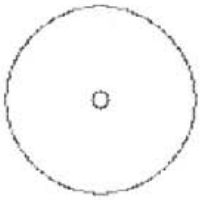
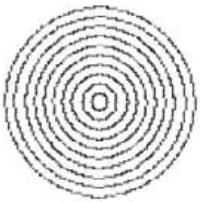
hard core

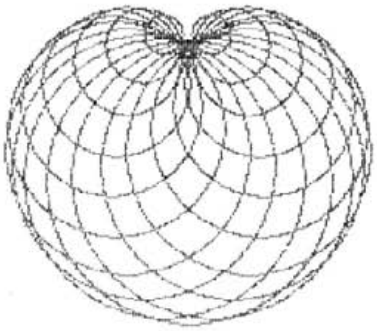
THE JOURNAL OF THE BRITISH APPLE SYSTEMS USER GROUP

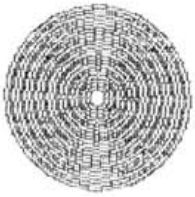
October 1983 £1 VOLUME 3 No. 5



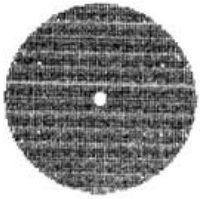


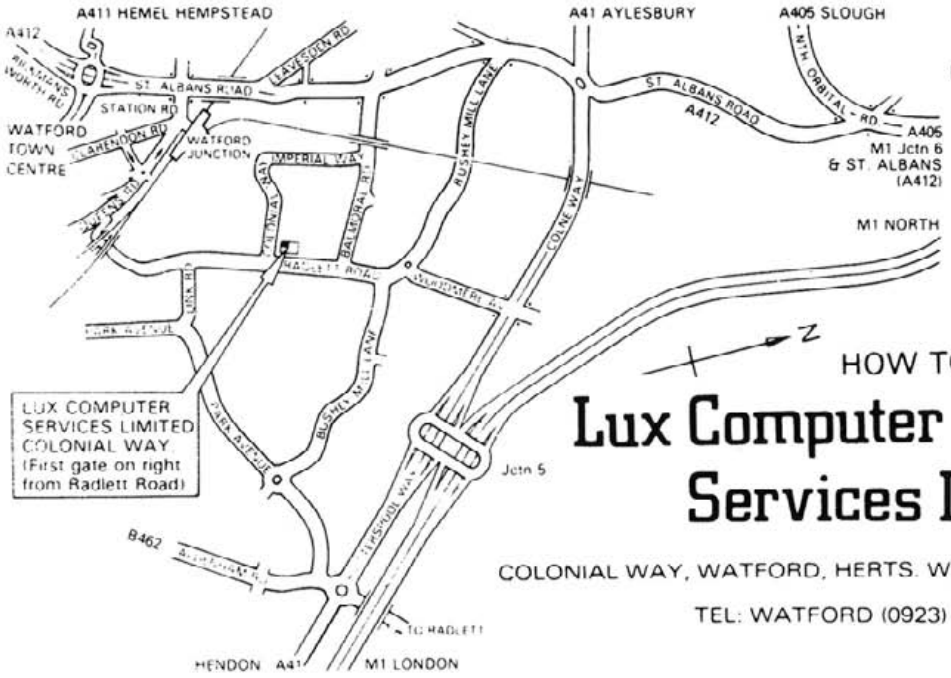
Fast circles





for APPLE/ITT 2020





We've moved!

HOW TO FIND

Lux Computer Services Ltd

COLONIAL WAY, WATFORD, HERTS. WD2 4A

TEL: WATFORD (0923) 473

BERKSHIRE MICROS LTD

We specialise in second-hand APPLE computers and accessories. We usually have a choice of APPLE II Europlus computers in stock.

Whether you want a complete business system or a RAM chip ring us for a quote. We can't guarantee having just what you want but if we do have it you will save money.

