplitude history details.

The software for the MusicSystem is interrupt-driven so that two programs can



The alphaSyntauri keyboard

run concurrently by means of a background/foreground mode of operation, and is written in a combination of 6502 assembly language and XPLO (a block-structured Pascal-like language also used in the Synclavier II). The use of direct memory access (DMA) enables all 16 digital oscillators to be updated every 32 microseconds, since a new oscillator update is performed every two microseconds, with table look-up carried out on 16 waveform tables simultaneously via DMA.

alphaSyntauri

The main alphaSyntauri program, Alpha III, is written in a combination of Integer Basic and 6502 assembly language, and in operation calculates what an equivalent analogue synthesizer would be doing every 20 milliseconds, and then controls the digital oscillators on the MH MusicSystem boards to achieve this.

Music input is derived from a five-octave, velocity-sensitive keyboard that can be tuned into the alphaSyntauri with traditional equal temperament, 24 tone scales like Just and International, and in the case of alphaPlus software, with microtonal tunings.

Either whole pieces or sequences can be stored in a polyphonic sequencer, edited to a limited extent, repeated, and subjected to real-time control of playback speed. Real-time playing can also be added on top of the playback, but only up to the limit of eight parts. Other real-time controls include instantaneous switching of presets and the selection of sustain and portamento (the latter two via foot pedals).

AlphaPlus software will additionally provide vibrato, timbral sequencing and frequency modulation. Visual feedback of real-time playing and sequence playback is derived from a multi-colour dynamic display.

Two further programs, Wave III and Analyzer III, enable the harmonic synthesis and analysis of waveform tables to be carried out. Apart from sine components, Wave III also provides the facility for using square, sawtooth and triangle waves in both additive and subtractive synthesis.

The main process loop also includes the option of jumping to a user-written subroutine, which allows any user familiar with 6502 assembly language to devise special effects and controls. Other software under development for the alphaSyntauri includes a training package for Computer Assisted Instruction in Basic Musicianship, a high resolution conventional musical notation graphics package for screen input of music, and a multi-layer sequencer package for multi-tracking.

Alf Apple Music II

The Entry program enables notes to be entered using the game paddles in conjunction with high resolution graphics, envelope specification of individual notes, and, to a limited extent, the programming of amplitude variation. The Play program can be engaged at any point during composition, and this also activates a multi-colour display of the pitches being played to provide an instant, alternative feedback to the compositional process. An additional Perform program allows the user to play note files created by programs other than Entry.

SOUND SYNTHESIS

A feature that stands out immediately upon booting a system disc and entering a pre-programmed note file is the "listenability" of different types of sound synthesis. The five runners fall neatly in to three categories of sound generation: analogue voice cards (Soundchaser), programmable sound generators (Vista Music Machine 9 and Alf AMII), and digital oscillators executing D/A conversion from waveform tables (MH MusicSystem and alphaSyntauri).

With the two systems using programmable sound generators – the Texas SN76489 (a more digitally-controllable version of the SN76477 games sound chip) in the case of the Alf AMII, and the General Instruments AY-3-8910 in the case of the Vista Music Machine 9 – one's off on a losing wicket to start with, as square waves have an annoying habit of infiltrating the senses with their buzziness – they're just unpleasant to listen to. Also a perennial problem with these sound generator chips is the low accuracy of notes higher up in the pitch spectrum. This is because notes are generated by dividing a master clock (approx 2 MHz in both cases, courtesy of the Apple Q3 system clock) by 32, giving a reference frequency of 63.9 kHz (Alf AMII), or by 16, giving a reference frequency of 125 kHz (Vista Music Machine 9), and this, in turn, is transformed into notes by means of programmable dividers using 10-bit (Alf AMII) or 12-bit (Vista Music Machine 9) numbers.

Obviously the larger the reference frequency and the number of bits being used for counting down the greater the accuracy of high frequencies. Thus at the top frequency provided for by Alf software (8th octave C, 4186 Hz), the resolution of the SN76489 is something like a third of a semitone out. With the AY-3-8910 the resolution is considerably better and 4186 Hz achieves the actual frequency of 4143 Hz, which is only about a sixth of a semitone away from that desired by the conventions of equal temperament.

Another limitation of the sound generators used in the Alf AMII and Vista Music Machine 9 is the very inflexible amplitude programming. In both cases a D/A converter is used as a programmable attenuator, but as only four bits of amplitude data are available this limits the available dynamic range to 30 dB, and, more importantly, what is available is provided in the less than ideal form of 16 2 dB steps.

Mind you, what can't be denied is that both these systems provide a lot of parts for very little in the way of financial input.

In addition to the MC1 board used in the Alf AMII there is also the older, three voice MC 16 used in the Alf AMI system. With this board, an on-board clock running at 1.78 MHz is used in conjunction with an Intel 8253, a chip containing three 16-bit timers for counting down (and therefore producing square wave notes) from the clock frequency. In this

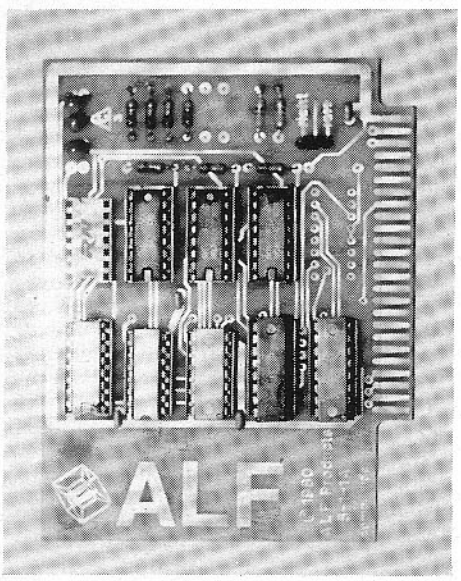
DIGITAL SYNTHESIS

case the resolution of high notes is greatly improved, so that 8th octave G sharp outputs at 6649 Hz rather than the equal tempered value of 6645 Hz; but what's 4 Hz between friends. The MC16 also uses 8-bit D/A converters for amplitude control, giving 72 dB of dynamic range.

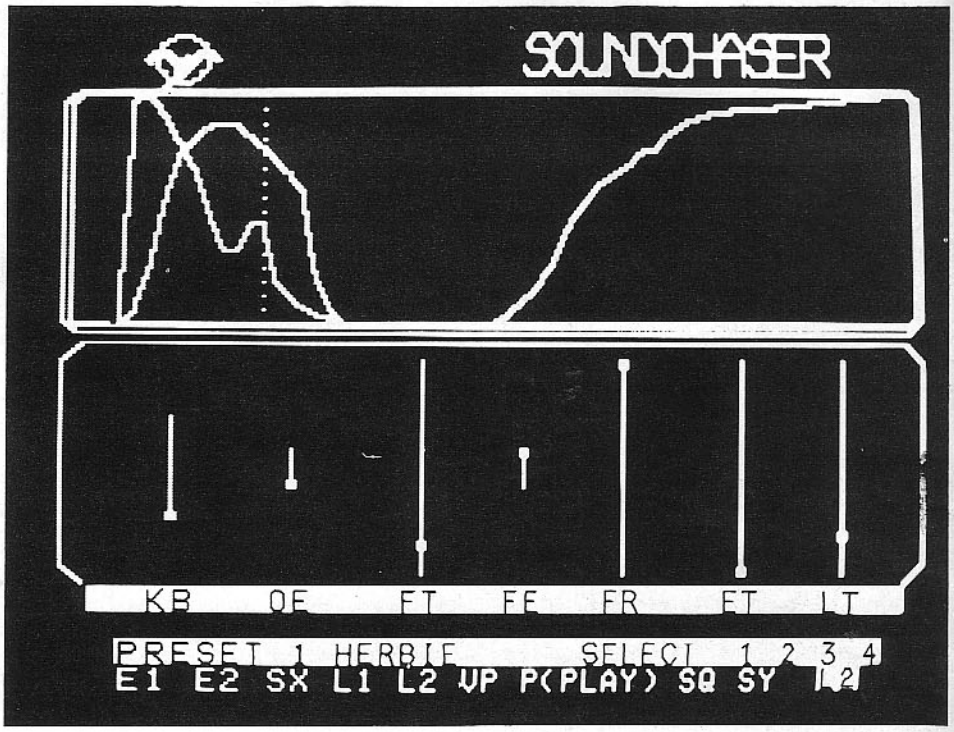
Because the individual 'oscillators' on the MC16 are more physically separated than in the case of systems using programmable sound generators, this does open up the possibility of cross-patching oscillators to produce pulse-width modulation (PWM) of the monotonous square wave output. This is, in fact, a common ploy of manufacturers producing cheap one oscillator synthesizers, where PWM introduces waveform animation through a type of phasing effect. Alf also provide this facility on the MC16 boards by gating the outputs of two oscillators with the third via the Chroma routine.

The MC16 then produces sounds of greater accuracy than from either the SN76489 based MC1 Alf AMII board or the AY-3-8910 based Vista Music Machine 9. However to obtain nine voices it would be necessary to invest in three MC16 boards, and the £300 spent on them might be more profitably directed towards the infinitely more open-ended synthesis offered by the MH MusicSystem for about the same price.

The Soundchaser voice card actually uses the same means of sound generation adopted by the Alf MC16. Here an Intel 8253 is run from the Apple clock rather than from an on-board crystal oscillator, but the square wave output is then subjected to a battery of processing. Firstly the square wave is treated by a wave-shaper to produce a sawtooth waveform. This or the original square wave passes



Alf AMII board



Soundchaser Edit display: curves show envelopes entered with paddle. Below them are the 'soft' switches and sliders that set up various control parameters.

through a Curtis Electro-Music 3320 VCF chip, configured as a low-pass filter with variable cut-off and resonance, and then an LM3080 transconductance op-amp, operating as a VCA.

The Soundchaser Edit display allows the user to program various parameters in this processor chain, using conventions of analogue synthesizers, but with distinctive features that are both good and bad.

One powerful feature is the ability to draw four contours on the display (using game controls) and assign these to control of oscillator modulation (LFO 1), VCA modulation (ENV 1), VCF modulation (LFO 2) and timbral sweeping (ENV 2), as well as envelope shaping (ENV 2). Since these contours can in theory be any size, shape, or form, remarkable possibilities of complex sounds are opened up.

However in practice the system fails to fulfill all one's synthetic expectations. Firstly, all voice card changes have to be entered in non-real time, so, unlike a conventional synthesizer, it isn't possible to change knob settings (ie, 'soft' switches, sliders and contours) as one is playing. Secondly, each voice is derived from only a single oscillator, with none of the facilities normally used in synthesizers to "thicken" this sort of voice, such as PWM or the mixing of the sub-octave and other waveforms to the basic oscillator output. While it is true that a sawtooth waveform is also available, it is only as an either/or option, and that is as a result of fiddling about with a DIL switch on the voice cards – hardly a real-time control.

To be fair, Passport tell me that they do intend implementing some form of real-time user control of the voice card parameters in the near future, and also plan to bring out a dual two oscillator/voice card using Curtis chips throughout for VCOs, ADSRs, VCFs and VCAs. This should provide a range of sounds equivalent to those obtained from a processor-controlled synthesizer like the Prophet 5, which also uses Curtis chips. Mind you, I think it's imprudent to launch a system like the Soundchaser with voice cards that are likely to be rapidly upgraded to something nearer the real McCoy – especially when each 3-voice card retails for $350 (probably £250 here). Perhaps Passport would consider a trade-in exchange on the old voice cards?

Moving away from digital control of largely analogue circuitry, we come to the two wholly digital systems, the MH MusicSystem and alphaSyntauri. Since both systems use the same hardware, it is hardly surprising that the sounds from them are similar, but there are important distinctions.

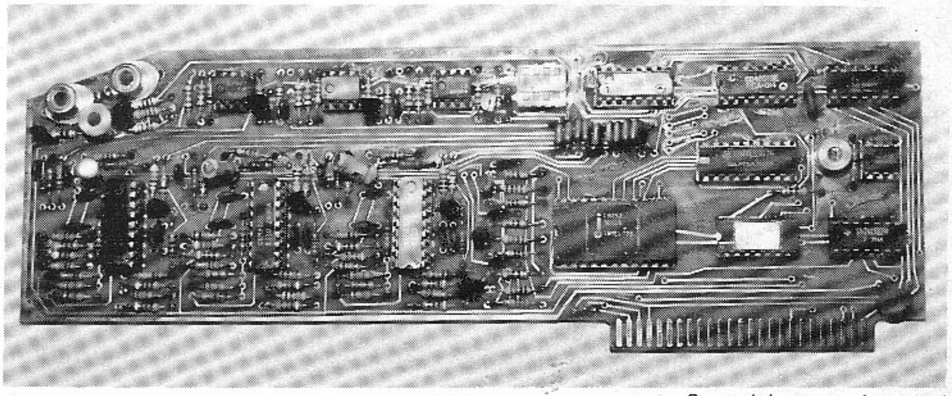
Though both systems have the same frequency response (30 to 13,000 Hz), the sound of the alphaSyntauri is noticeably cleaner than that of the MH MusicSystem. This is particularly noticeable at the bottom end, where the MusicSystem is inclined towards muddiness, and at the top, where the same lacks dynamic punch. Also the occasional digital gurglings and glitches apparent when the MusicSystem is outputting certain waveforms doesn't seem to be lurking behind the output of the alphaSyntauri.

However both systems show very good signal-to-noise ratios considering the limited resolution of 8-bit D/A conversion and display a dynamic range that must approach the theoretical limit of 48 dB.

Although the MusicSystem boards can run continuously under DMA once the appropriate registers have been initialized, playing from MusicSystem Comp and Play files, or from alphaSyntauri note files and real-time keyboard entries, requires changes in frequency and amplitude, while maintaining a specified tempo.

The MusicSystem accomplishes this by means of a constant rate of interrupt occuring every 8 milliseconds, which provides a time-base reference for a series of routines that update the registers in the MusicSystem boards. In the MusicSystem the interrupts synchronize the execution of a Supervisor program that, in turn, runs Producer (data from Play and Idef files turned into an output queue) and Consumer (output queue directed to hardware) programs. The accuracy of synchronisation is obviously 100 per cent dependent on the regularity of interrupts, and in the new alphaPlus software for the alphaSyntauri a clockWatcher program has been added to synchronize the instrument process loop and therefore to ensure maximum responsiveness of the MusicSystem boards.

Differences also abound in the way in which the two systems treat the subject of pitch and amplitude offsets. Taking envelope shaping to start with, the alphaSyntauri configures two oscillators per voice, with one oscillator having a fairly conventional ADSR envelope and the other a percussive PFSF envelope. Since the two oscillators are assigned to the two output channels, a note will pan from side to side according to the respective levels of points along the envelopes' duration.



Soundchaser voice card

This sterephonic animation of notes is quite stunning and, to my knowledge, only matched by the PPG Wave 2 digital synthesizer (cost ca. £3,500). Additionally the two oscillators per voice can be tuned apart with a variable offset to give rise to Leslie or chorusing effects. The Version 1.0 alphaSyntauri used a fixed offset (nominally 0.5 Hz), but alphaPlus software enables this offset to be derived from a 'soft' LFO, with rate, depth and waveform dynamically controllable. Unfortunately the MH MusicSystem lacks this sophistication of tuning offsets, and however many oscillators are used per voice, one is forced to use the same tuning over the sustain and release portions of the ADSR envelope cycle.

What the MusicSystem does provide though is the ability to program a 15-point series of frequency and amplitude offsets over the attack/decay portion of the ADSR cycle. With careful programming this enables some very interesting sounds to be created, but restricting offsets to the AD portion of the envelope is a limitation to the waveform animation possible with the present MusicSystem software.

There is little doubt that the MusicSystem hardware is capable of a sonic quality that bears comparison with much more expensive digital synthesis systems like the Fairlight CMI and Synclavier II. The lack of processor time to spend on sequencing through waveform tables, and thereby gain real-time timbral changes, is an important distinction between these 6502-based systems and the use of back-to-back 6800s in the Fairlight CMI.

However the alphaPlus software does include a form of timbral sequencing whereby the Consumer program receives a waveform table input that is switched through a sequence of look-up tables in a current preset bank. The effect of this is that each note struck on the keyboard triggers a pattern of sequenced sounds, with the option of variable sequencing speeds for this effect.

The recent introduction of The Mill 6809 card raises the interesting possibility of using this for the time-consuming task of music synthesis, for instance, by running the 6809 at full speed alongside the Apple's 6502 at 20 per cent efficiency to perform a rather more sophisticated type of timbral sequencing with the substantially increased processing power.

A little bird informs me that Passport Designs will also be bringing out software to run the MH MusicSystem from their own keyboard and interface, and at a projected price ($650) less than half the retail price of the alphaSyntauri in the States ($1425). This could really put the cat among the pigeons if their software lives up to what is promised.

Continued next month

Music synthesis session

NOW we have whetted your appetite on the various ways of creating music by computer, how would you like to try out the systems yourself?

Later this month Windfall is organising its first workshop on music and speech synthesis on the Apple, when you will be able to see the different systems in action and meet experts engaged in this fascinating new field of computing.

The date is Saturday, September 26. The place is the Royal Northern College of Music in Oxford Road, Manchester.

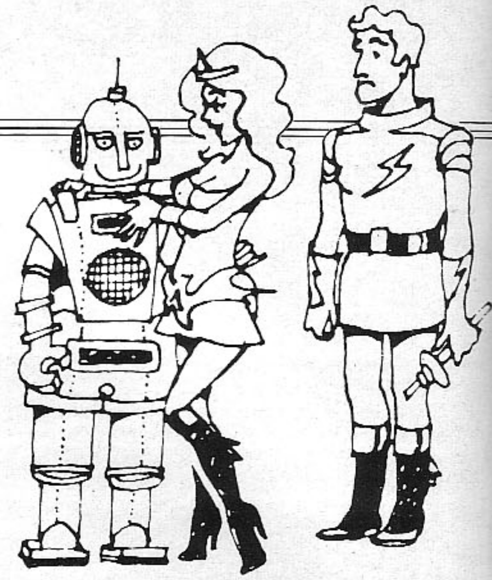
The day is being divided into two sessions, the first starting at 9.30am and the second at 2pm. Each session will begin with talks on music and speech synthesis given in the college's lecture theatre, followed by an opportunity to question the speakers.

You will also be given a chance to get your hands on some of the latest examples of music and speech synthesis equipment set up in an adjoining exhibition room.

This workshop, which is being organised by Windfall in association with Dr David Ellis and the UMIST-based North West Apple User Group, is the first of many events featuring the Apple which are now being planned.

Cost of the session is £2 per person. Tickets can be obtained from: Windfall, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY (tel: 061-456 8383).

The gentle art of gamesmanship



THE evaluation of computer games poses problems not apparent when dealing with hardware, for their attraction frequently varies according to the age, sex and predilictions of the player.

Consider then this and subsequent reviews as carried out by a gestalt offering the views of a moderately average, though slightly computer-oriented family of four. And one coming to the whole gamut of gamesmanship computer-style fresh, apart from the mandatory brush with the pub ridden whooping, flashing breed of space invaders and the like.

So Creature Venture, Highlands Computer Services' latest adventure offering, provided a stiff test for tiros. And still does after a few days of lurching from one diabolical doom to the next.

The ploy, in short, involves entering Uncle Stashbuck's mansion, ridding it of a horde of noxious creatures and laying hands on one's rightful inheritance, Uncle Stashbuck's treasure. And a darned sight easier said than done.

This is a fascinating, infuriating and challenging game. Not to be bought lightly if one values a good night's sleep. People have been known to rear up in bed in the early hours muttering something

By PETER GEE

about "I wonder if 2+2=22 because . . ." and then vanish below to the Apple for an hour or so.

There is seemingly always another stone unturned, another room unexplored, another cryptic clue to unravel. But that's just as well, for these games are not particularly cheap and it wouldn't do to finish it in one evening. The awful thought occurs that perhaps most people do just that. If so, spare our illusions.

Creature Venture has excellent graphics, stylish animation and colour thrown in, although the latter is hardly necessary. Who needs the odd bit of purple brickwork anyway?

One quite impressive feature of this game, incidentally, is its ability to sell Apple computers. Perhaps that is putting it too strongly. Let's say Creature Venture may well facilitate the introduction into the home of a small electronic stranger when all other arguments failed.

Good Lady Wife (smarm, smarm) previously considered computers as either overweening calculators fit only for the checkout desk at Tesco or whining, lunatic machines which trebled gas bills and then threatened the axe on non-payment.

GLW, let it be whispered, spent four hours before your actual Apple II trying to find her way out of the entrance hall of Stashbuck Manor on the very first night the devious disc was introduced into the house. Exit crosswords, enter computers.

Excuse me, please, I must ask her to move over. I want a game of football. . .

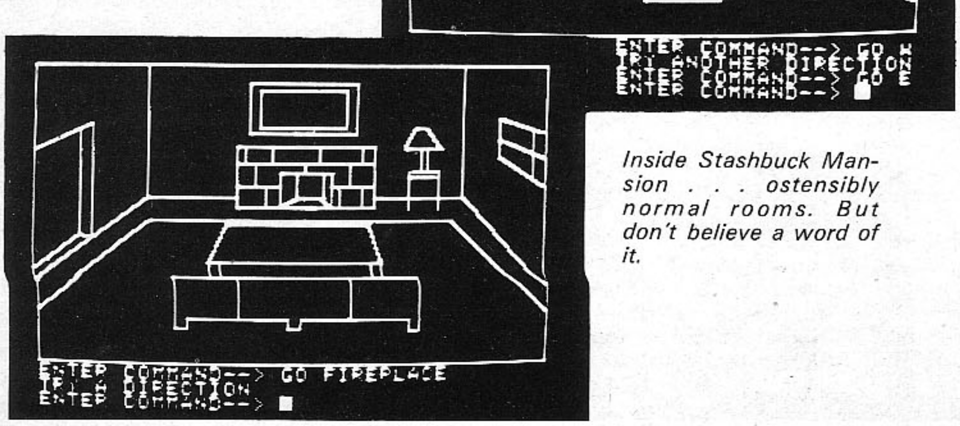
So to a disc of a very different type, where colour makes all the difference. High-Res Soccer, by On Line Systems, will not tax the imagination, but it certainly will reveal any sloppy reflexes.

Resident eggheads are unanimous that this is a technically very clever program. It is certainly an exceedingly good way to play football without getting beaten up by opposing fans.

Practically every movement on the real field of play can be duplicated on the small screen once one has mastered the quite complex series of movements offered by the paddle. Colour becomes important to easily identify the teams, and when this game was first tried on a mono monitor it got a unanimous thumbs down. For the hatching on the players is not all that easy to spot on a small screen.

But once a 14in colour set was acquired all was well. If an opponent is not to hand, one can play the Apple but you have to be a good loser to take this option.

Players can wrest the ball from the opposing striker, make a dazzling run



Inside Stashbuck Mansion . . . ostensibly normal rooms. But don't believe a word of it.

Starmines	**£17**
Creature Venture	**£17**
Hi-Res Soccer	**£24.95**

Games supplied by Sue Ben-David of SBD Software, 15 Jocelyn Road, Richmond, Surrey TW9 2TJ (tel: 01-948 0461).

down the field, dodging all and sundry en route and score brilliantly, the opposition panting at their electronic heels.

Just one thing. It seems rather hard to get the goalkeepers to spring into action. Perhaps mine was due for a transfer. Jay Sullivan and Ken Williams share the credits on the disc cover, but Sullivan alone makes a bow on the screen. Whatever, nice one fella or fellas.

To round off a lost weekend let's blast off with Starmines, an inter-galactic flight of fancy involving shoals of, presumably, meteorites, and swarms of lozenge shaped objects with evil intent.

If this seems vague the fault lies elsewhere, for instructions, or avowals of intent, came there none with the disc. It is a variant of space invaders with, happily, not quite the same ear-splitting sound effects.

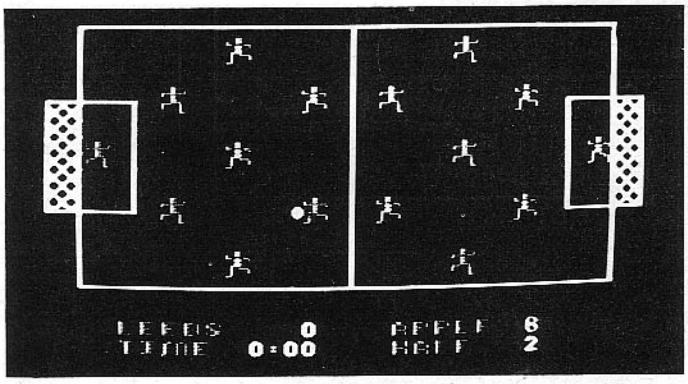
But, unhappily, with not the standard of control over one's environment one has grown to expect in the arcade machine. Pressing the paddle button releases a stream of death-dealing rays, and rotation of the other control moves the spaceship or whatever along the base line.

The object would appear to be to zap as many lozenges as possible while evading impact with the torrents of meteorites plus swooping invaders. But close examination revealed that the enemy tended to blow up if one shot within a parsec or two. And the game seemed to end without anyone in sight zapping the resolute defenders.

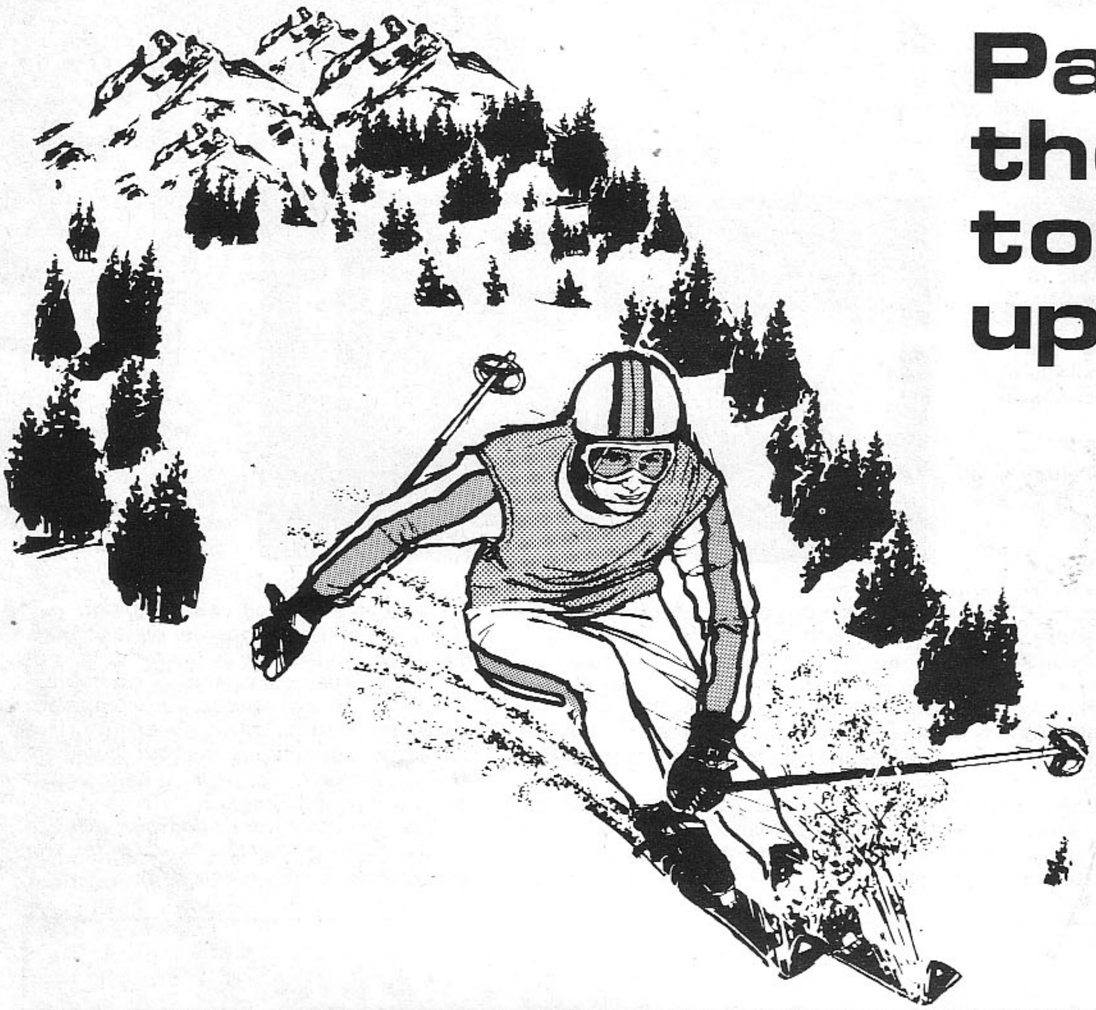
After a time in Stage 1 one progresses, with different colours, to Stage 2. And things move faster. Doubtless boastful friends say Stage 5 is up for grabs. But some people will tell you anything.

Having taken a moderate blast at Starmines one must look at the other side of the moon. The colour is good, no impressive. The sound effects are competent without being overpowering. The technical expertise is without question. The teenier teens will love it. I just wish I had got beyond Stage 3.

Next month Gamesmanship takes a look at the time travelling world of the war games men (and women) play.



Hi-Res Soccer – needs hi-res reflexes and a colour monitor.



Pass the pines to pick up points

"Ski Run" is an easy-to-programme slalom course where the skier's speed can be selected from 0 to 9. Not so easy for fast skiers is reaching the safety of an alpine hut without crashing into a tree.

```
1  GOSUB 500
10 S = 1024:V = 32
20  INPUT "SPEED (0-9)   ";K:U = 15:K = 450 - K * 50
25  CALL 768: FOR T = 1 TO U:P = 915 *  RND (1) + 2: POKE S + P,20: NEXT
30 R = S + 5: POKE R,33: POKE 1774,219: POKE 1775,221:R1 = S + 8 * 128
35  POKE S,96: FOR T = 1 TO 4000: NEXT : POKE S,32
40 P = R: IF  PEEK ( - 16286) > 127 THEN R = R + 1: GOTO 70
50  IF  PEEK ( - 16287) > 127 THEN R = R - 1: GOTO 70
60 R = R + 128: IF R >  = R1 THEN R = R - 856:R1 = R1 + 40
70  IF  PEEK (R) = 20 THEN  GOTO 110
80  IF  PEEK (R) = 219 OR  PEEK (R) = 221 THEN  GOTO 120
90  POKE P,V: IF  PEEK (R) = 160 THEN  VTAB 24: PRINT "TRY AGAIN FROM THE TOP";: FOR L = 1 TO 2500:
NEXT : HTAB 1: CALL  - 868: GOTO 30
100  POKE R,33: FOR Y = 1 TO K: NEXT : GOTO 40
110 . POKE P,V: POKE R,158: VTAB 22: PRINT "BAD LUCK"
115  PRINT "YOU CRASHED WITH ";U;" TREES ON THE PISTE": GOTO 150
120  POKE P,V: PRINT "WELL DONE "
130  PRINT "YOU MADE IT WITH ";U;" TREES"
135  PRINT "NOW TRY ";U + 2;" TREES":U = U + 2
140  FOR I = 1 TO 3000: NEXT : GOTO 25
150  PRINT "ANOTHER GAME? Y/N? ";: GET T$: IF  ASC (T$) <  > 78 THEN  GOTO 20
160  NORMAL : HOME : VTAB 10: HTAB 18: PRINT "BYE": END
499  REM *** INTRODUCTION ***
500  FOR J = 768 TO 783: READ L: POKE J,L: NEXT
510  DATA 169,32,160,4,162,240,32,237,253,202,208,250,136,208,245,96
520  CALL 768: VTAB 2: HTAB 16: PRINT "SKI RUN": INVERSE : VTAB 5: HTAB 2: PRINT "THE SKIER WHO LOOK
S LIKE '!' ": PRINT : HTAB 2: PRINT "MUST REACH THE HUT WITHOUT HITIING"
525  PRINT : HTAB 2: PRINT "THE TREES. YOU STEER HIM WITH THE ": PRINT : HTAB 2: PRINT "PADDLE BUTTO
NS. FIRST ENTER THE": PRINT : HTAB 2
530  RETURN
```

Foiling the microsecond munchers

By DOUGLAS MAY
MSc, PhD, FInstP.

THE Applesoft II Reference Manual advises (page 120) that variables should be used instead of constants in order to gain considerably in speed of execution.

It is not made clear whether this advice relates to the substitution of variables for all numerical constants or only for non-integers nor whether, when appropriate, integer variables should be used. For example, an efficient programmer wanting to execute

```
J = I * 2.5 + 3
```

within a loop, needs to know whether to write

```
A = 2.5 : B = 3
FOR I = .......
.
J 2= I*A + B
.
NEXT
```

or

```
A = 2.5 : B% = 3
FOR I = ........
.
J = I*A + B%
.
NEXT
```

or merely

```
A = 2.5
FOR I = .......
.
J = I*A + 3
.
NEXT
```

I have carried out some simple tests, on an ITT 2020, to compare the various possibilities. I used a program containing the assignments 100 A = 100 : P = 3.14159 : I = 3 I% = 3 : G$ = CHR$(7) and subroutines such as:

```
200 PRINT G$
210 FOR J = 1 TO N : B = A + 3.14159 : NEXT : PRINT G$
220 FOR J = 1 TO N : B = A + 3       : NEXT : PRINT G$
230 FOR J = 1 TO N : B = A + P       : NEXT : PRINT G$
240 FOR J = 1 TO N : B = A + I       : NEXT : PRINT G$
250 FOR J = 1 TO N : B + A + I%      : NEXT : PRINT G$
260 RETURN
```

Four other similar subroutines replaced '+' by '-', '*', '/' and '^', respectively. Input statements provided opportunities for keying in suitable values of N and for selecting the required subroutine. On execution, PRINT G$ caused six successive 'beeps' and the five intervals so defined were timed by means of a watch. The table below shows some of the results:

Times (in secs.) for N executions of loop involving 'A aop x'
A = 100 : P = 3.14159 : I = 3 : I% = 3

x	aop: +		−		*		/		^	
N:	2000	5000	2000	5000	2000	5000	2000	5000	1000	2000
3.14159	48	119	48	120	52	128	52	131	72	
3	8	20	8	21	9	25	12	31	51	
P	8	19	8	20	11	28	13	30	51	
I	8	20	8	20	10	23	12	30	52	
I%	9	22	9	22	10	25	12	31	52	

Note that the times quoted include the cost of testing and branching at the end of each cycle of the loop. From a separate test, this time was approximately 1 msec.

The main lesson from the table is clear: If non-integral constants are to be used frequently they should be replaced by variables, since their conversion to floating-point representation requires about 20 msec. for each execution. Among integer constants, real variables preset to an integral value and real variables preset to a non-integral value there is nothing to choose.

Integer variables need conversion to real representation and so impose a small penalty, of less than 1 msec. That is indicated by some, although not all, of the results quoted and is further borne out by the majority of other results, not quoted here. (The accuracy of these timings is insufficient to yield a better estimate of this small interval.)

Another point of interest in the table is the slight increase in the execution time of the operations of multiplication and division (relative to addition or subtraction) and the much larger increase for exponentiation. However, the differences between using a non-integral constant operand and any other operand remain essentially the same, whatever the operation, which reinforces the conclusions of the previous paragraph.

I am grateful to the Medical Faculty of Manchester University for providing facilities for these tests.

appletips

Applesoft stores its string variables dynamically. However when it runs out of free memory it must clean up the unused strings, a process which can take several minutes. There is no way to stop this process but you can give your program's user some warning.

The following line should be put in the main flow of your program.

```
1000 IF PEEK(112)-PEEK(110)=4 THEN
PRINT"STANDBY": A=FRE(0)
```

An alternative method is to execute a FRE(0) inside the major loop of your program. This would still in total take up the same time but at least this fractional delay would be less noticable.

THE most widely-accepted view of scientific method describes the initial step as the generation of hypothesis or model. The hypothesis is then used to structure an experiment that will test some or all of the qualities of the hypothesis. If the experiment fails to produce the predicted results then the hypothesis is modified to accommodate the new information that is a consequence of the latest experimental data.

Even if experimenters perform these operations subconsciously – and most do! – there still remains the need at the conclusion of the experiment to assess whether the experimental results conform to the current hypothesis.

There is a need to test the 'goodness of fit' of the data to the model, and to assess just how good the fit really is. This article describes the basic principles behind one approach to fit data to models and concludes with a description of a robust and general Apple II program. The description is restricted to experimental data that can be represented in two dimensions (x and y) and which can be plotted in a simple fashion.

Non-linear curve fitting using the Apple II

By Dr R J BEYNON
Department of Biochemistry
University of Liverpool

FOR the purposes of this article a function will be defined as the dependence of y upon x and is often abbreviated as $y = f(x)$. This is the general form of the equation, a typical function might be $y = 3x + 4$.

This expression illustrates that y varies as x changes and can be seen (or plotted) to be a straight line passing through the points:-

$x = 0, y = 4$
$x = 1, y = 7$
$x = 2, y = 10$

and so on.

A similar, but more complex function is given by $y = x^2 + x - 3$ or $y = \log(x)$. In the belief that his experimental system behaves according to a definable, if sometimes complicated, mathematical function the experimenter wishes to test his system to obtain a measure of the correspondence of the data to this function, working on the principle that poor correspondence means an inappropriate function.

Here, however, is a genuine problem that cannot be resolved without massive expenditure of resources. If we lived in an ideal world then the data would fit the correct function or model perfectly. However, even in an ivory tower the world is far from perfect and so our experimental data, describing the dependence of y upon x, are likely to contain errors.

The error may be due to difficulties in setting up the experiment, limitations in the precision of analysis of results, or simply due to the inherent variability that comes from sampling a small number of observations from an infinite world of experiments.

Errors can be minimised, often at considerable expenditure of resource. More often, however, the experimenter is stuck with error in his data and has to find methods of coping with this problem in his attempts to relate the real world (experiments and data) to his models (functions).

A microbiologist knows that during sterilisation of food, bacteria die in an exponentially decaying fashion, represented by the function

$N_t = N_o \exp(-k.t)$ or $y = N_o \exp(-k.x)$

Where

N_t is the number surviving at time t

N_o is the initial number at $t = 0$

k is a constant representing the rate of death

A plot of some representative data, obtained by counting bacteria at different times, is shown in Fig. 1. Although the scientist knows that the shape of the curve will always be the same he also knows that N_o and k, both constants, can vary according to the state of the food before processing, the sterilisation temperature, moisture content and so on. The constants N_o and k are the *parameters* of the function for this particular experiment and the values of these parameters are the unknowns that the experimenter wishes to find.

If the data were perfect then it would be a simple geometric matter to find the death curve that was represented by the data and hence obtain N_o and k. In fact the data are far from perfect, and so the problem becomes one of finding the best death curve that is represented by the less-than-perfect data. In other words an exponential curve must be found that is the *best fit* to the experimental data, as this curve will then allow the best estimate of the values of the parameters, N_o and k. This is the primary objective of curve fitting.

If we are looking for the line that is the closest description of the data we clearly must have some criterion of goodness of fit. Fig. 1 shows three lines (a, b and c). Subjectively, it is easy to choose line b as the best fit to the data – but why?

The most common response will be that b is the best fit because the data points are equally distributed on either side of the line. In other words the sum of the vertical distances between the data points and the line is as small as possible. The distance between the points and the line are termed *residuals.* In fact for statistical reasons it can be shown that under many conditions the line of best fit is the one that makes the sum of the square of the residuals (SSR) as small as possible (Fig. 2).

Any method that fits a theoretical line to a set of data by minimising SSR is using the method of "least squares" for curve fitting.

It is important to appreciate that linear curve fitting is not restricted to fitting a straight line to a set of data. The method of linear regression is perfectly acceptable for this process and a good program in Basic can be found in [1]. Non linear curve fitting has a special meaning that describes the dependence of y values upon the parameters of the equation, not just the x value.

It is beyond the scope of this article to discuss this at length, the interested reader is referred to [2] for further discussion.

The approach to non-linear curve fitting is not one that uses simple algebraic or geometric relationships and will often involve interactive procedures where, for example, the estimates of the parameters are refined in a series of trials, each trial being dictated by the success or failure of the previous one.

This approach has an additional

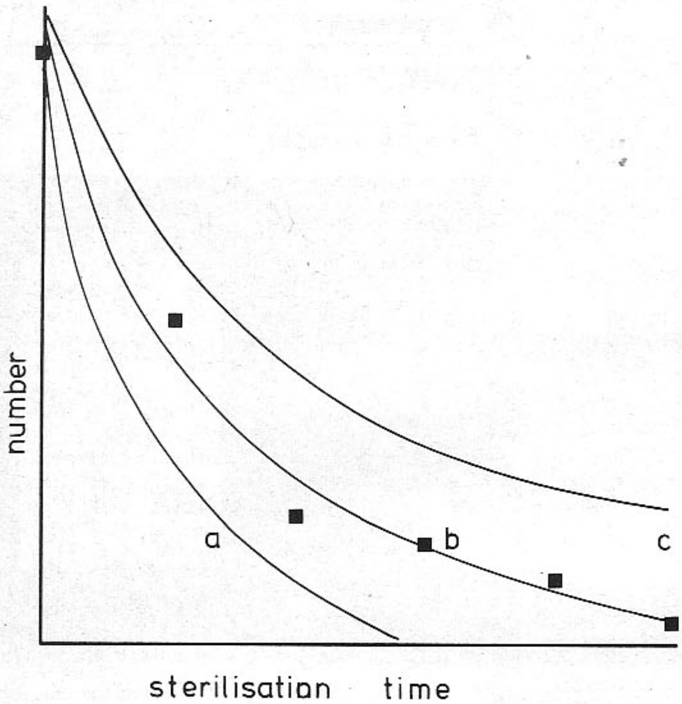


FIG. 1
An exponential decay curve representing the killing of bacteria in a sterilisation process. The experimental data are shown by the symbols and three possible lines describing the process are shown by a, b and c.

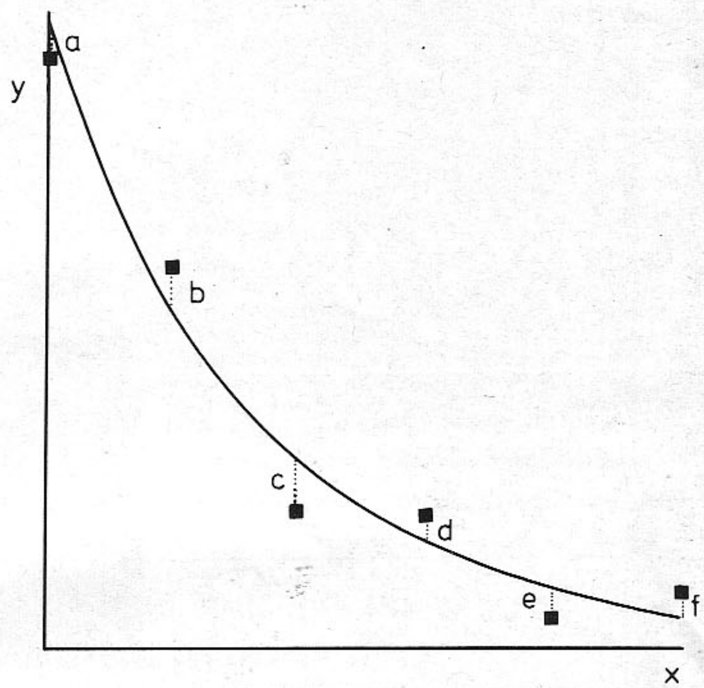


FIG. 2
Least squares optimisation aims to make the differences between the experimental points and the theoretical line as small as possible. In this case $a^2 + b^2 + c^2 + d^2 + e^2 + f^2 = SSR$ and is minimised.

advantage from the point of view of weighting of the data. Consider the bacterial death curve of Fig. 1. It is an easy step to plot the logarithm of number (N) against time to obtain a straight line. Application of linear regression such as in [1] can then give estimates of N_0 and k, but these estimates will not be statistically the best because an error at the smaller values of N will be magnified by the distribution of the axis produced by the transformation.