Used Equipment Currently in stock

APPLE II 48K EUROPLUS
ITT 2020
APPLE Disc Drives
Selection of Printers
Selection of Apple Cards
APPLE Monitor III
Selection of green screen monitors

We stock the following new equipment:

JUKI Daisywheel printers	365
STAR 510 Dot Matrix printers	260
SHINWA CP-80 Dot Matrix printers	260
APPLE Compatible Disc Drives from	195
APPLEWRITER IIe	110
Discs (Box of 10)	14
Fanfold Paper (2000 Sheets)	12

We also stock the complete range of PEANUT accessories

NEW THIS MONTH

WORDSTAR with the Micropro STARCARD. (Z80B, 6MHz and 64K RAM) All-in price of 285

We will take your APPLE II in part exchange for an APPLE IIe

We can repair your APPLE quickly at competitive prices.

Complete systems available for hire

We pay cash for used APPLE equipment

All prices quoted are exclusive of VAT

BRACKNELL (0344) 84423

AMPLE £1

Amplifier & Volume control for Apple II+ and //e

40/80 Column switches only £8.5 (state which card)

Small run hardware a speciality.

STILL ONLY £65 TO BASUG MEMBERS :-

ZAPPLE

Sound effects & Music Board

ALL PRICES INCLUDE VAT & P/P

Meekrose Ltd
85 Waterdell
Leighton Buzzard
Bedfordshire
LU7 8PL

Telephone: Leighton Buzzard (0525) 370621

MEEKROSE

hardcore

THE JOURNAL OF

THE BRITISH APPLE SYSTEMS USER GROUP

P.O.BOX 174 WATFORD WD2 6NF

BASUG Ltd. is a non-profit-making company limited by guarantee.

Basug Committee

****

Many regular features such as the editorial, chairman's corner, etc. are missing from this issue due to lack of space.

BANKERS

National Westminster Bank PLC, Brompton Square, London SW3 1HH.

CONTENTS

INTERNATIONAL APPLE CORE TM

MEMBER OF THE INTERNATIONAL APPLE CORE

HARDCORE is produced using Applewriter I, and printed on an APTEC Flowriter with a Madeleine Daisy-wheel

Front Cover: Design by Ray Harris. See article on Fast Circles in this issue.

The Apple Speaks To The BBC

By Nik KELLY 8 August 1983

Introduction.

This article describes the programs I have developed to link an Apple with a BBC'B via 4800 Baud RS232. The BASIC is simplified by three short machine-code programs, a Call -APPLE 'line input' for reading disk files and my serial 'Inline' and 'Outline' routines.

Request To Send?

The two obstacles to interfacing an Apple are the Disk Operating System (DOS) and the lack of a 'line input' in Applesoft. Apple DOS is involved in every I/O task. All conventional text transfers must pass through its input and output vectors but, sadly, DOS cannot keep separate input and output streams.

If I want to print a text file from the disk, I must use a program like:

```
'N = 0 : REPEAT : (d)OPEN : (d)POSITION, R N
: INPUT LINE A$ : (d)CLOSE : N = N+1 :
(d)PR# printer : PRINT A$: (d)PR#0 : UNTIL
end of data'
```

The reverse trip is little better,

```
'(d)OPEN F$ : (d)CLOSE F$: REPEAT :(d)IN#2 :
INPUT LINE A$ : (d)IN#0 : (d)APPEND F$ :
PRINT A$: (d)CLOSE F$ : UNTIL end of data'
```

The REPEAT UNTIL is usually implemented with 'ON ERR GOTO'. INPUT LINE must use GET, looped until the end of the line, to allow commas, quotes and colons to be transferred: One alternative to the resulting 'whirr then wait' is to put the strings into an array, then APPEND or PRINT it all at once. It takes a lot of RAM, though. A de-tokenised program can expand by a factor of five!

These programs bypass both of those problems. The serial input and output do not pass through DOS, so the IN# and PR# are not needed, and the 'Line Input' is performed by an ampersand utility.

The Hardware.

1) The BBC Micro.

The 'B' has an RS 423 interface with Baud rate and bit pattern selected by software. The 5-pin DOMINO socket carries RS 232 compatible CTS/RTS, Din/Dout and Ground signals, as shown in the BBC Guide. The plugs are reversible, so PLEASE mark the top clearly!