Minimising the SSR of a plot of log(N) against t will give a biased line that has been affected by the unequal weighting of the data points. In this instance, small values of N have more 'pulling power' in dictating the line than do large values for N.

It would be preferable then to minimise the SSR on untransformed data, i.e. N against t rather than log (N) against t. This will ensure that all data points exert an equal effect in deciding the final parameters of the best fit line. Because of this desire to perform curve fitting on untransformed, unbiased data we must use the iterative approach to non-linear curve optimisation.

Several programs for non-linear curve fitting have been written in a form that is suitable for implementation on microcomputers. Representative examples are given in [3-5]. The programs are specialised and some are restricted to the fitting of curves with only two parameters.

The rest of this article is concerned with an Apple program that I have written using a method described in [6] and based on an Algol 60 program published in [2]. The program, Pattern Search has been written with two main objectives.

Firstly, it was important that the data and function be presented graphically at the beginning or the end of the curve fitting procedure. Secondly, the program was written with generality in mind and can, by the alteration of seven lines of Basic, be applied to any new curve-fitting problem. The program has been used to date for the successful optimisation of five different functions – the record being the optimisation of seven parameters over 80 data points.

The Pattern Search method of curve fitting is intuitively pleasing if sometimes protracted. The program starts with initial guesses (user specified) of the parameters and calculates the SSR. It then systematically alters each parameter by a predefined amount (step size) and finds the pattern of altered parameters that makes the greatest reduction in SSR.

This process is repeated with the step size increasing as long as an improvement is measured as a decrease in SSR. When no further improvement is attained the step size is reduced and the SSR recalculated. This process of refinement is repeated until the parameters are being altered by less than a specified amount (e.g. 0.1 per cent of the actual value) or when the number of interactions that was initially specified has been exceeded. Further information on Pattern Search can be found in [2] and [6].

The translation of Pattern Search into Applesoft has produced a program of approximately 11 kbytes. The program is booted above HGR 1 before use so that display space is available. On running this program the user is presented with the menu shown on Fig. 3. A brief description of the options is now given:

● Enter parameters (1)

To find the best-fit values of the parameters of the function several numbers must be provided:

(a): the initial guess of each parameter – the better the guess, the shorter the run time

(b): the step size for each parameter – the value of the initial step used to modify the initial guess

(c): the reduction factor for each parameter – the amount by which the step size is reduced as the function is converging towards a good fit

(d): critical step for each parameter – if the program is adjusting the parameter by a value less than the critical step then a good fit has been attained and the program ends

An option is provided to use a standard search protocol that requires only the initial guess; (b), (c) and (d) are all automatically adjusted to preset values. Finally the maximum number of iterations is defined to prevent an over-intensive search.

● Enter exp data (2)

The experimental x, y data are entered via this subroutine. It is not necessary to specify the number of data points in advance as a null input is taken as the signal for "no more data".

● Edit exp data (3)

The facility to edit the experimental data is of greatest importance with large numbers of data points. The editing options are:

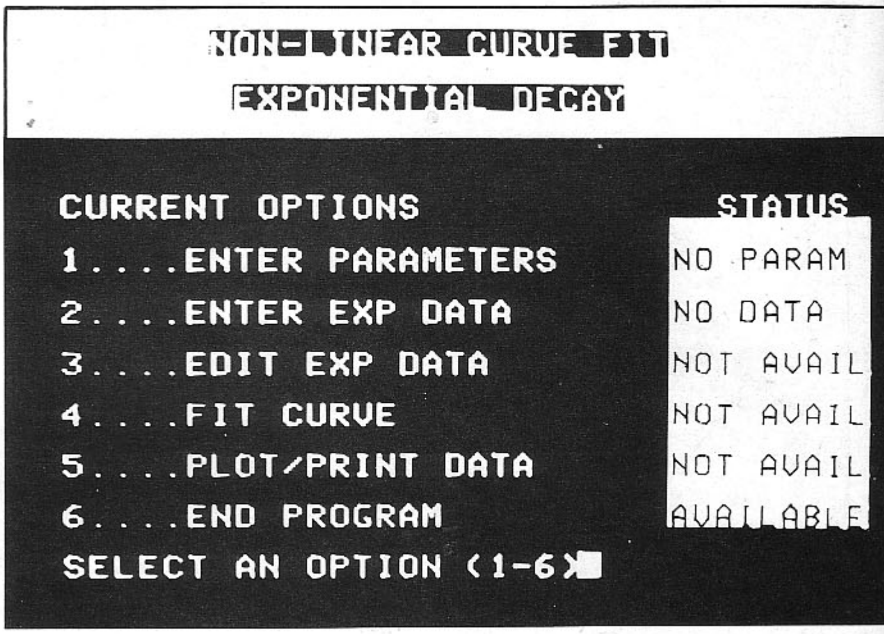
(a) delete a data point
(b) add a data point
(c) change a data point
(d) next data point
(e) finish editing

● Fit curve (4)

This option will only be accepted if a set of experimental data and a set of

→

FIG. 3
The main menu of the Pattern Search program. The status of each option is given on the right hand side and alters with the progress through the program. Each option is described more fully in the text.



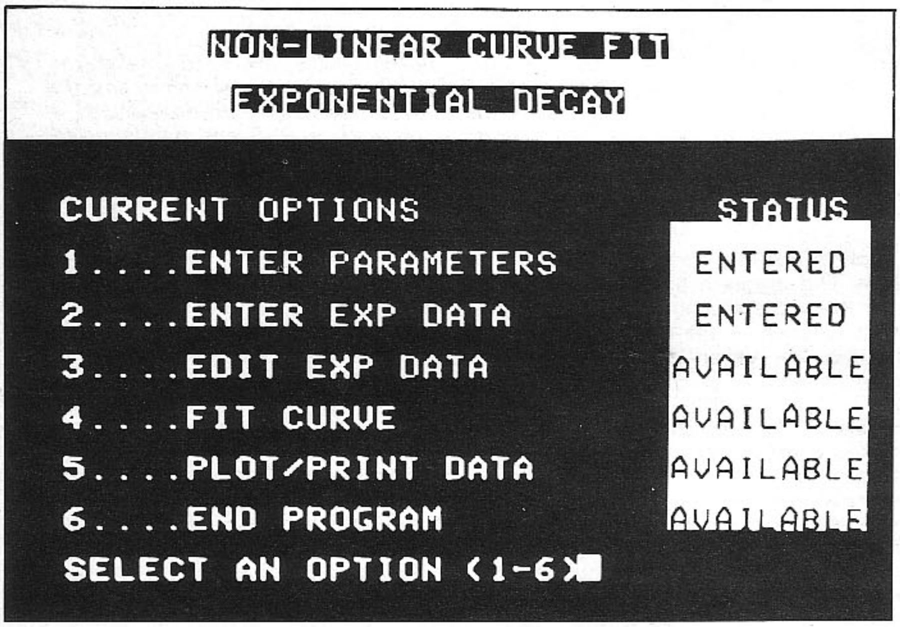
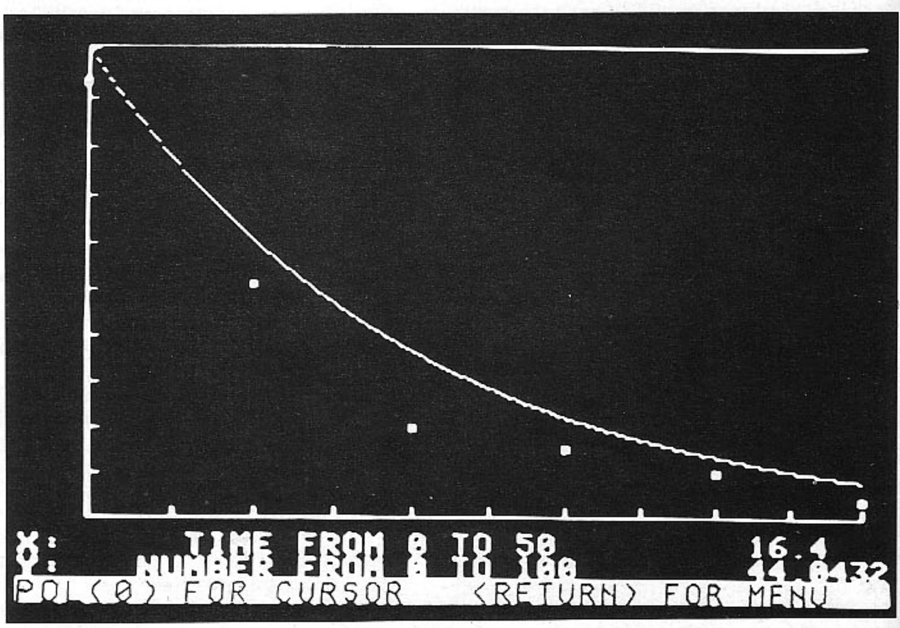


FIG. 4
The main menu after data and initial guesses of the parameters N_0 and k have been made.

FIG. 5
The trial plot of data and the function before curve fitting has taken place.



search parameters have been entered previously. If this criterion is satisfied then fitting is begun and a continuously updated screen display shows the number of evaluations, the current value of the parameters that are being fitted and the value of the function that is being minimised, usually the sum of the squares of the deviations. Although this slows the program it is instructive to observe the course of the search as it is taking place.

When the fit is completed, as judged by the critical step no longer being exceeded or the specified number of iterations being exceeded, the best-fit values of the parameters and the current step size are printed on the screen, together with the least squares final value.

● Print/plot data (5)

Several further options are provided by this subroutine:

(a) screen print of parameters at end (default)
(b) plot of data and function on the screen
(c) alteration of plotting parameters
(d) print data and parameters on printer
(e) return to main menu

Plotting of the fitted function superimposed on the experimental data is essential to view the quality of the best fit that the program has attained.

The option to enter the plot parameters such as axis limits has been kept separate from the option to plot so that the former need not be entered repeatedly. Similarly, the same data and function may be plotted over different plot limits as the program checks for out of range points.

To provide a permanent record of the result of the fit the opportunity is provided to list the data and the best-fit parameters on the line printer.

● End program (6)

An ordered exit from the program back to Basic command level.

The presentation of the data and functions (option 5) has one further refinement. The fitting process may be interrupted at any time by pressing the RETURN key, to allow a print out or plot of the current fitting parameters.

If this option has been selected then 5e becomes "continue fit" and the fitting will proceed without any disturbance to variables. The program will allow the user to plot the data/function at any time; before, during or after the fitting process. This means that the user may improve his initial guesses by trial plotting. In practice this does not make much difference to the number of iterations needed to produce a good fit but will reduce the possibility of the program locating and 'sticking' in a subminimum that is not the best solution.

A representative run, of the Pattern Search program is illustrated in Fig. 5. When the data and initial guesses have been entered the status of the program changes from the initial menu (Fig. 3) to the menu in Fig. 4. If option 5 is chosen and the data/function are plotted, the fit can be seen to be reasonable (No = 100, k = 0.05) Fig. 5. The function is now fitted, and after 79 evaluations (64 seconds) of the SSR, the parameters have been optimised to within 0.1 per cent.

If the data are replotted the line is seen to be considerably improved, the values of No and k now being 93.15 and 0.0647 respectively (Fig. 6). A similar run is printed out in Fig. 7.

The program described here is one example of a non-linear curve fitting program that can be run effectively on the Apple. I have found the program to be useful for five different curve-fitting applications to date, and we are continually trying new applications.

Copies of the Pattern Search program on a DOS 3.3 disc, complete with instructions on changing the function to be optimised can be obtained from me at the Bio-chemistry Department, Liverpool University, Liverpool, at a cost of £10 to cover handling and printing costs.

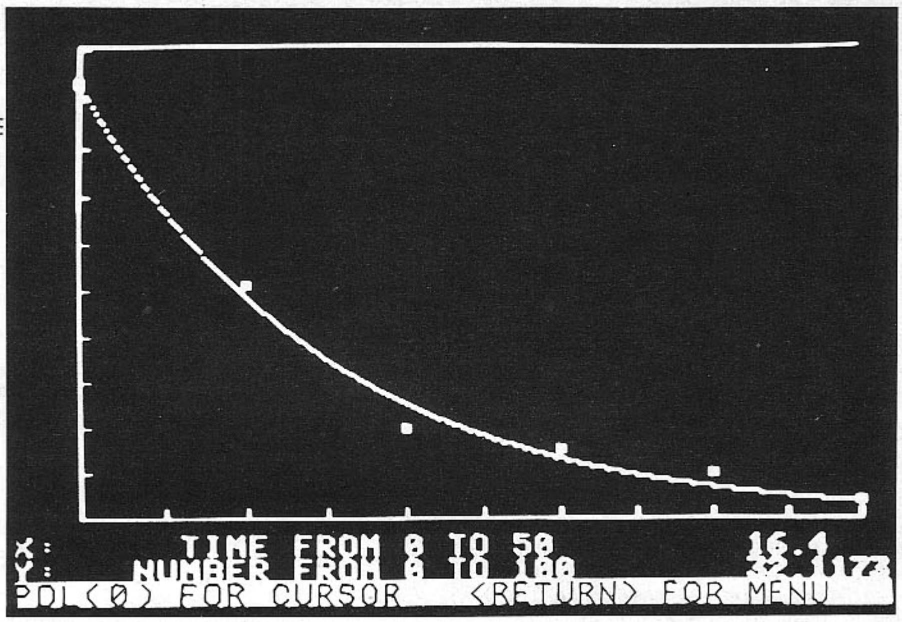


FIG. 6 *The re-plot of the data and the function after curve fitting has taken place.*

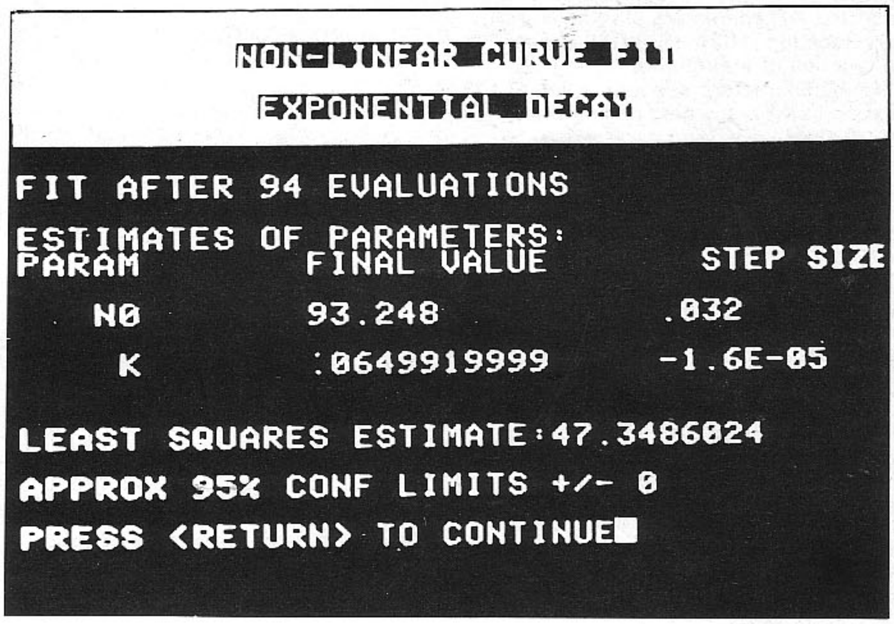


FIG. 7 *A screen print of the estimates after another fitting process has taken place. The figure for 95 per cent confidence limits has not yet been calculated.*

References

1. Lee, J.D., Beech, G. and Lee, T.D. (1978) "Computer Programs that Work" Sigma Technical Press, Wolverhampton.
2. Colqohoun, D. (1971) "Lectures in Biostatistics" Claredon Press, Oxford.
3. Peck, C.C. and Barrett, B.B. (1979) "Non-linear Least Squares Regression Programs for Microcomputers". Journal of Pharmacokinetics and Biopharmaceutics 7, 537-541.
4. Koeppe, P. and Hamann, C. (1980) "A Program for Non-linear Regression Analysis to be used on Desk-top Computers". Computer Programs in Biomedicine 12, 121-128.
5. Duggleby, R.G. (1981) "A Non-linear Regression Program for Small Computers". Analytical Biochemistry, 110, 9-18.
6. Wilde, D.J. (1964) "Optimum Seeking Methods", Prentice-Hall, Englewood Cliffs, New Jersey.

Putting the text in its place

By MIKE GLOVER
Leicester Computer Centre

THIS is the second in a series of articles designed to show the reasonably experienced Basic programmer how to write routines in machine code and get them into the Apple using the assembler. The assembler used for the examples is the one contained in the DOS toolkit.

Last month we looked at the way the Apple screen is memory mapped and saw that the first 40 characters starting from memory location 1024 ($0400) appear on the first line of the monitor screen, but the next 40 characters appear on line 9, the next on line 17, the next on line 2 and so on. A loop poking characters into memory from 1024 to 2039 shows how the memory is mapped on the screen. We also saw how the monitor routine at $FDED would print a character to the screen.

This month I want to show how to output blocks of text to the screen and position them where we want them. Also we can see how to input text and characters from the keyboard and finally take a look at the well published 'input anything' subroutine, which enables us to input strings containing commas and all manner of illegal characters without getting the extra ignored message from Basic.

To continue our analogy with Basic consider the following short program:

```
10 HOME
20 PRINT SPC( 14)
30 PRINT "TITLE"
40 VTAB 10
50 PRINT "LEFT";
60 INVERSE
70 HTAB 30
80 PRINT "RIGHT"
90 NORMAL
```

To emulate this in machine code we can utilise several of the routines already in the system monitor.

The HOME in line 10 can be reproduced by calling the monitor subroutine HOME which starts at $FC58.

```
BASIC                ASSEMBLY CODE
-----                -------------
10 HOME              JSR $FC58
```

Similarly, the monitor has a routine to print blanks which we can call having first loaded the X register with the number of blanks required.

```
BASIC                ASSEMBLY CODE
-----                -------------
20 PRINT SPC( 14)     LDX #$0E
                      JSR $F94A
```

The # sign means load the immediate (next memory location) value rather than the contents of memory location $0E.

Characters may be output by loading the Ascii value in hexadecimal in the accumulator and calling the routine starting at $FDED as described last month. So line 30 could be programmed like this:

```
BASIC                ASSEMBLY CODE
-----                -------------
30 PRINT "TITLE"       LDA #$D4
                       JSR $FDED
                       LDA #$C9
                       JSR $FDED
                       LDA #$D4
                       JSR $FDED
                       LDA #$CC
                       JSR $FDED
                       LDA #$C5
                       JSR $FDED
                       JSR $FD8E
```

There are more elegant ways of achieving the same end which we will look at later, but this way is the most straightforward. Note the JSR to $FD8E. This is the monitor routine for carriage return because a print statement not followed by ',' or ';' has an implicit carriage return.

Note also that the character set used for the Apple screen is taken from page 15 of the Apple II reference manual and is not the same as the Ascii codes you would send to a printer. We will cover the differences in detail in a later article when we look at a screen dump program.

Let's break off now and test the above coding. As it is rather short it might be a good idea to try 'hand assembling' it so that we can better appreciate what the assembler does for us. If we refer to pages 121-126 of the reference manual we can see that the op-code for JSR is $20.

The address FC58 would be put in memory as 58 FC so JSR $FC58 would become 20 58 FC. Line 20 would become A2 0E (LDX # 0E) 20 4A F9 (JSR $F94A) and the start of line 30 would become A9 D4 (LDA # $D4) 20 ED FD (JSR $FDED). Setting this out in the format used by the assembler we get:

```
20 58 FC    JSR  $FC58  ;HOME
A2 0E       LDX  #$0E   ;14
20 4A F9    JSR  $F94A  ;PRINT BLANK
A9 D4       LDA  #$D4   ;T
20 ED FD    JSR  $FDED  ;COUT
60          RTS
```

The RTS (RETURN) has been added so we can test the routine without dropping through the bottom and disappearing into the depths of the Apple's memory! Put this routine into memory by typing CALL -151 (from the Basic prompt):

```
*300:20 58 FC A2 0E 20 4A F9
     A9 D4 20 ED FD 60 'RETURN'
```

and test it with a *300G RETURN.

You should get a 'T' somewhat left of centre at the top of a clear screen.

Continuing our interpretation of the Basic program, line 40 is a VTAB to line 10. Due to the memory mapping we have already discussed, finding line 10 could be a little tricky so once again we can invoke a routine in the monitor that can do this job for us.

The routine that does the work is BASCALC at location $FBC1, but our most convenient entry point is TABV at $FB5B, having first loaded the accumulator with the desired line number. So line 40 becomes:

```
40 VTAB 10             LDA #$0A
                       JSR $FB5B
```

In line 50 we have some text but this time we will use a different technique to output it using the assembler. Assemblers contain several 'directives' which appear in the operation field and tell the assembler to take some course of action during the assembly. We have already looked at one of these, ORG, which when placed at the beginning of a listing tells the assembler which memory location should contain the start of our source code. For example, ORG $300 tells the assembler to start assembling code beginning at $300 (768 decimal).