The BBC is a strange machine for an Apple owner. There's a 6502 CPU, ROMs at the top of RAM, memory-mapped I/O and ROM-paging, but the organisation seems weird. It isn't. The simplest way to look at it is to imagine that I've used one of those expensive utilities that relocates DOS onto a second RAM card. Now, I'd have FP, INT and DOS on different 16 K pages, to match my BBC's BASIC, WORDWISE, Forth and DFS 16K chips. The big difference is that the 4K Apple Monitor gets over-layed, while the 12 K BBC equivalent, the OS, is common to all. Other than that, the similarities are remarkable. The Monitor contains the slot and cassette I/O routines, the screen editing, graphics (Lo-res), even sound: the Beep! for ctrl-G. The BBC OS has those, with suitable extensions for the extra hardware, but it also has a Command Line Interpreter to ease the access: DOS grabs every printed line beginning with ctrl-D, while the BBC OS intercepts commands marked by an asterisk.

The serial 'effects' commands are;

```
* FX 2, 1          ( Take input from serial,
                     locks out kbd
      0            ( Restores default
* FX 5, 2          ( Printer is serial
      1            ( default Centronics
* FX 6, 0          ( Allow 'line-feeds'
     10            ( default, blocks lfs
* FX 7, 6          ( talk at 1 150  3  600
                     5 2400 Baud
    8, 6           ( hear at 2 300  4 1200
                     6 4800
* FX 156,20,227    ( 8 data, noPar, 16 = 1,
                     20 = 2 stop bits
```

Low FX calls are quoted from the BBC Guide. FX 156 is from Acorn leaflet BR001-6/83.

2) The Apple.

The Apple has a disk controller in slot 6 and a serial card in slot 2. The software in this card's PROMs is not needed, as the programs go directly to the 6850 ACIA. This uses two locations, CONTROL at $C0A4 = 49316, and DATA at $C0A5 = 49317. Baud rates between 150 and 4800 are selected by a

rotary switch. These addresses are different for other slots and cards, but the programs are easily changed. Modifications for other ACIA chips are welcomed.

3) Cables.

My serial card has a 12 inch ribbon with header-plug and D-25 socket. I made a 'null-modem' adaptor for the BBC, wired CTS to RTS, Din to Dout and vice-versa, with common Signal Ground. If my cabling was based on D-25, I would have used a D-25 null-modem and an in-line adaptor, CTS to CTS, Din to Din etc.

For short distances, the cable can be a piece of 5-way ribbon, but for lengths over a metre please use quad screened core, or similar. As with any signal carrier such as printer or joy-stick cables, route the serial link away from sources of interference.

The ACIA buffer chips protect against open and short-circuits on the RS 232 lines, and are not vulnerable to moderate levels of static. They will NOT survive telephone, current -loop / Teletype or 'mains' voltages - nor will the computers!

The Software.

The software deals with strings. For speed, my serial I/O requires the first variable to be the send/receive string, so

10 A$ = "" : REM I/O VAR.

The & Utility is not limited in this way.

The machine -code occupies approx 160 bytes of the Apple's page $300, with the Line Input utility for disk files, and the serial OUTLINE and INLINE programs. I've kept them separate as you may not need all of them.

First, at $300 for a CALL 768 or BRUN, is the jump to SETUP which sets the serial parameters and establishes the $3F5 jump for the ampersand. This is followed by jumps at 771 and 774 to OUTLINE and INLINE respectively. I didn't like using so many jumps but there is no significant delay and the CALLs are easy to remember.

After SETUP comes D. A. Lingwood's &INPUTLINE utility, from 'Call Apple in depth #1 1981'. When I discovered it recently, I realised that it was too neat to beat. It reads upper and lower-case, punctuation and quotes into the keyboard buffer through the current DOS vectors. When it reaches RETURN or CHR$(13), it points the named string to the data. If you need to save the string, you must use a line such as ' A$ = "" + A$ ' before the buffer is re-used.

Next is OUTLINE. This sends the contents of the I/O string to the serial port. The format is ' A$ = "MESSAGE" : CALL 771 ' Using the ampersand, ' 40 A$ = "" 50 & A$ 60 CALL 771 : GOTO 50 ' can READ a text file fast enough to keep my disk spinning.

Applesoft stores its strings in two parts: VARSP @ $69,6A points to the start of the variables. Strings have two bytes for the name, then a byte for the length, then the address of the text in lo,hi format, which may be anywhere in memory. My program copies the length and lo,hi bytes into page-zero, checks for a null string, then initialises the Y register as the position counter.