We will use this directive a lot as, provided we don't interfere with DOS hooks starting at $03D0, this is a very convenient and safe place to put short machine code programs.

One of the directives that will help us with text is ASC. This data definition directive converts the text between two delimiters, usually 'TEXT' or "TEXT" into Ascii code. Take care with the delimiter used as some assemblers check them to decide whether or not to turn the most significant bit on or off.

A simple experiment, or as a last resort reading the manual, should resolve this!

One way to use ASC in an assembly listing is as follows:

```
BASIC                      ASSEMBLY CODE
-----                      -------------
50 PRINT "LEFT"            LDX    #$00
            LOOP1          LDA    TEXT1,X
                           JSR    $FDED
                           INX
                           CPX    #$04
                           BNE    LOOP1
                           BEQ    SKIP
            TEXT1          ASC    "LEFT"
            SKIP           .........
```

In this example we are using an addressing mode with the grand title of 'Absolute Index Addressing'. This simply means take the address you first thought of and add the contents of the X (or Y) register to it to form the actual address used.

Here we don't yet know what address we will be using and have labelled it 'TEXT1' followed by ASC "LEFT". This tells the assembler to place into memory the Ascii code for 'LEFT' and remember the address of the 'L'. Now if we say LDA TEXT1,X with X set to zero, we pick up the first character of our string. Adding 1 to X (INX) gives us the second and so on.

As (in this instance) we know the length of the string a simple comparison tells us when to quit – CPX # $04 compare X to see if it is 4. If it is not equal go back to LOOP1 and send out another character. The comparison that follows is not quite so obvious as we are now testing for equality which we already knew.

In other words if the test didn't produce an equal result we wouldn't be here in the first place so why not just jump to skip and re-write the bottom lines:

```
                BNE    LOOP1
                JMP    SKIP
        TEST1   ASC    "LEFT"
```

There are two reasons for this. Firstly, the second way takes one more byte – very important if you are short of space but for most applications not the main consideration. The second reason is that when a JMP instruction is encountered it causes a branch to a specific part of memory whereas a BNE, BEQ, etc., cause a branch to a location relative to the current location. This means that the program is not tied down to one memory location in order to work and is said to be 'relocatable'.

Inverse and normal are both monitor subroutines and the cursor horizontal position is controlled by the value in location $24 which is decimal 36 (you may well be using POKE 36,nn for tabbing on a printer).

So lines 60, 70 and 90 can be coded as follows:

```
BASIC           ASSEMBLY CODE
-----           -------------
60 INVERSE      JSR   $FE80
70 HTAB 30      LDA   #$1E
                STA   $24
90 NORMAL       JSR   $FE84
```

To simulate line 80 (80 PRINT "RIGHT") we will use a different assembler directive, DCI. This works just like the ASC directive except that all bytes have the most significant bit clear (bit 7), apart from the last one. This means we can test each byte to see if the most significant bit is set, and if it is, know we are at the end of the string.

```
BASIC                        ASSEMBLY CODE
-----                        -------------
80 PRINT "RIGHT"             LDY   #$00
            LOOP2            LDA   TEXT2,Y
                             BMI   LASTCHAR
                             ORA   #$80
                             JSR   COUT
                             INY
                             BNE   LOOP2
            TEXT2            DCI   "RIGHT"
            LASTCHAR         JSR   $FDED
                             JSR   CROUT
                             RTS
```

Having worked out the routines we need we can now use the assembler to put in the complete program. Run the assembler by typing RUN EDASM RETURN, change the ID stamp as required, RETURN and type AD RETURN.

```
                ORG    $300
*
CH              EQU    $24
PRBL2           EQU    $F94A
TABV            EQU    $FB5B
HOME            EQU    $FC58
CROUT           EQU    $FD8E
COUT            EQU    $FDED
SETINV          EQU    $FE80
SETNORM         EQU    $FE84
*
START           JSR    HOME
                LDX    #$0E
                JSR    PRBL2
                LDX    #$00
LOOP            LDA    TITLE,X
                CMP    #$8D
                BEQ    DONE
                JSR    COUT
                INX
                BNE    LOOP
TITLE           DFB    $D4,$C9,$D4,
                       $CC,$C5,$8D
DONE            JSR    COUT
                LDA    #$0A
                JSR    TABV
                LDX    #$00
LOOP1           LDA    TEXT1,X
                JSR    COUT
                INX
                CPX    #$04
                BNE    LOOP1
                BEQ    SKIP
TEXT1           ASC    "LEFT"
SKIP            LDA    #$1E
                STA    CH
                JSR    SETINV
                LDY    #$00
LOOP2           LDA    TEXT2,Y
                BMI    LASTCHAR
                ORA    #$80
                JSR    COUT
                INY
                BNE    LOOP2
TEXT2           DCI    "RIGHT"
LASTCHAR        JSR    COUT
                JSR    CROUT
                JSR    SETNORM
                RTS
```

Note that I have introduced another technique in printing out TITLE – where we look for a specific terminator, in this case $8D (a carriage return).

To conclude this month's article here is the 'input anything' subroutine. I first saw it in the Call A.P.P.L.E. magazine.

```
0300-   A2 00       LDX   #$00
0302-   20 75 FD    JSR   $FD75
0305-   A0 02       LDY   #$02
0307-   8A          TXA
0308-   91 69       STA   ($69),Y
030A-   C8          INY
030B-   A9 00       LDA   #$00
030D-   91 69       STA   ($69),Y
030F-   C8          INY
0310-   A9 02       LDA   #$02
0312-   91 69       STA   ($69),Y
0314-   4C 39 D5    JMP   $D539
```

It works like this. Basic keeps a table of variables starting from the end of the program and working up through memory. As programs change in length Basic needs to know where the variable table starts and it does this by keeping a record of the address in locations $69-6A. Try typing in this short program:

```
]FP
]10 AA$="XXXX"
]RUN
```

Now CALL-151 and type 800,817

RETURN, followed by 69.6A RETURN. You should have something like this:

```
*800.817
 0800- 00 10 08 0A 00 41 41 24
 0808- D0 22 58 58 58 58 22 00
 0810- 00 00 41 C1 04 0A 08 00

*69.6A
 0069- 12 08
```

This is the program in tokenised form and you can see the end of the program at \$0811 (Basic programs end with triple zero). Location \$69 and \$6A contain 12 08 which reversed gives address \$0812 and if we look at this address we can see \$41 which is Ascii for "A" and in \$0813,C1 which is Ascii for "A" with the most significant bit set. This way we know that this is a string pointer. (Refer to page 137 of the Applesoft reference manual.)

The next byte contains \$04 – the length of the string and the address in the next pair of bytes is \$080A and if you look at these locations you can see that the next four bytes:

```
$58 $58 $58 $58  =  "XXXX".
```

Note that as we have not altered this string in any way there was no need for Basic to move it from program memory. Try adding a line:

```
20 AA$=AA$+AA$
```

Run it and look at the pointers again.

What the 'input anything' subroutine does is to call \$FD75 (NXTCHAR), a monitor subroutine to input a line until a carriage return is found. On exit the string is in the keyboard buffer and the length in the X register.

Provided our variable is the first one declared we know exactly where it is as \$69 points to it so we can store the length in (\$69),2 and the location in (\$69),3 and (\$69),4. The addressing mode used here is called indirect addressing and works like this.

Find the address pointed to by \$69 (that, is, \$0812), add \$02 to it giving \$0814 and that's the address we want. Having pointed the first variable to the keyboard buffer a jump is made to \$D539, a routine to clear the most significant bit.

Now all that remains is to persuade Applesoft to 'adopt' this variable and move it out of the keyboard buffer before we overwrite it. One way to do this is to say AA\$=MID\$ (AA\$,1), which fools Applesoft into thinking that a change has been made so it dutifully puts the string elsewhere in memory and rewrites the pointers.

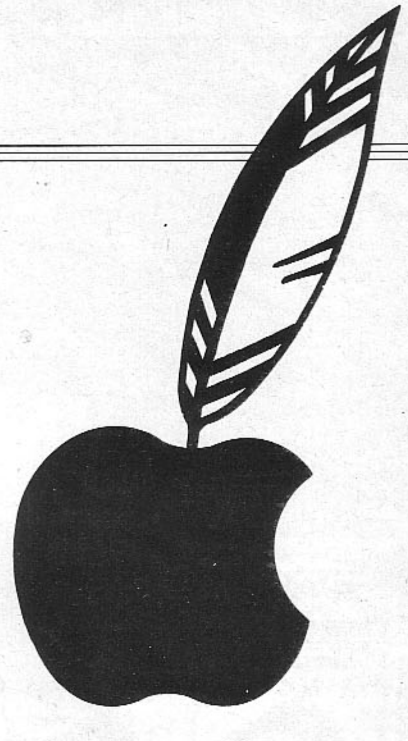
Use the routine from Basic as follows:

```
10 IA$="":REM MUST BE THE FIRST
   VARIABLE

20 PRINT"?";:CALL 768:IA$=MID$
   (IA$,1)

30 PRINT IA$:GOTO 20
```

Don't forget to have the machine code routine in memory at the addresses shown in the listing! Now RUN and type in anything you want, followed by RETURN.

Next month we will look at number base conversions and dumping the screen to a printer.

FEEDBACK



Pragmatism, not puritanism

WRITING irate letters to magazines is not a regular activity of mine, but having just received a couple of issues of Windfall I am unable to restrain myself.

The issue which has stirred me to such paroxysms of rage that I feel obliged to put fingers to keyboard is the matter of amateurs bringing amateur techniques to professional computing. They're quite welcome to do this in their home hobby computers, but it's somewhat distressing to find these horrors in a magazine which is aimed outside of the home hobbyist.

The article which particularly incensed me was the one by a Dr G J Boris Allan who railed for three pages against the restrictions on the imagination and creative flair of newcomers to programming, which the disciplines of structured programming impose – or at least they do according to Dr G J Boris Allan.

Dr G J Boris Allan claims that "too few programmers are being taught to exercise their ingenuity" and that "they enjoy programming in hex or using assemblers". Far be it from me to curtail either ingenuity or enjoyment, but I'd certainly like to curtail downright bad practice.

The promotion of structured programming is not based either on a spoilsport, puritan attitude nor on a desire to crush the creative flair of the fair flower of our youth. Rather it is based on experience and pragmatism.

The reason that computer scientists (and the data processing industry at large) have eschewed the practices that Dr G J Boris Allan (of the School of Sociology at the Manchester Polytechnic) would dearly have us instil into the innocent minds of children, is that they have consistently proved to be disastrous.

"Exercising ingenuity" means doing things in an idiosyncratic way, and doing things in an idiosyncratic way means that people who come along later to deal with the idiosyncratic creation are going to find it difficult to understand.

If they can't understand the creation they won't be able to correct any errors in it, they won't be able to change it to cope with new circumstances and they won't be able to use it as the basis for new developments.

Virtually every program that is used in the commercial world is the result of work by several individuals. These individuals need to work to common standards, and it's essential that restraints be put on their natural desire to exercise their creative flair in their part of the program.

Professional programmers write most of their programs in a high-level language because it has consistently been shown that programs written in high-level languages take less time to write, are more likely to do what they're actually meant to do, are more reliable, and can be modified more readily at a later point in their life.

Structured techniques are used because they enhance these same benefits, and also enable the programming task to be split up into sensible chunks.

Apples are being put into commercial and scientific applications where the requirements for the quality of the software are much more rigorous than are the requirements of the home hobbyist.

I suppose that, as someone working in the consultancy business, I should be rubbing my hands with glee at the thought of all the extra business that is going to come my way sorting out the systems that result from the archaic practices that Dr G J Boris Allan is promoting. Perhaps what's most alarming is that I can't really escape them, with people putting micros into just about every gadget that you can think of.

I can't help harking back to the thoughts of Edsger Dijkstra, who is as distinguished in the field of computer science as Dr G J Boris Allan (of the School of Sociology at the Manchester Polytechnic) is, no doubt, in sociology.

Dijkstra commented on the programming and design methods which the microcomputer users are now using, that if these same people were using micros to control lifts, he'd rather walk!

Yours apoplectically – **Jeff Hosier, Reading.**

Elusive service

HAVING recently become a micro owner it may be of interest to your readers to know how difficult it is to buy one. Like hi-fi the choice is bewildering to the novice.

Having narrowed my choice down to three machines, Apple, TRS 80 and Video Genie, to then get prices was easy. But prices with after sales service somewhat less easy.

Having decided to buy locally, which meant paying a higher price but with after sales service provided, I set out to buy a computer with a friend who was also interested in a possible purchase.

First stop is Tandy where we are informed the computer salesman is away but another person can help. We go to the showroom. Unfortunately the salesman is unable to get any computer working. "Must be the electrics," he thinks.

Second stop, Micro C. "Good morning, we want to buy two computers."

"Yes sir. What type of application?"

"For business use," I reply.

"Well sir, you will have to make an appointment to come back to see a demonstration next week."

Third stop is Leicester Computer Centre. It is 2pm and we are offered a cup of coffee. Let's see if we can do a deal. At 4.30pm we leave carrying one Apple and one Sharp MZ 80K, plus names of other contacts to assist.

Having since been back to Leicester Computer Centre for software and practical assistance they are very helpful and Mike Glover has the patience of Job with beginners like me.

By the way the Apple is performing faultlessly and doing everything expected of it. It is the operator that needs improving. – **J. Jennings, Kibworth, Leicester.**

Printout binders

I AM writing to say how much I enjoyed reading your first edition, and would like to wish you well for the future.

I have an Apple II Plus, used for business accounting, which I find first class. The manuals are very detailed and concise except in one area, that is indexing and cross-referencing of files used in the same program. Perhaps you could persuade someone to write an article on this topic, which I am sure would be of great assistance to everyone.

Another criticism that I have is that Apple stockists are not compelled to give users details of where to obtain supplies of paper, etc., and in this connection do any of your readers know a source of binders for printouts produced on small printers? – **J Davies, Thatcham, Berks.**

CRASH COU

YOU have just bought one of the wo versatile computers, either for serio your business or for fun in your home. is directed towards the pure beginne help to show how to instruct your Ap use of computer programming. Thi achieved by using simple techniques eventually, be built up into the mor procedures that make up the languag puters. Along the way you may make but, hopefully, you should learn from

YOU have unboxed your machine, connected all the wires and plugs, linked together the Apple, the TV screen and the cassette player (or disc drive) and switched everything on. You are now ready to take the plunge.

On your screen there should be a little flashing box, the cursor, which indicates the Apple is ready for action.

First type the word FRED and press the key marked RETURN. As you type, note that the letters printed on the keys you press are displayed on the screen.

The Apple has been designed to make this happen so that you can visually check what you have just typed.

The RETURN key is the key used to instruct the computer to look at what you have just typed and act on it accordingly. In the case of FRED our instruction seems to have met with disapproval. The words SYNTAX ERROR have appeared on the screen. The reason for this is quite simple.

Inside the computer is a section of circuitry which contains a library of instructions which are allowed when programming the Apple. These instructions form an intermediate language, which both codes human instructions into machine operations and codes machine information into a form easily understood by humans. This intermediate language is called Basic, which stands for Beginner's All-purpose Symbolic Instruction Code.

Alright. Now type PRINT "FRED". To type the quotes (") press the SHIFT key and, while keeping it pressed down, press the key with the number 2. Once the quotes have appeared on the screen you can release the SHIFT key.

When you have typed the whole line you can press the RETURN key, telling the Apple to look at what you have just typed and act on it.

Surprise, surprise. The name FRED has appeared on the screen – and without the quotes you should note – on the line immediately below where you typed PRINT "FRED".

What can we learn from this? Three things:

- To "talk" to the computer we must use an intermediary language – Basic.
- The command PRINT instructs the Apple to display on the TV screen the words you have typed. The screen is therefore being used as a visual prompting device allowing you, the operator, to see that the computer is actually working or accepting information from you.
- If we want to display a word or

world's most
erious use in
me. This guide
inner and will
Apple by the
This will be
es which will,
more complex
uage of com-
ake mistakes
om them.

Learn how to programme in just seven days..

By JEFF TURNER

words on the screen we have to tell the computer to PRINT the message. It is important to note that the word or words must be preceded by and followed by quotes (").

We know that the Apple can respond to one command because it printed FRED on the screen just as we told it to. That's fine if your name happens to be Fred. But what about a series of instructions?

Forget the computer for a moment and imagine that you have just been asked to give directions to the town hall. They could go something like this:

Straight down this road.
Turn right at the traffic lights.
Turn left at the roundabout.
Town Hall is 500 yards on the right.

This sequence of instructions is logical and well-ordered. A computer program follows the same principles, with one extra proviso. The instructions must be numbered so that the computer will know which instruction to act on next.

If I said:

(3) Turn left at the roundabout.
(1) Straight down this road.
(4) Town Hall is 400 yards on the right.
(2) Turn right at the traffic lights.

You would sort the instructions in ascending order, 1 to 4, and act on the instructions. Let's try that with our computer.

Type in the following as it is shown, and as an extra prompt I have put RETURN in brackets at the end of each line to remind you to press the RETURN key.

10 HOME (RETURN)
20 PRINT "I CAN PROGRAMME" (RETURN)
30 GOTO 20 (RETURN)

Now that the program has been typed in you should have noticed that the Apple has not done anything yet. The reason for this is that by typing the line numbers 10, 20 and 30 the machine recognises each line as a line of a program.

Because it does not know how long the program is going to be it will only accept each line when you press the RETURN key, and store that line ready to be acted on at a later time.

The first thing we are going to do is to check that the program has been stored and that the typing was correct.

Type LIST and press the RETURN key. The program should be displayed on the screen and should look like this:

10 HOME
20 PRINT "I CAN PROGRAMME"
30 GOTO 20

If your program doesn't look right, there may have been an extra key pressed by mistake. Just type the whole line again, remembering to put the same line number at the beginning of the line.

OK, we now have a program in our computer. What do we do with it?

The computer is waiting for our command so we will tell it to RUN the program. Type RUN and press the RETURN key.

There you are – it's up in lights. You *CAN* programme!

Let me explain what is happening. If we take each line number in turn I will describe what is going on.

10 HOME

This line instructs the computer to erase the screen of all information currently being displayed (remember FRED?) The cursor position is now at the top right hand side of the screen.

20 PRINT "I CAN PROGRAMME"

This line instructs the computer to PRINT (display) the contents of the quotes. The position at which the message, I CAN PROGRAMME, will be displayed is the position of the cursor.

Our message is therefore printed at the top left of the screen. Once this has been done the cursor moves down one line, still keeping to the left of the screen.

Let us note at this stage that the message in the quotes could be anything. I have simply used the message, I CAN PROGRAMME, because in the early stages of learning to programme it is nice to have the computer giving you a visual pat on the back.

30 GOTO 20

This line instructs the computer to back-track, or to use program jargon, to LOOP. Because the computer acts on the instructions in line number order it will have acted on line 10 first, then line 20 and on to line 30.

At this point we make it act on line 20 again. The effect of this is that the message I CAN PROGRAMME is printed over and over again.

To stop this process from carrying on too long press the key marked RESET. This key instructs the Apple to break into the running of a program and return to the command mode.

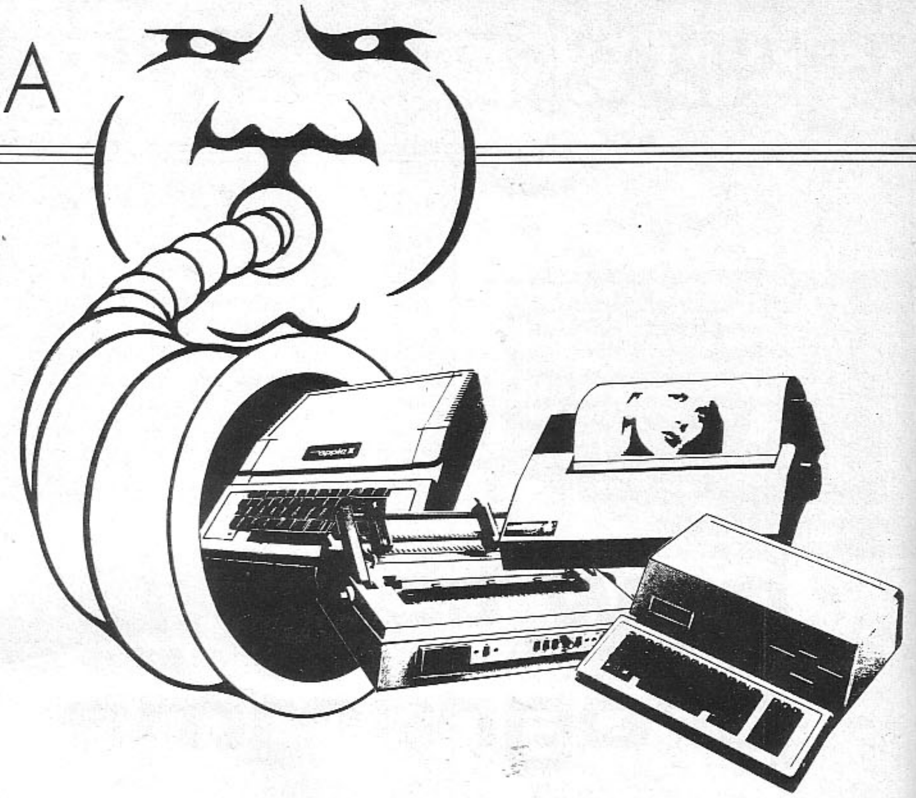
You can now LIST the program, RUN the program, change it, re-write it or do whatever you want with it.

Let us recap on what we have achieved so far. We have used the RETURN, SHIFT and RESET keys, as well as several of the letter and number keys. We have used the commands PRINT, HOME, RUN, GOTO, LIST. And most important of all – *you have written and run your first program.*

Not bad going for your first day. Try changing some of the program, just to see what happens. And remember, we all learn by our mistakes.

Until tomorrow then. Good luck – and good programming!

COMPUCOPIA



HOT on the heels of the Z80 chip for the Apple now comes the 6809 processor. Designed for a different type of operation than the Z80, the 6809 contains a 16 bit instruction set and is able to execute completely position independent code.

To take advantage of this Floppysoft have produced a package containing the board holding the processor, called The Mill, a 6809 assembler (ASSEM) and Gemini, which sets up a working environment for both the 6502 and 6809 processors.

The whole system is known as the Apple II 6502/6809 development system. It enables the user to enter, edit, list, assemble, save and run 6809 programs by single commands.

The advantages of using the 6809 processor are significant. It contains the most complete set of addressing modes, with the ability to index from a base address into a data structure or stack with a single instruction. It has direct addressing anywhere in the memory map, long relative branches, true indirect addressing, expanded indirect addressing and many other features.

It has been called the programmers' processor. As well as having most of its data manipulation in the form of 16 bit integers and addresses, the 6809 can be run in parallel with the 6502 processor. This allows the 6809 to run at full speed, while the 6502 runs at 20 per cent of its maximum. It also means that monitor routines such as I/O can be processed simultaneously with 6809 code.

Speed of the Apple, particularly when Pascal is being used, can be increased, with a 50 per cent decrease in the time taken to interpret P-code.

The complete package is supplied at a price of £249. The Pascal speed-up kit is a further £79, and the Floppysoft assembler can be purchased separately, either as a 4k Rom or on disc for £59.

Contact Floppysoft, 13 The Gables, Haddenham, Bucks (tel: Haddenham 291059).

Interfaces galore

A RANGE of interface cards for the Apple has been developed by Aughton Microsystems of Kirkby, Liverpool. It includes an eight-channel relay output card, a dual PIA card, a 16 channel, 12 bit A/D converter and an eight channel D/A converter.

The eight channel relay output card has eight individually addressable relays, with normally open contacts. Contact ratings are: I max - 0.5 amps; V max - 200vdc; W max - 10 watts.

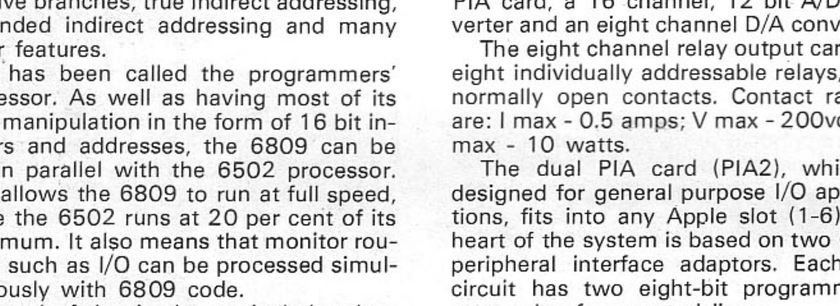
The dual PIA card (PIA2), which is designed for general purpose I/O applications, fits into any Apple slot (1-6). The heart of the system is based on two 6520 peripheral interface adaptors. Each PIA circuit has two eight-bit programmable ports, plus four control lines.

The card provides 32 non-isolated digital lines programmable as inputs or outputs. In addition to the 32 lines there are eight control lines available, four of which may be programmed as inputs or outputs and four are interrupt inputs only. A typical application for this card is to link a GC Oertling Balance to the Apple.

The 16 channel 12-bit A/D card (ADC1612) contains an analog multiplexer with switch selection of either 16 single ended or eight double ended differential inputs. Conversion time is 25 microsecs, and it has the normal 10 volt range.

The eight channel card, which also fits into any slot on the Apple, comprises a series of eight DACs, complete with latch and output buffer. Voltage range is 10 volts, either unipolar or bipolar, and resolution is, of course, eight bits.

Contact Aughton Microsystems, 29 Woodward Road, Kirkby Industrial Estate, Kirkby, Liverpool L33 7UZ (tel: 051-548 7788).



Tune in to the atomic clock

Clock in spot on

YOU want accuracy! We got accuracy! By installing a receiver on a card and putting it into the Apple you can pick up the transmitted signal from the Atomic Clock at Rugby.

This gives you the time, date and day of the month to an accuracy as good as you will get anywhere. Universal time in fact.

Thorpe Management Technology can supply the card, software listings and a disc to help set up the system and to link the system into personal programs.

The system, which will work within 1000km of Rugby or one of three other transmitters, cuts out the need for battery backup, as the correct time signal is constantly being received whenever the Apple is switched on. Software in the EPROM converts the raw frequency data into readable dates and times.

A further advantage to the clock card is that the space on the EPROM can be used for additional information, such as serial numbers, allowing software writers the ability to use the EPROM as a means of software protection. The software can be constrained to run only on Apples containing a clock card with the relevant serial number. The card costs £120.

Contact Thorpe Management Technology, of 171 High Street, Barnet, Herts (tel: 01-499 1334).

Busy is beautiful

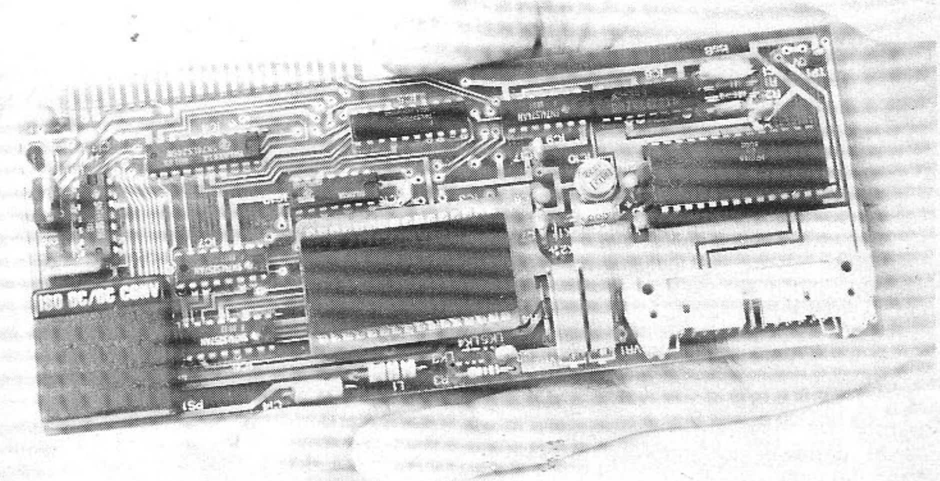
EVERYONE knows what an unholy mess the average Apple computer workstation can get into after a hard day's work, with discs and printout lying around, together with the paddles from a surreptitious game of invaders.

Emkay Systems have produced a workstation to overcome this problem, and at the same time help to increase efficiency.

The workstation is literally a control centre, providing power for all the units in the system from a back-lit, front-mounted console. As well as stopping you fumbling around the back for on/off switches, the unit is designed to leave the top of the Apple free, giving immediate access to its insides for changing boards.

As it is built to be used in office environments, the monitor supports and shelf units for discs are constructed in a pleasing wood-grained material. All wiring is fully fused and built to heavy industrial standards, with an in-line filter to eliminate spikes, and the whole vulnerable mass of wiring, so familiar to Apple users, is packed away inside a full width back panel.

Contact Emkay Systems, The Business House, 37 St Michael's Square, Gloucester GL1 1HX (tel: 0452 424411).



Sixteen channel 12 bit ADC from MC Computers

With industry in mind

A NEW range of I/O cards specifically designed for the Apple computer has been unveiled by MC Computers of Newbury. Intended to be used in an industrial environment, the range includes a high speed multi-channel ADC, a 16 channel digital input card, and a 16 channel digital output card.

The 16 channel ADC, offering 12 bit resolution, claims to have none of the compromises usually associated with such micro devices. The conversion rate is 25 microseconds, with a non-linearity better than +or— 0.025 per cent. The device, which is also protected to a maximum overvoltage of +or— 30 volts, costs around £600.

The input and output cards are both isolated devices based on already proven designs. The input card accepts logic '1' signals of 3 to 10mA, while the output card, rated at 24 volts and 100mA, provides contact closure outputs.

All three cards have board mounted ribbon cable terminals and will plug into any slot of the standard Apple.

Contact MC Computers, Park Street, Newbury, Berkshire (tel: 0635 44967).

SOFTWARE SCENE

Visicalc's business range expands

THE ever-popular Visicalc has now developed into a full range of related products, all designed to help run a business more effectively.

Visicalc: This needs no introduction, being the most widely sold program in use on personal computers in the world. It is a real time problem solver, used to produce budgets, projections, financial statements and many other functions. Now available in an improved package, incorporating many new features and in 16 sector format.

Visiplot: Create graphic displays of Visicalc models for better presentation and understanding of data. Visiplot gives six different display capabilities and has an easy interface to Visicalc.

Visitrend/Visiplot: Professional and business users can perform rapid regressions, time series analysis and data transformation with this package, which also incorporates charting and graph capabilities. Can be used for more sophisticated trend forecasting and statistical analysis.

Visidex: An unrestricted personal filing system with unlimited cross-referencing. Visidex is useful for storing and retrieving information such as names and addresses, important numbers, dates, ideas, lists, report and memo highlights and other facts requiring constant and quick access. With an optional clock card, one can turn it into a personal diary.

Visiterm: This allows communication with other computers, either personal or mainframe. It gives access to timesharing services, for databases, stock reports, newswires and other services.

Desktop Plan II: Another development of a popular package. Desktop Plan II now comes with charting and graphing

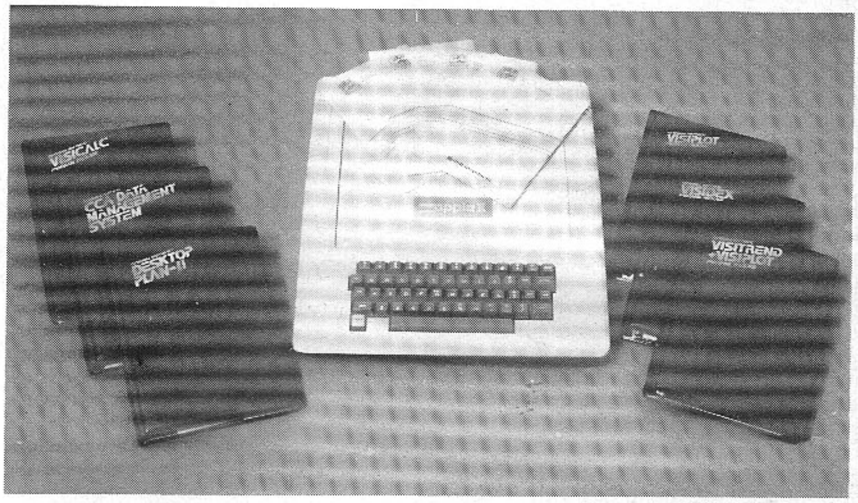
COMPUCOPIA

capabilities to improve presentation. It also includes more complex and frequently used financial calculations – including multi-step operations using chains of test and formulas – providing an efficient and easy-to-use package for organising finances, budgets and business planning.

The quality of reports available with Desktop Plan II are suitable for presentation at boardroom level.

The personal software range of packages, which can be used together or as independent programs, are available from Apple dealers throughout the country. The quality of the software has led to them being endorsed by Apple Computer Inc.

Prices, all excluding VAT and carriage: Visiplot £100, Visitrend £144, Visidex £111, Visiterm £84, Desktop Plan II £111 and Visicalc £111.



Expanded Visicalc family

Insurance package

IT has been difficult in the past to find a substantial Accounting package for insurance brokers on the Apple, but now CPR Systems of Suffolk have come up with a totally integrated system to handle up to 5,500 accounts.

The system, using a hard disc for fast and efficient data storage and access, is designed to meet all the needs of a normal general insurance broker.

GIBS (General Insurance Brokers System) handles policy recording, debit note production, client accounting and insurance and company accounting.

A full range of reports is produced with each section, which apart from providing client directories and dated renewal lists provides insurance-related business statistics. A general ledger can also be added to GIBS to provide profit and loss and balance sheet reports.

The system is provided on a leased basis over five years at £210 per month, including VAT, and in addition to this CPR Systems can provide maintenance, training, stationery and appropriate software amendments to fit a system to a particular customer.

Contact CPR Systems Ltd., 37-39 Ipswich Street, Stowmarket, Suffolk (tel: 04492 5488).

. . and one for commerce

USING the Megastor eight inch disc unit, which is now being widely used on Apples in Europe, Vlasak Computer Systems are offering a suite of commercial packages specifically designed for larger companies.

Termed Ledger 11+8 and Invoice. 11+8, the software will handle standard sales ledger accounting for up to 1,500 accounts and 600 transactions, capable of integration with a bespoke invoicing routine.

The sales ledger is both balanced forward and open item, and can be linked to Stock 500, Vlasak's stock control package, via the Invoicer.

As a useful addition to these packages Vlasak are now manufacturing their own computer stationery, and can supply preprinted stationery for all their products.

Cost of the Ledger System is £315 and Invoicer 11+8 is £260. Purchase and nominal ledgers are also available as is a Payroll at £395.

Contact Vlasak Computer Systems, Vlasak House, Stuart Road, High Wycombe, Bucks (tel: 0494 448633).

Command Editor

AN ACE set of utilities to enhance your Apple is provided by Applesoft Command Editor, which is co-resident with DOS 3.2 or 3.3. ACE is designed to simplify and speed up program development, providing such functions as:

- Line editing
- Auto line numbering
- Display values of non-array variables
- Macros to define keywords, variables and other strings as single control characters.
- Execution of monitor commands from Applesoft.
- Easy Dec/Hex conversion.
- Search for variables.

At a price of £19.95 this package has just got to be a must for anyone seriously involved in program development.

Available from Leicester Computer Centre, 67 Regent Road, Leicester (tel: 0533-556 268).

Useful check

ANOTHER useful product from Leicester Computer Centre is the Visicalc Utilities Program, which assists you in checking formulae without being confined to the Apple screen, and allows you to re-format the printout from Visicalc with variable column widths, additional text headings, dates, page control and numberings. Price: £34.95.

Bit 3

ANOTHER 80 column card is Bit 3, which offers switchable software, avoiding tedious recabling and switch selection, and an optional English character set. The card has true descenders and comes with three different configurations. Price is £240.

Contact Computopia Ltd, 30 Lake Street, Leighton Buzzard., Bedfordshire (tel: 0525 376600).

Data Factory

ONE of the best selling databases on the Apple, The Data Factory, is now available from Personal Computers Ltd.

Costing only £100, this universal database management system has already been acclaimed in the States because of its flexibility and extensive search facilities.

The package consists of nine modules, with only one being loaded for use at any one time. The extended search permits up to four retrieval variations, and will function on record numbers or data with up to 20 levels.

Contact Fred Bullock of Personal Computers Ltd, 194 Bishopgate, London EC2M 4NR (tel: 01-626 8121).

HERE'S another useful routine by John Rennell for formatting numbers as Pounds and Pence. Perhaps this will make up for the omission of part of the routine from the original program in last month's Windfall.

```
]LIST

10  REM  POUNDS/PENCE
20  REM  FORMATTING ROUTINE
30  REM  LINES 75 TO 77
40  REM
41  REM  THE NUMBER TO BE
42  REM  FORMATTED SHOULD
43  REM  BE PLACED IN LP$
44  REM  THEN GOSUB 75
45  REM
46  REM  THE FORMATTED
47  REM  NUMBER WILL BE IN
48  REM  LP$. NUMBERS
49  REM  GREATER THAN 9 MILLION
50  REM  WILL NOT BE FORMATTED
51  REM  CORRECTLY
52  REM
53  REM  +VE NUMBERS ARE ROUNDED
54  REM  DOWN. -VE NUMBERS ARE
55  REM  ROUNDED UP
56  REM  TO ROUND TO NEAREST
57  REM  PENNY CHANGE 'R' TO
58  REM  0.5
59  REM
60  REM  'R' MUST BE DEFINED
61  REM  ELSEWHERE IN THE
62  REM  PROGRAM
70 R = 0.001: GOTO 100
75 L1 =  VAL (LP$) * 100: IF L1 <
     0 THEN R1 = 1:L1 =  ABS (L1)

76 L1 =  INT (L1 + R):LP$ = "000"
      +  STR$ (L1):L2$ =  RIGHT$
     (LP$,2):L3 = L1 -  VAL (L2$)
     :L3 =  INT (L3 / 100):L4$ =
      STR$ (L3):LP$ = L4$ + "." +
     L2$: IF R1 = 1 THEN LP$ = "-
     " + LP$
77 R1 = 0: RETURN
100  TEXT : HOME
110  INPUT "ENTER A NUMBER ";LP$
120  GOSUB 75: PRINT
130  PRINT "THE NUMBER ENTERED IS
     $";LP$;
140  GET A$: GOTO 100
```

Sorting out that space program

WHEN is a space not a space but a nuisance? The answer is in a listing which you wish to edit on the Apple screen.

In order to give you a better view of your mistakes Applesoft inserts a seemingly random number of spaces in the program lines as it lists them. Applesoft usually ignores these spaces during editing.

There are, however, at least two occasions when these spaces will become a permanent part of your program. If spaces have appeared between quotation marks – that is, within a string – these will be added to the program as the cursor passes over them during editing. This plays havoc with screen formatting.

An even more insidious effect occurs during the editing of lines containing data statements. Spaces can become appended to the data and cause all sorts of problems.

The means of avoiding these and other possible difficulties lie at your fingertips. If you reduce the width of the text screen to 33 characters wide Applesoft will stop trying to "help" you and will print the lines as you have typed them in. The current width of the text screen is held in location 33 (decimal). The command POKE 33,33 will have the desired effect.

It is useful to have a standard procedure when editing. The method I use is:

1. Press SHIFT and ESC while simultaneously pressing P. (That is, press all three keys at the same time.) This clears the screen and will prevent the rather confusing display which will result if you already have text on the screen.
2. POKE 33,33 (set width of text page).
3. Type LIST.
4. Carry out editing as required.
5. The text page can be returned to normal (40 characters wide) if you:
   Press RESET
   Type TEXT
   POKE 33,40.

It is good practice to include a TEXT command early in any program to ensure the display is correct.

I make no apology for expanding on what already appears in the Apple manuals, as I am sure that many new Apple users start producing programs (and hence mistakes) long before they have read the manuals from cover to cover. – **P F Brameld, Department of Polymer & Fibre Science, UMIST.**

Stringing along

THIS technique allows you to mix strings, print statements and 'DIRECT' commands in your EXEC files.

When run this program writes an EXEC file that will display Applesoft's pointers without altering your program.

The way to put commands on file is to print them inside quotes. Try that on strings, though, and double quotes cancel with strange results.

I use CHR$() so that new quotes appear as old ones vanish – **N. KELLY**

```
10 D$ =  CHR$ (4): REM CTRL-D
20 P$ =  CHR$ (27) +  CHR$ (68) +
      CHR$ (63) +  CHR$ (34):P$ =
     P$ + "  ": REM

 P$ = ESC D + ?" + TWO SPACES

30 Q$ =  CHR$ (34): REM ONE QUOTE

40 F$ = "PF": REM  POINTER FILE
50  PRINT
60  PRINT D$"OPEN"F$
70  PRINT D$"WRITE"F$
80  REM
90  PRINT P$"  APPLESOFT POINTERS
     ."Q$
100  PRINT P$"HIMEM SET TO     "Q
     $"PEEK (115) + 256 *  PEEK (
     116)"
110  PRINT P$"STRINGS DOWN TO  "Q
     $"PEEK (111) + 256 *  PEEK (
     112)"
120  PRINT P$"NUMERICS UP TO   "Q
     $"PEEK (109) + 256 *  PEEK (
     110)"
130  PRINT P$"VARIABLES UP TO  "Q
     $"PEEK (107) + 256 *  PEEK (
     108)"
140  PRINT P$"LOMEM IS SET TO  "Q
     $"PEEK (105) + 256 *  PEEK (
     106)"
150  PRINT P$"PROGRAM TOP IS @ "Q
     $"PEEK (175) + 256 *  PEEK (
     176)"
160  PRINT P$"PROGRAM BOTTOM @ "Q
     $"PEEK (103) + 256 *  PEEK (
     104)"
170  PRINT D$"CLOSE"F$
180  END
```

Buy in haste, repent at leisure

By ANDREW JOHNS

BUYING an Apple computer is relatively easy. Buying a printer to go with it is another matter. The cost of adding a printer to an Apple system could be, in many cases, as expensive as the initial computer. And if it doesn't produce the results you want could become so much worthless junk.

Printer prices range from a couple of hundred to many thousands of pounds. There are slow and fast printers, different ways of producing print, with graphics and without. And that's before you start thinking about noise levels, size and reliability.

The first step is to decide exactly what the printer is to be used for. A programmer, for instance, might well be quite happy with one having a restricted width of 40 columns. This is quite adequate to transfer listings produced on a screen onto hard copy for easier reference and the cost would not be high.

But as soon as he starts to produce commercial or statistical packages calling for formatted output, a printer wide enough to show how his program is developing from a user point of view is needed.

The quality of the print need not be high, unless he wishes to start using the word processing packages available to produce reports based on his software, whereupon the increased mass of text in both upper and lower case needs to be both legible and easy on the eye. The quality of character formation then becomes important.

If, however, he wishes to use the same program to produce personal or business letters, the daisy wheel or letter quality printers must be considered to give correspondence a professional look. As soon as this is done printing speed diminishes drastically and costs increase heavily.