For each character, the program calls the READY subroutine. This loops until 'bit 1' of the ACIA CONTROL register goes to 1, showing that the line is free for use. The first character is then copied from the string and put into the DATA register, automatically setting CONTROL 'bit 1' to zero. OUTLINE then loops back until the entire string has gone. Finally, a CHR$(13) RETURN is sent to complete the line.

INLINE is very similar, but the I/O string should be filled with spaces before use to reserve space for the incoming message. Again, it copies the string's lo,hi address into page zero. The Y register is used as the length counter, while the X holds the bit-pattern that will set the ACIA to show BUSY and prevent an over-run. The BUSY is turned off, and a rapid loop checks for a received character. A 1 in 'bit 0' shows there's a catch, when the gate is promptly set to BUSY. The DATA is read, and if it isn't a CHR$(13), it is added to the string text. When the end-line arrives, or the length counter fails-safe at 256, the length is reduced by one and this result is put into the string header.

The Uses.

I hoped the link would save the cost of a disk for the BBC and allow me to treat the BBC as an intelligent graphics display. The serial software took too long to develop. Both machines now have disks and the graphics are · to program on the BBC, but the link

has proved useful. My current word-processor is Wordwise, in the BBC, so that machine has the 'parallel' Epson. The Apple uses a old 40 column Silentype for 'quickie' listings, but anything else goes via the BBC. The BBC's built-in symbolic assembler, though fine for little programs, is no match for LISA on the Apple. The assembly listing in this article was put on disk by LISA 2.2, then sent through the link to Wordwise. I just had to remember that after I typed '*FX 2,1' at the BBC, the Apple needed to send a 'Return' and an 'Escape' to put Wordwise in Insert-mode, then finish with an 'Escape' and '*FX2' to set it free!

The Tale

At the start, using the serial card's software and a 'receive' patch for the BBC's 0.1 OS, the link ran at 300 Baud. The Apple's looped GET statements effectively halved that, which meant that transfers averaged five minutes.

While I waited for the BBC's promised 1.0 OS ROM, I used its 1200 Baud tape socket and tried to improve the 'receive' patch. Both machines had 6850 ACIA chips in their interfaces, but while the Apple's was well behaved, the BBC's would not follow the data sheet. I knew that a ULA chip intercepted some of the CONTROl bits, but I couldn't get any information for several months.

The BBC's 'receive' was stalled, so I turned to the 'send'. The higher Baud rates promptly over-ran the Apple. I checked for a disconnected hand-shake line, discovered that the serial card wasn't providing the signal! I couldn't believe it. I'd paid extra for a versatile serial card, so surely that signal was in its list of options? Yes and no. The hardware was there, but fitting all the clever printer routines into the card's PROMs had squeezed out the dozen bytes needed for a hand-shake loop.

I abandoned IN#2, wrote a mini-assembler program to provide hand-shake and store the received character at the end of the program. The BASIC GET then became 'CALL 768 : A$ = CHR$(PEEK(790))'. The BASIC ran much quicker without those DOS commands! After losing another struggle with the BBC patch, I wrote a prototype INLINE. To my surprise, I had some trouble with handshake above 600 Baud. Several of the standard loops just couldn't keep up, and I spent weeks trying different ways to stop the machine-code doubling or dropping initial characters.

At that stage, I got some information on the BBC chips, and saw how to turn the BUSY on and off quickly. Fitting that to the scarcely documented BBC I/O vectors took some weeks, but I finally sent a large file to and fro at 4800 Baud. That was a Saturday night. On the Tuesday, a packet arrived with Wordwise and a OS 1.2 chip ....

The BBC machine had one final trick to play: The OS 1.2 used different default parameters for the serial. Until I could check the chips, I thought I'd blown a buffer. Even then, I had to wait for Acorn to send the *FX 156 data.

The next step.

The programs can be improved. Using named strings through-out, monitoring several ACIAs and linking three or more machines in a loop come to mind. If the Out of one is connected to the In of the next, and every message is preceded by two bytes of address, then you could specify one, many or all of 16 machines without complex addressing. Just AND the address with your own, which has a single bit set. If not yours, then send the message on. If yours, then clear that bit and send the message on. If both bytes of address come in as zero, then it has been all the way around and you can dump it. To do it properly, of course, the I/O would need interrupt programming.