The secret of successful purchase, therefore, is a thorough analysis of all possible uses of the machine, both now and in its projected life span, and a careful optimisation of the features available which meet the user's requirements.

This may mean one may have to discard some features of relatively minor importance, but rest assured that there are no printers which are the ultimate in performance at any price. If your requirements are diverse, and let's face it, the Apple encourages use for many different applications, the printer will probably perform well on the main task and not so well on secondary tasks.

Because of the number of printers available for the Apple this review will concentrate on those which use a dot-matrix to form their characters. They make up the cheapest and most widely used range and many possess features which go beyond standard text production.

However, the full range of printers available include:

● **Thermal:** A small rectangular chip is heated in a matrix configuration corresponding to the character to be printed. This then etches the shape onto heat sensitive paper. Although these are generally the cheapest type available the paper tends to be expensive and only single copies can be produced. The images tend to fade after a period of time and care must be taken in storage and copying the printout.

● **Electrostatic:** The same effect is created with these printers although the image is transferred by the use of sparks.

● **Dot-matrix impact:** This is the most common type and uses magnets to fire pins at an ink impregnated ribbon, transferring the matrix image onto paper. The quality of the printout is constrained by the size of the matrix but any type of paper can be used. Most printers of this kind use continuous stationery pulled through by tractor or pin feed but some are able to handle single sheet paper as well.

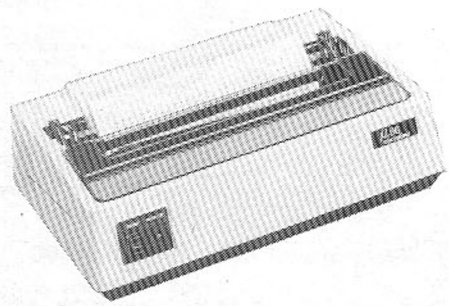
● **Letter quality:** These have typewriter type print heads such as golf-balls, thimbles and daisy wheels. They are slower than dot-matrix printers, but the quality of their output is much higher and can be used for business letters. They used to be more expensive than the standard dot-matrix kind but even here, in spite of the more robust and sophisticated construction, prices are falling.

● **Ink jet:** Ink sprayed onto paper in a precisely controlled format produces very high quality output, sufficiently good to be considered in rhe realm of phototypesetting, but these printers are very expensive.

● **Laser printers.** Forget it, unless you want your Apple to produce output at the rate of 2,000 lines a minute and can afford to spend in excess of £120,000.

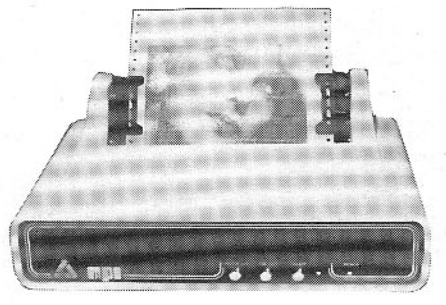
Dot-matrix printers are capable of providing all the capabilities that the average Apple user requires with the exception of professional quality letters. Just to digress for a moment, the

→



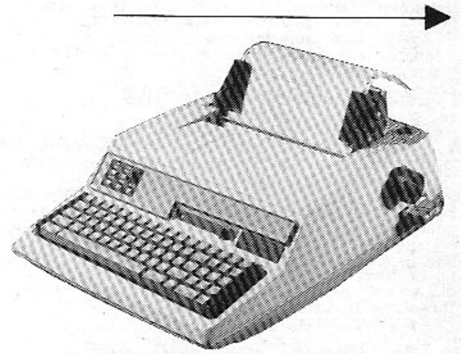
Okidata Microline Series

Manufacturer: Okidata. **Type:** Impact dot-matrix. **Paper feed:** Tractor, friction. **Width:** 9.5in. **Columns:** 80, 132 compressed (80 and 82A); 136, 230 compressed (83A). **Speed:** 80cps (80); 120cps (82A, 83A). **Character format:** 9 x 7 (80); 9 x 9 (82A, 83A). **Character set:** 96 Ascii. **Graphics:** 64 block graphics. **Interfaces:** RS232C, Centronics parallel. **Buffer:** One line buffer. **Ribbon:** Spool. **Other features:** Screen dump facility with Tymac parallel interface; super Pix Oki graphics software supplied with interface. **Price:** £299 (80); £425 (82A); £799 (83A). **Main supplier:** Northamber, Great Oak House, Albany Close, Esher, Surrey. (tel: 0372 62071).



MPI 88G

Manufacturer: Russett Instruments. **Type:** Impact dot-matrix. **Paper feed:** Friction or tractor. **Width:** 9.5in. **Columns:** 80, 132 in compressed mode. **Speed:** 100cps; bidirectional. **Character format:** 7 x 7 dot matrix for data printing; 11 x 7 dot matrix for letters. **Character set:** 96 Ascii chars. **Graphics:** Dot-addressable; vertical res; 72dpi, horizontal 50-82dpi. **Interfaces:** Serial RS232C; parallel (Centronics). **Buffer:** 1k standard, 2k optional. **Ribbon:** Cartridge. **Price:** £420. **Main supplier:** Russett Instruments, Unit 1, Nimrod Way, Nimrod Industrial Estate, Reading, Berkshire (tel: 0734 868147).



Weyfringe KSR II

Manufacturer: Weyfringe. **Type:** Impact dot-matrix. **Paper feed:** Tractor; friction. **Width:** 9.5in. **Columns:** 132 at 16cpi. **Speed:** 112cps; logic seeking, bi-directional. **Character format:** 7 x 9 dot-matrix. **Character set:** 96 Ascii + control characters. **Interfaces:** RS232C. **Buffer:** 1,780 chars FIFO. **Ribbon:** Cassette ribbon. **Other features:** Designed for fast data entry with FIFO memory of 16 characters on keyboard, and N-key roll over as standard. Key release not required for acceptance of further characters. Self-diagnostics with LED display. **Price:** £1,095. **Main supplier:** Weyfringe, Longbeck Road, Redcar, Cleveland TS11 6HQ (tel: 0642 470121).

difference between using a word processer and a dot-matrix printer for your mail is probably the same as the old habit of using a fountain pen as against a biro before biros became socially acceptable.

The first requirement of the printer is whether it is capable of actually doing the job. If it is only required to list out programs the answer will undoubtedly be yes. If you are producing invoices, however, it must be wide enough to allow the invoice to be formatted to include all relevant columns. Most invoices and ledger systems are designed to run on 80 column width stationery.

Reports, however, are better seen on wider paper, up to 132 columns wide, allowing a whole year's figures to be columnated without drastically squeezing in headers and chopping figures. Certain professions, such as lawyers, occasionally like documents even wider and this can be provided in two ways. There are manufacturers who produce printers with carriage widths in excess of 15in, which are quite reasonably expensive.

Most, however, have a facility to compress print so that an 80 column printer can compress its characters to give you 132 columns and a 132 column printer can extend to at least 220 columns.

Once the width has been considered the next most important feature is the type of paper to use. Most dot-matrix printers are designed to handle continuous stationery and cannot take single sheets. If you want to print addresses on envelopes or need to change stationery continually a printer with a pressure or platen feed is needed. Rollers on tractor or pin feed printers are loose enough to allow unimpeded throughput of paper. They are there purely as a guide.

A friction feed printer needs to grip single sheet paper sufficiently tightly to feed it through as accurately as a tractor feed. Both types should be capable of handling multi-part stationery, giving the last copy – up to 12 on some – a readable impression.

The point about changing stationery needs expanding slightly. When trying to choose a printer, visualise how it is going to be used. If it is in an office environment, being used for accounts, producing quotations, occasional customer statements and reports, the situation may occur where a customer makes a transaction and wants an invoice.

A second customer walks in and asks for a copy of his account. The manager then asks for a list of current stock or a quotation for a third customer. If the printer is geared only towards continuous stationery the paper has to be changed between each transaction, or all transactions and invoices have to be recorded on plain listing paper.

Being able to batch your work gives you more scope over the choice of machine. Incidentally, individually headed and good quality notepaper can be provided in continuous format although it is rather expensive. This also applies to envelopes.

The next consideration is speed. This is relevant in a commercial environment where work is batched with a print run at the end of the session. If you are using a printer at 80 characters a second and are producing a payslip run for 100 staff with each requiring 12 lines, the time taken to print all of them would be more than 30 minutes, including record search times. If you are using one rated at 30cps the time could be three times that. Reasonably priced dot-matrix printers run from 30cps

Seikosha GP80A

Manufacturer: Seikosha. **Type:** Impact dot-matrix. **Paper feed:** Tractor. **Width:** 8in max. **Columns:** 80. **Speed:** 30cps. **Character format:** 5 x 7 dot-matrix. **Character set:** 96 Ascii char + 32 European. **Graphics:** Dot-addressable. **Interfaces:** Serial RS232C, Parallel. **Buffer:** One line of data. **Ribbon:** Endless ribbon with inked roller. **Other features:** Repetitive graphics; self test. **Price:** £225. **Main supplier:** DRG Business Machines, 8 Lynx Crescent, Winterstoke Road, Weston Super Mare, Avon BS24 9DW (tel: 0934 416392).

Epson MX-80, Epson MX-80F/T2

Manufacturer: Shinsu Seiki. **Type:** Impact dot-matrix. **Paper feed:** Tractor (MX-80); Friction/tractor detachable (MX-80F/T). **Width:** 10in. **Columns:** 80, 132 cols at 16.5 chars/in. **Speed:** 80cps. **Character format:** 9 x 9. **Character set:** 96 Ascii chars with descenders. **Graphics:** 64 block characters; Graphics support available in software and on Interface cards (Simon). **Interfaces:** Centronics parallel; optional RS232C. **Ribbon:** Cartridge. **Other features:** Emphasised print; low noise. **Other models:** MX-82; MX-80 Type 2, TX80. **Price:** MX-80 £360; MX-80 FT2 £449. **Main supplier:** Epson, Sherwood House, 176 Northolt Road, South Harrow HA2 0EB (tel: 01 422 5612 for nearest supplier).

Paper Tiger 560

Manufacturer: Integral Data Systems. **Type:** Impact dot-matrix. **Paper feed:** Tractor. **Width:** 14.75in. **Columns:** 132, 220 in compressed print. **Speed:** 150cps proportional; 110 cps mono-spacing. **Character format:** 24 x 9 dot-matrix, overlapping horizontally and vertically. **Character set:** 96 Ascii characters. **Graphics:** 84 x 84 dpi; software routines available. **Interfaces:** RS/232C, parallel. **Buffer:** 2k standard. **Ribbon:** Cartridge ribbon. **Other features:** Eight software selectable character sizes; automatic justification; fine positioning for word processing; acoustic cover; lower case descenders. **Price:** £995. **Main supplier:** Teleprinter Equipment, 70-82 Akeman Street, Tring, Herts (tel: 0442 82 4011).

Paper Tiger 460, 445

Manufacturer: Integral Data Systems. **Type:** Impact dot-matrix. **Paper feed:** Tractor. **Width:** 11.5in. **Columns:** 80, 132 in compressed mode. **Speed:** 80 cps. **Character format:** 12 x 9 dot matrix, overlapping horizontally and vertically (460); 7 x 7 dot-matrix (445). **Character set:** 96 Ascii characters. **Graphics:** Dot-addressable; software facilities available. **Interfaces:** Serial RS/232C, Parallel; Centronics (460 only). **Buffer:** 2k standard. **Ribbon:** Cartridge. **Other features:** Proportional spacing; automatic justification; fine positioning for word processing. **Price:** £595 (445); £835 (460). **Main supplier:** Teleprinter Equipment, 70-82 Akeman Street, Tring, Herts (tel: 0442 82 4011).

Qantex 6000, 6010

Manufacturer: Qantex. **Type:** Impact dot-matrix. **Paper feed:** Tractor; front or bottom entry. **Width:** 15.5in. **Columns:** 136. **Speed:** 150cps; bi-directional, logic seeking. **Character format:** 9 x 9 dot matrix; lower case descenders; underline. **Character set:** 96 Ascii chars. **Interfaces:** Parallel, Serial RS232C. **Buffer:** 6000 one line buffer; 6010 2k buffer. **Ribbon:** Cartridge. **Other features:** Low noise integral sound cover; self test; high reliability design. **Price:** £749. **Main supplier:** Northamber, Great Oak House, Esher, Surrey KT10 9BR. (tel: 0372 62071).

EG800M 80 column printer EG1000FT 136 column printer

Manufacturer: Shinsin Seiki. **Type:** Impact dot-matrix. **Paper feed:** Tractor. **Width:** 10in (EG800M); 12in (EG1000M). **Columns:** 80 at 10cpi, 132 at 16.5cpi (EG800M); 136 at 10cpi, 233 at 16.5cpi (EG1000M). **Speed:** 80cps; logic seeking, bi-directional. **Character format:** 9 x 9 dot-matrix. **Character set:** 96 Ascii char. **Graphics:** Dot-addressable; 64 graphics chars; 72 x 60 or 72 x 120 bpi. **Interfaces:** Parallel standard; RS232C optional. **Ribbon:** Cartridge. **Other features:** Self test; enlarged and compressed print; low noise, lower case descenders. **Price:** £370 (EG800M), £575 (EG1000M). **Main supplier:** Electrographic Peripherals, Printinghouse Lane, Hayes, Middlesex UB3 1AP. (tel: 01-573 1826).

Epson MX-100

Manufacturer: Shinsui Seiki. **Type:** Impact dot-matrix. **Paper feed:** Friction/tractor detachable. **Width:** 15.5in. **Columns:** 136; 233 in compressed mode. **Speed:** 80cps; bidirectional. **Character format:** 9 x 9 dot-matrix. **Character set:** 96 Ascii chars with descenders. **Graphics:** Dot-addressable 72 x 120 dots per inch; software support available. **Interfaces:** Centronics style parallel, RS232C. **Ribbon:** Cartridge. **Other features:** Emphasised and double printing; low noise operation. **Price:** £575. **Main supplier:** Epson UK, Sherwood House, 176 Northolt Road, South Harrow HA2 0EB (tel: 01 422 5612 for nearest supplier).

Anadex DP9500

Manufacturer: Anadex. **Type:** Impact dot matrix. **Paper feed:** Tractor. **Width:** 13.5in. **Columns:** 132, 220 in compressed mode. **Speed:** 50-200 plus lpm, up to 200cps. **Character format:** 9 x 9, 7 x 9, 11 x 9; switchable, bi-directional, logic seeking. **Chracter set:** 96 Ascii. **Graphics:** Dot addressable, 72 x 60 dpi. **Interfaces:** RS232C. **Buffer:** 600 char; 2k optional. **Ribbon:** Cartridge. **Other features:** Graphics interface available; other models in range. **Price:** £995. **Main supplier:** Anadex, Weaver House, Station Road, Hook, Nr. Basingstoke, Hants RG27 9JY. (tel: 025 672 3401).

Silentype

Manufacturer: Apple. **Type:** Thermal. **Paper feed:** Friction/heat sensitive paper. **Width:** 8in. **Columns:** 80. **Speed:** 80cps. **Character format:** 7 x 9 dot-matrix. **Character set:** 96 Ascii characters. **Graphics:** Screen dump facility. **Interfaces:** Own parallel interface. **Price:** £349. **Main supplier:** Microsense Computers, Finway Road, Hemel Hempstead, Herts. (tel: 0442 40472).

Integrex CX80 Colour Matrix Printer

Manufacturer: Integrex Epson. **Type:** Colour impact dot-matrix. **Paper feed:** Tractor. **Width:** 10in max. **Columns:** 80. **Speed:** 55 lpm. **Character format:** 5 x 7 (Characters); 6 x 7 (graphics). **Character set:** 96 Ascii chars., 15 chars programmable. **Graphics:** 64 graphics chars.; colour dump card now available; dot-addressable. **Interfaces:** RS232C; Optional Apple interface; plus Colour Dump Card. **Ribbon:** Multi-coloured ribbon to produce seven print colours. **Other features:** Self test, elongated and reverse printing; microstepping for word processing applications. **Price:** £895. **Main supplier:** Integrex, Church Gresley, Burton-on-Trent, Staffordshire DE11 9PT. (tel: 0283 215432).

Centronics 737, 739 and 150

Manufacturer: Centronics. **Type:** Impact dot matrix. **Paper feed:** Roll paper, tractor, friction. **Width:** 8.5in. **Columns:** 80, 132 in 16.5cpi. **Speed:** 80cps proportional, 50cps monospaced (737), 100cps monospacing (739), 150cps bidirectional logic seeking (150). **Character format:** N x 9 proportional, 7 x 8 monospaced (737 and 739), 7 x 7 (150). **Character set:** 96 Ascii chars; optional international set. **Graphics:** Dot-addressable, 74 x 72 dpi. **Interfaces:** Centronics Parallel, RS232C Serial (739 and 150). **Buffer:** One line buffer (150 serial only). **Ribbon:** Continuous with mobius loop (737, 739); cassette (150). **Other features:** Self test; quiet operation; prints subscripts and superscripts; adjustable snap-on tractors (150). **Price:** £399 (737); £499 (150); £504 (739). **Main supplier:** Centronics, Victoria Way, Burgess Hill, Sussex. (tel: 044 46 45011).

up to about 150cps. There are faster models available, but at greater cost.

The quality of the printout is dependant on the size of the matrix which forms the characters and whether the matrix can incorporate descenders in lower case mode. A fairly common matrix configuration is 5x7 dots which enables the printer to form an easily recognisable character and allows certain characters in lower case to project below the bottom level of the text.

High quality printout is more often achieved with a 7x9 matrix. The quality can really be detected by the position of the lower case "j", which should have an extension below and above the standard. Some printers have the capability of switching the matrix configuration.

If a printer is to be used for reproducing graphics output from the screen the facilities available on each printer must be known. There are some models which are unable to be used to create shapes other than the standard Ascii set. These tend to be of the older generation. All newer printers have the capability of producing graphics but the means of accessing the facility differs in each one.

This is not a problem of the machine itself but rather the amount of work done by a third party to create software or hardware routines which will allow you to produce graphics without having to learn dot-addressing techniques in machine code. Make sure also that the routines provided are capable of being either inserted into your own programs, or callable as sub-routines from Rom.

With these routines – or, if you write your own routines – a picture or graph can be moved from the screen directly onto the printer. Depending on the routines available this picture can then be enlarged, lengthened or printed in inverse representation.

Standard features on many machines include the facility to treat characters in the same way, to produce double or treble size text, embolden text or invert text. Another new feature is the ability to print multi-coloured text. This is superb for the presentation of statistical analysis like bar charts and graphs.

Once you have a printer that matches your requirements and if there is still a choice between models, things like size, noise level, ease of maintenance and cost must be considered. The mechanism for producing an impact on dot-matrix printers is not dependent on achieving a heavy strike as on a letter quality printer, and machines can, therefore, be constructed smaller and in lighter materials.

The operating noise is also considerably reduced although there is still a fair degree of difference between the various dot-matrix printers. A noisy machine hammering away all day can be a big nuisance in a small office. Although acoustic covers are available for most of them, it's much better to start with a quiet one in the first place.

Back-up for maintenance is important, more so than on the Apple computer itself (although this is also essential). Because of their greater use of moving parts printers are more susceptible to small faults which require correction by an experienced engineer. Ensure that the supplying company has the capability to support the printer as well as the Apple or can arrange a secondary back-up at short notice.

Finally, there's the question of cost. To use the computer seriously you must buy the printer that can do your job, not the printer you can afford. Think about it.

LAST month Windfall described how networks of Apple computers could be linked together to share data and peripherals. This month we go a step further and look at the use of Apple computers linked into communication networks with larger computers. The problems are inherently different, so the first part of this section will be a brief introduction to data communications, followed by a look at a package currently on the market which enables Apple users to access one of Britain's best known information services – Prestel.

Making the most of the modem

By DAVID CHADWICK

DATA communications is a method of transferring information between two places, requiring a transmitter or record source, a transmission medium or path, and a receiver.

If we look at the simplest and most common example, the inter-connection of two telephone subscribers, the caller is the transmitter, the telephone line connecting them is the transmission medium, and the called subscriber is the receiver.

However the transmitter needs to translate the source information from its original form into a form that can be carried along the path. The transmission medium must then pass the signal along its length without introducing an excessive amount of distortion or noise, and the receiver must concert the signal back into something that can be understood by the receiver, whether it is a man or a machine.

Our present communications network, the telephone system, was designed and set up in the first half of the 20th century, before it was ever imagined that it could be used for transmitting anything else besides the voice. The facilities that were provided were obviously not developed to transmit signals at speeds that digital computers require.

Similarly computers were not specifically designed to be connected to the existing communications network. The techniques for linking computers with peripherals are not adequate for connecting devices over telephone lines.

Despite the inadequacies of the telephone network, a number of different forms of linking have been developed, specially designed for computers.

The standard telephone network, where a circuit is established between two points and then held until the transfer of data is completed, is called a circuit switched system. Message switching, where discrete entities of data called messages are accepted, transmitted and received in their entirety,the transfer being controlled by computer, require dedicated lines.

Packet switching is a form of message switching, where the message can be broken down into discrete lengths, each packet being sent by different routes to its destination where it is reassembled into a complete message. This method allows the computer to optimise the traffic on high speed lines, allowing faster and more efficient use of a network.

A distributed processing network is one where the data processing is divided among smaller computers throughout the network. The use of multiple computers provides greater reliability, and each group of terminals within such a network has the ability to transfer information to any of the computers.

At the moment the most relevant type of data communications system for micro-computer users is the circuit switched system.

The next issue of Windfall will describe methods of linking Apple computers to mainframes to imitate the action of indigenous terminals, like the IBM 2780 and 3780. This occurs where Apples and other terminals are required, because of the extra facilities they offer and their cheaper cost, to emulate the standard terminals provided by the manufacturer of the computer.

How is such a communications link set up? First of all, data is entered from a terminal, either a standard terminal or an Apple. The terminal encodes the method into bits and then transmits them serially into a transmitting modem (modulator/demodulator). This converts the digital signal into an analog signal which is then transmitted to another modem at the receiving computer's end of the transmission line, which reconstructs the transmitted data stream in digital form and passes it to the computer.

Each piece of equipment selected must be compatible with other elements of the system. Besides ensuring the compatibility of the system the other criteria to be considered include speed, half-duplex or full-duplex operation, synchronous or asynchronous transmission and the type of line being used.

At the computer end of the network the host computer must be fitted with a processor interface. This is the device which establishes a port providing set patterns which terminal manufacturers can conform to. It also prepares the incoming data for processing by the computer.

The software part of the interface controls the new equipment added to the system. This includes terminal and interface drivers and any other programs which are required to interface with applications programs, all of which are added to the operating system.

Telephone lines were designed to carry analog signals, and the data from the computer must be converted into this type of signal for transmission. The modem converts the digital signal at one end of the path and reconverts it at the other end.

There are three types of modem – those used for transmission over voice-grade lines, those for very high speed transmission, and hard wired limited distance modems, operating at the highest speeds.

Voice-grade modems operate up to 300bps (bits per second) for low speed systems and up to 9,600bps for medium speed systems. High speed modems operate from 9,600bps upwards.

Low speed modems are used primarily for the dial-up network and for interfacing low speed asynchronous terminals such as teletypes. Medium speed modems are used with terminal controllers and faster terminals, and high speed modems are used in data transmission between computers, or for high speed batch terminals.

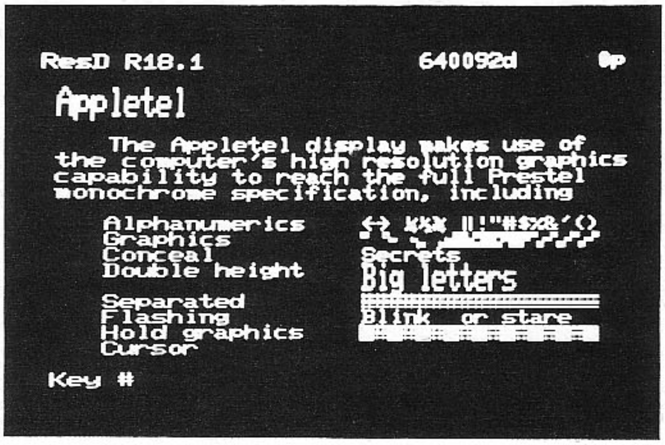
The transmission lines are also divided into three different types. There is the low speed (sub-voice) line, usually used for telex, TWX and other teletype transmissions and which can operate up to 300bps. The medium speed (voice-grade) line allows transmissions at speeds up to 4,800bps on switched lines, and up to 9,600bps on leased lines. This constitutes the public telephone network. The high speed or wideband lines can operate at 19,200bps or more, and are used primarily for intercomputer links.

A switched line is provided with a dial-up service, and on this system costs vary dependant upon connection time. The line is shared with other subscribers. A leased line is a private connection between points, where a constant connection is made. Costs are therefore high but static.

Because of the constant link higher speeds can be maintained and the lines are more secure. There are higher grade lines for connections between computers, with more lines per cable, for more efficient use of the line because of the facility for simultaneous two-way transmission at high speeds.

'Bits per second' is the usual

COMMUNICATIONS (Part 1)



way of denoting transmission speed. Another common expresssion is 'baud'. A baud refers to the number of signal events per second. If each bit represents one event then the speed is expressed in bits per second. The rates are not always similar.

Terminals are classified, however, by the number of characters they can process in a second, and there are three categories here as well. Standard terminals operate at speeds up to 30cps (characters per second), medium speed terminals operate up to 480cps, and high speed terminals operate over 480cps.

Data can be transmitted by a terminal using one of several techniques. Most standard and many medium speed terminals use the asynchronous mode of transmission. This means that every time a character is generated on the keyboard it is wrapped up in a start bit and a stop bit and despatched to the host computer.

The speed of transmission depends on the speed of the operator, and there can be periods of inactivity between characters of a message. Bits within a message are sent at prescribed time intervals, and timing is established independently in the host computer and the terminal.

Synchronous transmission, available with more sophisticated terminals, occurs when long blocks of data are sent with only a single framing pattern at the beginning of the block. Synchronous is so termed because sync characters prefacing the message synchronise the terminal signal with the host computer signal, allowing the unheralded flow of blocks of data. The sync characters are used as infill between transmissions. The timing in such a system is established and maintained by transmitting and receiving datasets.

In addition to this terminals have three modes of operation. Simplex is a form of one-way transmission only, used where a terminal is used only for output. Half-duplex allows two-way communication, but in only one direction at a time, and full-duplex allows two-way communication, transmitting and receiving simultaneously.

Setting up the system

IF an Apple is to be used in a communication network it needs to be set up to conform to the specifications listed above. Taking each consideration in turn, we first of all need to establish a compatible link with the host computer.

This is achieved by the addition of a standard Apple communication card, or a viable alternative such as the useful CCS asynchronous serial interface card which converts the output of the Apple into a serial configuration suitable for transmission.

Each card requires a slightly different wiring configuration based on the standard three wire serial connection. The handbooks provided give sufficient assistance to enable even the most inexpert user to set up the connection.

The communications card requires no additional software to enable you to start communicating straight away in teletype mode. Other, more sophisticated cards are used to provide more elaborate functions, and these have to have inbuilt software on the card or require routines to be written. They allow the Apple user to start interrupting the signals coming from the host computer, or to transfer files direct from the line to disc, or to spool straight onto a printer.

An Apple in teletype mode can be used to communicate directly with the host computer. It can also be used to dump data from the host directly onto a printer for hard copy by implementing printer commands concurrently with the access commands.

While this function can be very useful, it is a bit of a waste of the capabilities of the Apple. Operating an Apple in this way just replaces the action of the teletype, which has been in operation for more than 40 years, with a slightly more sophisticated version which displays data on the screen. You might just as well continue using the teletype which, while noisy and slow, is very cheap.

Far better to use one of the packages available, so that connection can be made within the total operating system and you can still retain control via the Apple to use your discs and printers as and when required.

BRITISH Telecom's Prestel system provides a tremendous amount of information for people with Prestel terminals. Now Apple users can access the same information, as described here by CLAIRE BACKENBURY of Owl Computers.

Appletel gets the facts fast

PRESTEL is the viewdata service developed by British Telecom to utilise the link between computers, television and the telephone network. Its function is to prove easy access to a vast amount of information to anyone who has suitable equipment.

Such equipment must, to attract users, be reasonably priced for purchase or rental and be simple to operate. The basic Prestel system, as supplied by British Telecom, consists of a Prestel terminal with a jack plug connection to the telephone line.

There is a modem between the jack plug connection and the terminal. This is protected from the high voltages of the cathode ray tube by some kind of barrier device. In Prestel sets the modem is actually incorporated into the set. The modem is set up to send data at 75 baud and to receive at 1200 baud.

Once the system has been set up it is registered with a terminal identity code, programmed remotely into the terminal. This operation is carried out at the appropriate regional Prestel centre. The Prestel terminal is usually operated in the home or office by a 12-key numeric pad, containing the numbers 0-9 and the two symbols * and # . This pad may work the set by remote control or it may be wired in directly.

When the terminal is not being used to access Prestel it acts as an ordinary TV set. This means that the user does not have to purchase a specific Prestel terminal, which may be thought to be an extravagance.

As well as Prestel users who need only the simple sets described here, with the possible addition of a printer for creating hard copy, the Information Providers must also be considered.

An IP creates frames and therefore needs full alphanumeric and graphic input capabilities. He must also be able to

→

create a frame table for directing, or routeing, the frames from one to the next.

The IP's system, therefore, has full editing facilities with a complete keyboard. This generally incorporates various features such as a 3x2 matrix section to select graphics shapes more easily and single-key entry of some coded groups.

There is now a great deal of information on Prestel – currently about 180,000 frames. These cover a multitude of topics, ranging from highly technical stock market reports to children's stories and games.

When a user enters Prestel he may either follow the indexing through to frames of interest to him or he may simply call up specific frames using the simple command *n# , where *n* is the frame number.

Indexes on Prestel are easy to follow. The user merely has to type the digit corresponding to his choice, or some frames will require him to press # to get to the next frame in a sequence.

Prestel is paid for in three ways. Firstly the Information Provider, who supplies the information on the frames, has to pay a fee for his frames. This works out rather expensive as he must pay a flat fee as well as a charge for each frame. Many IPs buy large blocks of frames and sub-let them in small blocks to other IPs. These large purchasers are known as "Umbrella" IPs.

This is a mutually convenient arrangement as the Umbrella IP can get frames much more economically, while the smaller IP obtains the unit he requires, rather than having to take blocks of 1,000 frames, the standard letting.

The other charges for Prestel are paid by the user. There is a Prestel charge – approximately equivalent to a local telephone call charge – while he is on-line and there may additionally be a frame charge, between 0.1p and 50p, for viewing the frame.

These charges are entered on a special Prestel bill and are separate from the normal telephone bill. The phone bill, however, will contain the charge for the actual use of the telephone line while linked to Prestel. This is charged at the normal rate.

Whereas Prestel has provided a system based on the domestic TV set, Apple II owners can access Prestel as users or IPs on an Apple-based system, Appletel. Like a Prestel terminal, it is linked into Prestel via a barrier device and modem, and Prestel is called up by telephone.

This system, developed by Mike Gardner of Owl Computers, allows the user to access Prestel quickly and easily. Frames can be stored on disc in a few seconds and studied later, off-line, when it will cost nothing.

The system operates in black and white, but can have an Appletel colour card and an RGB colour monitor added for full Prestel colour. It is essential to have an Appletel colour card rather than any other card as Prestel graphics are generated differently from Apple graphics, so the two colour systems are incompatible.

Using Appletel, frames can be created and entered onto Prestel. Appletel acts as a normal dumb Prestel terminal for most of the time and routeing is carried out online in the normal fashion. The frames themselves, however, can be created and edited off-line and sent down to Prestel one at a time.

Because of the presence of the Apple computer in the system an experienced programmer can write programs in Basic to extract selected information from Prestel or to send commands to Prestel.

For the Information Provider a further package is available, again from Owl Computers. This is the Appletel Professional Editing System (APES) which has several extended features, including online editing, local database use, various utilities, printing and (though not yet released) bulk update.

As with any of the on-line alternatives the on-line editing feature operates the system purely as a standard Prestel terminal, cutting out the Apple and its related software. Frames can be set up off-line, as with the Appletel system, and the frame table can be included. These can then be sent down the line to Prestel, each frame with its own table.

This gives the Apple IP the advantage of being able to ensure that the routeing is satisfactory before committing the frames to Prestel.

With the ability to route frames, APES is very convenient for those people who want to run a private viewdata system. For a large viewdata database, however, a new product will be available shortly. This is OverView, and is being developed jointly by Zynar and Owl Computers. It will operate using an APES type of system to give full viewdata capabilities in the private network.

The utilities section of APES has many useful features allowing the operator to alter various parameters. It is in this section, for instance, that the system can be informed of the addition of an external editing keyboard, a piece of equipment which greatly simplifies the entry of encoded data while producing frames.

The Owl keyboard has the same general features as an ordinary Prestel editing keyboard, plus a useful "Help" key which takes the operator to a series of instruction frames appropriate to the part of the program he is in when he keys it. There is also a "Command" key which calls up editing features such as word wrap and margin setting.

Viewdata is one of today's most important information systems and is developing rapidly. Using his micro in conjunction with this system the Apple owner has the tremendously powerful resources of viewdata networks, both public and private, open to him at all times.

Appletips

Do you get irritated by having to type **PR#1** and **PRINT "CTRL–I 80N"** whenever you want a quick 80 column listing of your program on the printer? This routine in machine code allows you to do all that with one command – the '&'.

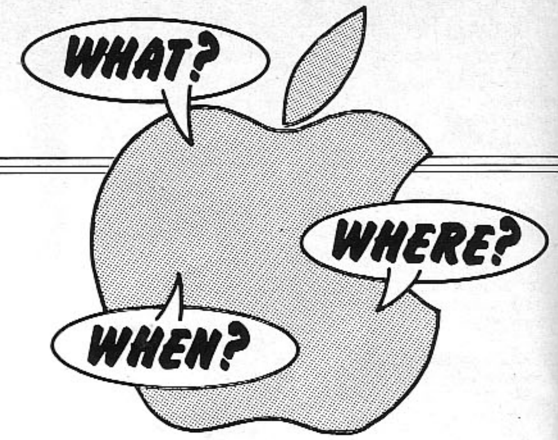
```
+CALL-151
*300.347
0300- A9 00 85 36 A9 C1 85 37
0308- A9 80 20 00 C1 20 51 A8
0310- A2 05 BD 2F 03 20 ED FD
0318- CA D0 F7 38 A9 35 ED F6
0320- 03 8D F6 03 60 A9 00 8D
0328- F6 03 A9 03 8D F7 03 60
0330- 8D CE B0 B8 89 A2 03 BD
0338- 43 03 20 ED FD CA D0 F7
0340- 20 1B 03 60 89 C9 89 00
```

First you **BLOAD** the program and go to part of it with a **CALL 805**. This moves the & pointers to your routine at **$300**. Then, whenever you want the printer come on, you type **& RTN**. Then type **LIST RN** as usual. This gets you full 80 columns printing but no screen. If you don't mind short lines and need the screen then try another **& RTN**. Hey Presto, the screen returns and the printer is still on!

RESET cancels this routine any time. The '&' may be used again but, depending upon where you 'reset' it, you may need an extra **& RTN** before it 'takes'. Try it!

● *This is the corrected version of an Appletip which appeared on Page 41 of the July issue of Windfall.*

WHAT'S NEWS...



More than a bit unfair..

THERE are quite a few versions of the notorious Bit Copiers floating around the country. For the uninitiated, these are discs containing programs which by reasonably sophisticated techniques copy programs where the author has taken considerable pains to prevent such copies being made.

It seems inevitable that the use of these copiers is going to increase, and sooner or later – although they have already been on and off the market – they are going to be widely offered for sale.

We dislike the current direction of this trend. It is plainly dishonest to want to acquire for nothing something which costs time and money to develop, and which is sold as a means of earning a livelihood. The persistent users might like to consider how they would react if some of their potential customers found a way of using their services or products for nothing. All hell would be let loose.

Suppliers of software may themselves be in a position to curb some of the misuse of their products, however, especially those supplying heavily-protected discs for commercial applications. When someone buys software for serious use, such as word processing or financial planning, the customer's first reaction if he knows anything about security is to make a backup disc, so that if the main disc becomes corrupted for any reason he can continue to function while he replaces the original disc. A number of packages are not supplied with more than one disc, and the customer must resort to skulduggery to provide his copy.

With regard to games, it is likely that their high price is contributing to the amount of copying going on. A lower price might not anaesthetise the customer's conscience, but a more responsible attitude might develop towards their purchase.

There are currently a large number of methods in existence for protecting software, but we have yet to find one totally uncopyable by one means or another.

The non-copy guarantees people are asked to sign are pretty far removed from reality, as there are minimal ways of finding out who has your program. The only solution lies with the attitude of the end user.

And it seems unlikely that people will agree that there is a lot of truth in the saying "You don't get out for nowt," when they can see plainly that they do, and in enormous quantities.

• • •

DECISIONMODELLER, son of Micromodeller, made its TV debut last month – on the same day as it was featured in the Sunday Times Business News.

In BBC-2's "The Money Programme," against the background of a giant Apple poster, presenter Valerie Singleton heard from management consultant Geoff Smith how this new package can help businessmen, both in simplifying accounts and in making them more understandable to employees.

She told viewers there was less and less excuse these days for management not to embark on a full-scale reappraisal of their financial methods. For the first time new tools of management were available at a price any company should be able to afford. They offered tighter control of budgets, easier calculations of the effect of changes and better communications.

Geoff Smith said that with Decision-Modeller a manager could recalculate the effect of a decision on prices and values, pay, profits and the amount of debtors and creditors the firm could afford to have and the amount of stock they could afford to keep.

Asked about its value in human relations he said: "It simplifies the communication and explanation of the real purpose of the business to employees. Most of them can readily grasp it. The need to explain is now more important than ever before, and this is where the microcomputer can really score."

• • •

THE impressive range of software to be marketed by Apple under the new Special Delivery Software label is to include a package designed and in use already in this country.

Apple Project Manager – the only British contribution to the list of 23 software packages in the scheme – has been written by Geoff Reiss and Robert Fearnley of Construction Programming Services, Bradford. And it had to undergo rigorous testing by Apple before it could be accepted.

Reiss and Fearnley wrote it as a critical path analysis tool to assist managers in the development of projects. It contains facilities to produce charts, calenders, reports and schedules for a number of large projects simultaneously.

Projects consisting of up to 1,900 individual activities can be handled by APM, which has already seen service in companies like Bovis, Birds Eye, the British Sugar Corporation and a major oil company.

Apple themselves also use APM to monitor their own internal projects. In fact, the package, once known as Micronet, could be used to control the development of future Apple computers and related products.

If you are interested in obtaining more details about the package you could ring Geoff Reiss direct at Bradford, on 0274 671859. The company's address is 11 Meadow View, Wyke, Bradford.

The Special Delivery Software scheme also includes a lot of other useful products, ranging from computer games like the excellent Bridge Tutor at £18 to sophisticated statistical packages like Personal Finance Manager and the most expensive of all, APM, which costs £121.

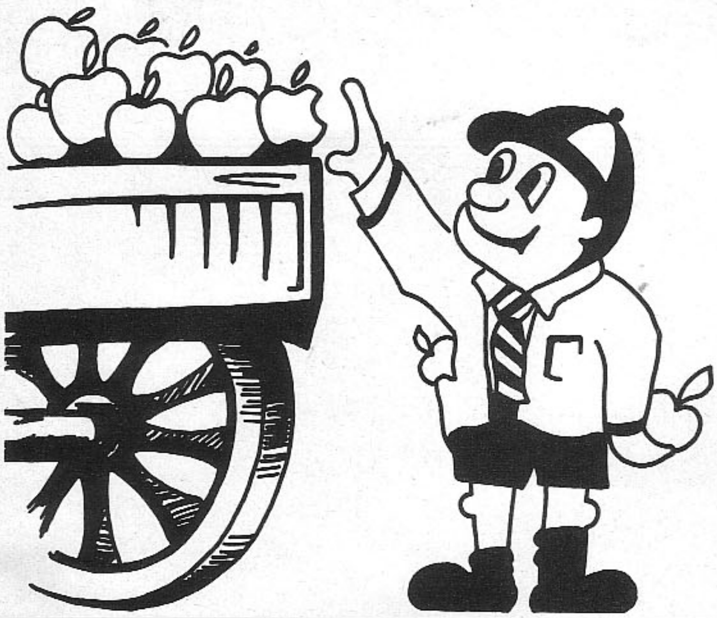
Apple usually writes and markets its own software, occasionally making deals with external suppliers to distribute their products, as with the MicroFocus CIS Cobol recently. The SDS scheme is different in that all of their products are tested and evaluated by Apple and then released under the Apple logo.

It's apparent that people at Apple HQ in California are expecting a considerable increase in the range of Special Delivery Software – and they would like to see more British contributions when their third catalogue comes out next year.

"We're committed to bringing Apple users good, independently written software at a reasonable price," they say. "But we can't do this on our own."

So they want to hear from anyone who can write programs of professional quality. They add: "It makes no difference whether you're a software professional or an amateur with a good idea. What does interest us is whether your program would prove valuable to thousands of other people with similar needs."

If you think you fit the bill, don't write to us but to: Software Evaluation Group, Apple Computer Inc, 10260 Bandley Drive, Cupertino, California 95014, USA.



applecart

Monthly review of Apple in education

CAL 'explosion' is on its way

WAY back in January Microsense Computers decided to make a special spring educational offer to UK schools and colleges. This offer of a 20 per cent discount off recommended retail prices was scheduled to expire on May 31, but was subsequently extended until July 31 due to pressure from schools wishing to buy and from Apple dealers wishing to sell to schools.

The offer has now closed and I am happy to report that more than 700 Apples have been sold during this period, bringing the total number of users in schools and colleges in Britain to well over 1,000. It is a great testament to the capabilities of the Apple that at a time of financial and political constraints it remains more popular than ever with the education community. Also encouraging is the fact that the sales have been widespread throughout the UK and not only into specific areas. Certain regions have taken the opportunity to obtain more Apples, and many others have welcomed the chance to become first-time users, having purchased several or many systems in a block order.

Not surprisingly we are very pleased with the outcome of the campaign. We are now a firmly established supplier of microcomputer equipment to the education community in the UK and our future in this field is assured. What is also highly significant is that as our user base continues to grow so will the amount of educational software for the Apple increase dramatically, as users themselves write programs and as commercial publishers realise the commercial good sense of writing material for the Apple.

I anticipate that 1981/82 will see an explosion in the number of good quality computer assisted learning programs, and a move towards using the Apple as a learning tool across the curriculum and not limited only to use in maths or computer studies. We are, I believe, at the beginning of a very exciting time in education because the microcomputer as a teaching aid lends itself ideally to long-awaited curriculum development.

The majority of teachers remain sceptical as to the value of a microcomputer in the classroom, not surprisingly in view of the poor quality software on which they are expected to make a judgement. As quality software becomes available, however, more and more teachers will be saying, "*There* is something I can use", and, "*There* is something that will allow me to teach that particular aspect of my subject better than I have been able to in the past".

The same piece of software, because of the philosophy adopted by the programmer, may have a high degree of relevance for some teachers, while others may dismiss a program as unusable as far as they are concerned. This attitude is fine. I would not expect a teacher to use a computer program in a teaching situation if he or she felt that the pupil's learning ability was not enhanced by the introduction of a microcomputer. The decision finally will rest with individual subject teachers as to whether or not they should use a computer program, and not with the DES, MEP, LEA or anyone else.

I would like to dispel the popular myth that computers will replace teachers in schools. This is, of course, nonsense because effectively the microcomputer will enable teachers to be better at the job of teaching, which ought to be an objective which all teachers set for themselves,

What form do we expect CAL software to assume? As I see it there are two main areas of development. Drill and practice type programs which are interactive and

→

essentially text-based are ideal for reinforcing learning. They are used normally on a one-to-one basis, pupil-to-computer, and will help weaker children to keep pace with the rest of the class, and for brighter children to improve on their individual knowledge and performance in an area of a subject where they may need extra assistance. The microcomputer has an increasingly important role to play in this respect, since most teachers have a responsibility to a class of pupils and often the simple time factor means that individual attention is not given to all those pupils who may need it.

The second important area of CAL development is that of classroom demonstration, using a microcomputer in areas where traditional methods of explanation have proved difficult. I believe that the best of these programs will have a large high-resolution graphics content, making use of a picture, diagram simulation or model which is interactive, and as the conditions are changed so the computer model instantaneously reacts. Specific examples may be open to debate but as a typical illustration of the point I will refer to a CAL software package I reviewed recently on the Ideal Gas Laws. The inter-relationships between pressure, temperature and volume are complex and therefore difficult to explain to the uninitiated.

This program allowed the user, normally a teacher, to individually or simultaneously vary these parameters and the effect was immediately translated onto the simulated model on the screen. Questions such as: "What happens to the pressure value if the temperature is increased by 10°C?" are instantly answered and the simulation reveals a wealth of information, in an easily understandable form, and gives rise to spontaneous questions from and discussions among pupils. The computer brings another dimension to the teaching situation and pupils appear to accept information from a computer presented in this way far easier than if the teacher only was presenting the information verbally.

Apple users should rejoice that Apple Pilot is available for them to write CAL material for use in ways like this, since programs can be created without prior knowledge of a particular programming language. The true potential of authoring languages like Pilot is yet to be realised in schools, but as more teachers become aware of it so more CAL software will be produced and teachers from a wide range of subject areas will gradually appreciate the valuable role the Apple will be able to play as a teaching aid.

David King

● *David M. King, BSc (Hons), Cert Ed (Loughborough), is manager for educational and scientific services with Microsense Computers (Apple UK).*

COMPUTER games for people with physical impairments can be much more than mere entertainment. Bill James, designer of the Alpha Menu, feels that the quality of the software he provides is probably more important than the elegance of the hardware.

"Functional as a given combination computer and apparatus may be, if it is not fun for the user it will not be used," says James. "Programs with seemingly no utility whatsoever may be vastly more rehabilitative than the most ingenious utlitarian programs."

Applying his own philosophy James has come up with a series of computer games which can be easily modified to run at various levels of difficulty. Children or newly disabled adults can play with these programs until they have mastered the use of a joystick, switch, or other input device. Once they can play the games, they are ready for the hard work of using the computer for communication and environment control.

Judy McDonald, a communication disorders specialist in Seattle, Washington, has reached the same conclusion in her work at the Maplewood Handicapped Children's Centre.

"You and I prepared for talking by babbling," she explains. "These kids need a chance to play with the movements they have, before being asked to use them in the very specific and demanding task of communicating."

Maplewood's programmer, Paul Schwejda, has developed a series of motor training games guaranteed to be fun for children with or without physical handicaps. At Maplewood, these games have the double job of training the children to use the computers and of allowing the therapists to assess the most natural movements available to each child.

McDonald emphasises this assessment process. "The tricky thing about providing a child with a communication device is that you have to find the most appropriate motor movement the child can use.

"This is of concern for two reasons. First, you don't want the child to be frustrated because the movement is too difficult or too inefficient; and second, the extended use of an abnormal reflex pattern may actually lead to physical deformity."

At Maplewood physical and occupational therapists are an integral part of the assessment process to make sure that the movement reflected is the optimal one for the individual child. Once the children have demonstrated motor control with games, they begin using specialised programs in communication and academics, also designed in the Maplewood project.

Another challenge to providing a child with a communication device is finding the most flexible system that the child can understand and use. At the age of five an average child has a working vocabulary of about 5,000 words, well beyond the limits of any mechanical devices a five-year-old can operate.

And communication systems adequate for a five-year-old will be too limited – and probably too small – for an eight-year-old.

"It's hard to convince an insurance company that the device they paid for three years ago is no longer doing the job," points out Chris Thompson at the Trace Centre for the Severely Communi-

First grab their attention – then rehabilitate

catively Handicapped.

"After all the years we've had to use specialised equipment that has to be replaced periodically, we are very pleased to see the microcomputer come along."

Thompson says she sees the Apple as very useful as an assessment and training tool for children, whose communication needs change almost daily.

Thompson is working with Gregg Vanderheiden, who envisions some basic improvements in technology which he thinks could revolutionise efforts in this area.

Vanderheiden explains his goals with one of his many metaphors.

"We've built the equivalent of go-carts for these kids now", he said. "A go-cart is mobility, and it's a big improvement over nothing.

"But – at least here in Wisconsin – nine months out of the year a go-cart doesn't do you much good. What we have to do is get these kids into cars if we expect them to move with the rest of us."

Vanderheiden says that two improvements are imperative before personal computer systems for the handicapped can begin to fulfil their potential: the systems must be able to run standard software unmodified, and they must be able to perform many functions at once.

All of the communication programs now available translate specialised user input into standard characters which the computer can process. By including a translation system in the software a handicapped person can program an Apple to keep the accounts, play games, write novels, or anything else.

But since the computer can run only one program at a time, a user without keyboard ability cannot run standard, off-the-shelf software.

"If you have to write custom programs for each application, you're going to be very limited, because it takes a lot of time to write good software," Vanderheiden points out. The Trace Centre is now compiling a registry of programs for the handicapped.

Existing systems are also limited by their inability to perform more than one function at a time. "What if you're sitting there writing a letter, and the phone rings?" Vanderheiden asks.

"If we're going to make any real impact on people's lives we're absolutely going to have solve these problems," he insists. "We can't afford to stop and solve them for any one client, but once we find the solutions it will make things considerably easier and cheaper for everybody."

Primary target

WRITTEN by a primary school headmaster, a suite of five programs is being offered by Nicomtech Ltd. of Plymouth. While aimed at the primary school age group, older pupils needing remedial help may benefit using this type of program, with up to six children taking part at any one time.

The first four programs on the disc deal with spelling, word recognition and the building of words, while the fifth is an arithmetic program illustrating the four main mathematical functions in varying complexities. The programs are all written in a game/reward format with happy/sad faces, moving objects and targets to fire at as rewards.

Racing car: This program gives children practice in spelling. The technique used is a modification of "Look, Cover, Write." The child is presented with a word which they look at for a time, then they have to write the word.

Heads and Tails: The recognition of initial and terminal sounds is practised here. Three letter words are used, and there is a facility for deciding which sound is to be worked on. The program allows words to be either loaded in from disc or typed in by the teacher prior to running the program.

Two Faces: With this program the children can have a bit of fun while learning something about the structure of words, and devising simple strategies. The program is a non-violent form of Hangman, where instead of hanging a person the child tries to build up a happy face before building up a sad one.

Spell Invaders: This is an enjoyable way to practice spelling. It is very loosely based on the Space Invaders type of game. The object of the game is to shoot down the correct letters for a displayed word. Up to six children can play at any one time.

Close Encounters: The aim of this program is to give children practice in arithmetic using a game situation. Up to six children can play at any one time, the program calling each child in turn to have their go. It will handle all four operations in arithmetic, at one of five levels. These levels are chosen by the teacher to suit the particular children operating the machine.

The hardware you need to run these programs is a 48k Apple with colour – although they will run perfectly well in black and white – a disc drive and game paddles. Price of the software is £19 and can be obtained from Nicomtech Ltd., 5 Windsor Villas, Lockyer Street, Plymouth, Devon.

Applecart

A voice for the dumb

JAY Hewitt at the University of Missouri has come up with an Apple-based communications system that actually speaks out loud. With this equipment, a person with speech difficulties can say up to 20 words at a time, drawing on a stored vocabulary of 5,000 words.

To make the program accessible to people with little motor control Hewitt has arranged the word lists in nested menus. Typing two letters into the computer brings up a list of all the words in the system starting with those letters.

The user then needs to type only the number of the desired word. This minimises the required typing and allows children to use words they can recognise but can't spell. The program will accept switch or joystick control, but works more slowly with them. Hewitt encourages his clients to use the keyboard directly if they can.

A "key guard", a plastic frame with finger holes over the keys, gives many people the control they need to hit only one key at a time.

Hewitt's system uses an Apple II or Apple II Plus with monitor. But instead of the standard floppy discs and disc drives it requires a Corvus hard disc and drive. This accessory costs about three times as much as the computer itself, and makes the system more economically feasible for institutions where many people can use it.

A shape for the word

LEARNING how to link the visual shape of a word with the sound it makes is vitally important for dyslexia children, just as it is to associate the pattern of the writing with the meaning it conveys. A computer is an ideal tool to teach shapes, sounds, pattern and meaning, as it can provide instantaneous reinforcement through three senses – vision, sound and touch.

An Apple II is now in active use at the headquarters of the Dyslexia Institute in Staines, Middlesex, together with a Mountain Hardware Supertalker providing the essential sound link. Both have been donated to the institute by Microsense.

New pupils there have to go through a detailed assessment procedure so that psychologists can discover the exact nature of the child's dyslexia before working out an appropriate teaching programme. Assessment is painstaking and repetitive and, until now, very time-consuming. But with the help of the Apple it can now be carried out more efficiently and quicker.

The Apple is also being used to store and collate data on the 2,000 dyslexics assessed each year, which should prove invaluable in analysing the extent of dyslexia difficulties.

The equipment was handed over to Dr Harry Chasty, the institute's director of studies, by David King, the manager for educational service at Microsense.

He said: "The teaching procedures which have been found to be most efficient for dyslexics demand that the teacher builds on organisational structure in language which the dyslexic pupil is unable to create alone.

"The meticulous control and planned extension required to develop this structure to meet each individual student need can be readily achieved on the Apple."

Apple Crumble

LECTURING – or perhaps I ought to come clean – guiding students over the rocky path of an 'O' level computer studies course has enabled me to understand some of the problems associated with introducing young people to computers.

When a large contingent of the class would rather be indulging in the gentle art of kicking each other on the football pitch or visiting the usual cycle shed rendezvous, what chance do the intimicies of Charles Babbage and Lady Lovelace's relationship to computer development have in holding their attention.

What a difficult subject it is to teach when a mountain of jargon surrounds the hardware and 'softoys'. How can one cope if the pupils think that a high level language is one which is spoken by an extra-terrestrial being, or that a digital computer is when you "work it out on your fingers". How do we proceed up the prickly path of computer literacy?

Computer appreciation classes are a step in the right direction, making the school computer facility readily available to pupils in free periods. Many schools already implement ideas such as these despite difficulties in staffing and time-tabling. CSE and 'O' level computer studies are excellent courses for these students who opt to take them, despite the magical look back into computing history which many boards continue to make compulsory.

But what of little Joe Average who could easily go through school never coming face to face with a computer of any type? By 1982 every school will, might or should have a micro, thanks to the marvellous Government scheme, which we all support – don't we? But what will the schools do with them in terms of computer literacy?

I believe that the answer lies with the wealth of well-documented, reasonably-priced software that will enable teachers to show young people what the computer can do, rather than how it does it.

Bob Senior

Personal computing at South Eastern Gas

By DAVE WILCOX, Chief management accountant

IN the beginning we had mainframes. We were used to making considerable use of computers, both with batch systems (financial accounting) and timesharing (management accounting and planning). We use IBM 3032 and Amdahl 7B in-house, and Sigma 9 at a bureau.

So why the interest in micro-computers?

Most of our experience with computers was through expert intermediaries such as analysts and programmers. These people had developed expertise in creating large-scale applications systems that would run very efficiently. But they had also developed a heavy workload and an equally heavy backlog. This meant that small-scale applications were either scheduled 'next year' or were not even in the schedule at all.

But we need the system today!

We therefore decided, about two years ago, to investigate the possibility of using micros as a means of solving these small but immediate problems. Two of us who had a technical background set about reviewing the available hardware and its applicability to our requirements. After three months of part-time work we had identified some potential applications and also short-listed some machines.

The next step was to seek professional advice from a recognised micro consultant. This proved invaluable, not only in identifying the particular machine most suited to us but also in establishing a useful contact for later.

Armed with the necessary information and a list of potential applications we then had to present a case to our director, justifying the expenditure on a machine. This justification was partly on an R&D basis since we could not *prove* that the purchase was financially justifiable. Nevertheless, we went ahead.

Now that we had purchased our first Apple we had to make sure that it was used – but none of us had any experience. So where to start?

The first job was to take one of the potential applications and show that this was feasible. The two of us developed the system while, at the same time, learning Basic.

Although we ran into a few minor problems in learning how to make 'best' use of the micro and Basic we soon overcame them. Within a couple of weeks we had a small system that worked and we were so confident that we offered to demonstrate it to the director. And it worked!

Looking back, I have to admit that this is far from being an elegant suite of

> ❛Within five and a half months of buying our first Apple the savings we made on bureau expenditure were enough to cover our capital expenditure❜

programs – but does it matter? We have a solution that is still being used on a regular basis, even though it may take an extra couple of seconds over a more efficient suite. Who cares once a month?

The two of us very quickly built up solutions to most of the potential applications and rapidly learnt ways of becoming relatively efficient in our programming.

Initially the micro had been located in my office for reasons of security and ease of access (for me!). I had, however, made it clear to my staff that they were quite welcome to use it during the lunch-time for playing games (or even work). This had the effect of encouraging them all to feel at home using the computer.

Fairly soon one or two of them began asking about more serious use and about writing some programs themselves. Within five months of obtaining the micro the demand for its use had reached the level where it was necessary to move it out of my office in order that I could get some peace and quiet.

It is worth noting that within five and a half months of purchase the savings we made on bureau expenditure were enough to cover our capital expenditure. We were now in the position of having several staff actively interested in developing knowledge and skills in the use of micros. The next step was therefore to capitalise on this and send some of them on a training course. But which one?

I spent many frustrating hours reading course 'blurb'. Most of the sales literature seemed to offer only two options, either a one-day or two-day 'executive overview' or a one-week or two-week 'programming course'.

Neither of these two approaches was what I was looking for, so I went back to our original consultant and between us we developed a three-day course which appeared to satisfy our requirements.

It is worth noting that the course contents were developed using micros. Rather than writing letters to each other we used a word processing package and communicated using floppy discs. This obviously saved a great deal of time and also helped to increase flexibility in approach.

The course covered the major areas in which we were interested and lasted three days (residential), of which about two thirds was hands-on experience. Included in the agenda were:

- What is a micro?
- Some useful packages.
- Other people's applications.
- Programming in Basic.

We sent ten accountants, from all branches of Finance, on this course and upon their return they were asked to identify any potential applications that they considered we could develop ourselves. This resulted in a list of 54 unique applications, plus some duplication.

Three of those items have since saved more than the total cost of the course, which I would suggest is enough justification for anyone! (Incidentally, about 40 of these applications are now live.)

We now had a number of accountants who were capable of making good use of micros, but how many would actually follow this up?

As with all types of training some of the knowledge fell on stony ground and has not, so far, been used. Some was absorbed by staff who do not feel inclined to write programs but can now readily recognise potential applications. And yet others fell on fertile soil and the applications have grown and flourished.

> ❛Decisions involving considerable sums of money are now being taken on a much more objective basis than would have been possible previously❜

Indeed, we have found several others who, perhaps through jealousy, have taught themselves in order not to lose out on the availability of this useful management tool.

The use of the original machine had grown to such an extent that within a year of its purchase I was getting complaints from my staff that they 'could not get on it' when they had some work to do. We took some samples of the level of usage and found that, on average, it was being used for 45 hours a week!

We therefore thought about obtaining another micro. Being in Finance we obviously had to observe the same disciplines that we impose on others, so a proper financial justification was duly prepared. This led to us purchasing two further machines.

Recent sampling of usage has indicated an average operation of over 40 hours a week on each micro. So where do we go from here?

We are in the position of having over a dozen accountants who are quite capable of developing their own application software. In addition, we have many others, including clerical staff, who are quite at home using micros.

I wonder whether they would have been so happy using mainframe systems, since there is still a degree of uneasiness in using a 'big computer'.

We are beginning to realise many benefits from micros, not only in getting faster processing, but also in being able to provide functional managers with better planning information. There have been several instances lately where we have been able to provide them with information that would previously have been inordinately time-consuming and expensive to produce.

This has resulted in decisions, involving considerable sums of money, being taken on a much more objective basis than would have been possible previously.

The number of users surely indicates a need for a 'proper' time-sharing system. Should we not consider expenditure on a system such as this, rather than on more micros? (We already have an in-house time-sharing service.)

This is not as simple a question as it may appear. Some of the many advantages of the micro are its availability, portability and its ease of use.

☐ A considerable body of micro knowledge has been developed. Should this be thrown away?

☐ The micro is 'user friendly'. Would a TSS be similarly so?

☐ A dedicated micro can provide a very fast response time. Would a multi-user system provide similar response times when several users are logged-in?

☐ I can take a micro home at the weekend. Similar flexibility would require acoustic couplers. And what about my phone bill?

☐ The micros need very little support, indeed most of the support that is provided is done by other micro-users on a mutual basis. A TSS would require a significant level of support on both software and hardware/network.

☐ In the event of a micro hardware failure there are other compatible machines available. If an in-house TSS fails, then generally everyone is without the service.

I cannot say what will happen. However, as a user I am happy with the availability and flexibility of micros for small-scale applications. If mainframes can improve on this (cost-effectively) then I will use them in preference.

But remember that the computer should provide the service that is required by the user. After all, your departments are known as computer services.

☐ *This article is based on material presented at the Xephon seminar on "Personal Computing for IBM Users" held in London recently.*

● *In next month's Windfall Dave Wilcox describes a case study of how South Eastern Gas used their Apple in a successful marketing exercise.*

Hot line service with Young ideas

ALTHOUGH Systematics International Microsystems is headed by two men called Young, it is by no means a newcomer to the micro scene. Ron and Mike Young, who are not related and come from entirely different backgrounds, have been building up the company into one of the foremost suppliers of commercial packages on the Apple for a number of years.

Ron is the founder of the Systematics International Group of companies. At the end of the 1960s he became one of the first of the computer professionals to foresee the tremendous impact that minicomputers would have in the commercial market, and set about establishing a highly successful computer organisation running a full scale bureau based on Prime and IBM minicomputers.

In the late 1970s, when it became evident that microcomputers were going to play an equally important role in the growth of data processing, he steered his company into the new field, using the experience and resources already gained in the minicomputer market.

Heading this new group, the Systematics International Microcomputer Division, is the marketing manager, Mike Young. He has been in the computer industry for 13 years and has seen the complete revolution from mechanical accounting machines to the minicomputer, and now to the microcomputer.

Mike's first contact with computers was as group advertisement manager for two of the industry's leading publications, Computer Management and Computer Digest. Moving to the States in the mid 70s, where he could witness the birth of the micro industry, he organised the first ever microcomputer exhibition in Boston and became East Coast sales manager for the Logical Machine Corporation.

Returning to the UK he established his own computer marketing consultancy, which included the introduction and marketing of a range of US 16bit microcomputers in the UK and Europe.

Mike and Ron Young eventually met early in 1980. By this time Ron's company had been successfully established, providing international market exchanges for a number of prominent blue-chip companies in automotive and other industries, together with bureau services for local businesses. The company was also selling both Apple and ITT systems, and as a response to dealers asking for commercial software systems, set about developing a range of small business applications.

Mike Young (left) and Ron Young with their new Financial Controller software

As Mike says: "That's where the user was facing real problems. Many of the standard packages available were pretty badly written, and in some cases the documentation was appalling." The familiar cry known to us all, as software often lags two years behind hardware development.

It was obvious that a company developing a logical range of well-written software was desperately required by the microcomputer market, to cover the needs of the smaller commercial user.

Mike Young joined the company at this stage. He found that SI was ideally placed to develop such systems because of its long experience in writing commercial programs.

Together they formed SI Microsystems to develop and market a range of modular business systems based on the Apple, supplying either stand-alone modules for individual tasks or a completely integrated system.

Pascal was chosen as the ideal language in which to write the systems, rather than Basic, because it is easy to use and rather more efficient with its use of memory and disc space.

The three systems which were developed using these principles – the Financial Controller, the Stock Controller and the Administation Controller – are now firmly established as among the leading and most professional packages available for the Apple computer.

They cover all facets of business accounting, and will shortly include 80 column word processing, addressing, mailing, time recording and a data base. The documentation supporting the packages has been carefully written to simplify the application of the systems for new users, and SI have provided a hot-line service so that customers with problems can get a fast response to enquiries.

SI also provide extensive dealer and end-user training courses at their own training centre in Suffolk. They also run residential courses at historic Cleves House, where potential customers can get 'hands on' experience.

It is quite obvious that the main asset that both Mike and Ron Young have been able to bring into Apple computing is the professionalism of a company experienced in providing a complete service to established computer users.

The small businessman is not so much impressed by the technical wizardry of the computer system he buys as by the company supporting the financial gamble he generally thinks he is taking by becoming computerised.