Recently, I became interested in linking several machines after watching my young cousins playing (?) D&D on a map-board. I've seen many computerised Adventure games, from Cavern to Zork, The Valley to Wizardry and, real-time, Aphsai to Hellfire, but everyone is really a solitaire. When you're knee-deep in zapped whatsits and they're still coming, a map can be a very lonely place. Also, those puzzles and hazards are algorithm-bound. There are few opponents or partners better than the live variety and, provided the game is real-time, the programming could be simplified. For instance, the BASIC ROMs can serve as the map! .. Any ideas?

The programs should soon be available from the software library.

```
0  REM  APPLE DISK TO WORDWISE
1  A$ = "": REM  I/O STRING
10  CALL 768
20  CALL 771
30  INPUT " FILE NAME ";F$
40  D$ =  CHR$ (4)
50  A$ =  CHR$ (27): CALL 771
60  PRINT : PRINT D$"OPEN"F$
70  PRINT D$"READ"F$
80  ONERR  GOTO 120
90  A$ = ""
100  & A$
110  CALL 771: GOTO 100
120  PRINT D$"CLOSE"
125  POKE 216,0
130  A$ =  CHR$ (27): CALL 771
140  A$ = "*FX2": CALL 771
150  REM  FROM *W.
160  REM *FX7,6
170  REM *FX8,6
180  REM *FX156,16,227
190  REM *FX2,1
200  REM  THEN APPLE TAKES OVER


0  REM  BBC TO APPLE DISK
1  A$ = "": REM I/O STRING
10  S$ = "      ": FOR N = 1 TO 51:
    A$ = A$ + S$: NEXT
20  INPUT " TO DISK FILE NAME ";F$
30  D$ =  CHR$ (4)
40  PRINT D$"OPEN"F$
50  PRINT D$"WRITE"F$
60  CALL 774
70  IF A$ < > "ENDFILE" THEN  PRINT
    A$: GOTO 60
80  PRINT D$"CLOSE"F$
90  PRINT "DONE !"
100  REM TEST
```

The Assembly listing.

```
0800                1  ;
0800                2  ;
0800                3  ;***  LINE I/O  ***
0800                4  ;
0800                5  ;     NIK  KELLY
0800                6  ;   6 AUGUST '83
0800                7  ;
0800                8  ;
0800                9  ;
0300               10  CALL768  ORG $0300
0300               11          OBJ $1000
0300               12  ;
0300               13  ;
03F5               14  AMPERS    EQU $03F5
C0A4               15  ACIAC     EQU $C0A4
C0A5               16  ACIAD     EQU ACIAC+1
0003               17  RESET     EPZ $03
0015               18  SERON     EPZ $15
0055               19  SEROFF    EPZ $55
0300               20  ; 8 DATA NOPAR 1 STOP
0069               21  VARSP     EPZ $69
00FA               22  ZP        EPZ $FA
0300               23  ; $FA - $FC USED
0002               24  MASK      EPZ $02
000D               25  EOL       EPZ $0D
00FD               26  ZP2       EPZ $FD
0300               27  ; USES $FD - $FE
0300               28  ;
0300               29  ;
0300               30  ;
0300               31  ;**  SETUP JUMPS  **
0300               32  ;
0300 4C0903        33  BRUN      JMP SETUP
0303 4C3D03        34  OUT       JMP OUTLIN
0306 4C6F03        35  IN        JMP INLINE
0309               36  ;
0309 A94C          37  SETUP     LDA #$4C
030B 8DF503        38            STA AMPERS
030E A923          39            LDA #ENTRY
0310 8DF603        40            STA AMPERS+1
0313 A903          41            LDA /ENTRY
0315 8DF703        42            STA AMPERS+2
0318 A903          43            LDA #RESET
031A 8DA4C0        44            STA ACIAC
031D A955          45            LDA #SEROFF
031F 8DA4C0        46            STA ACIAC
0322 60            47            RTS
0323               48  ;
0323               49  ;  END OF SETUP
0323               50  ;
0323               51  ;
0323               52  ;
0323               53  ;
0323               54  ;***   & INPUT  ***
0323               55  ;
0323               56  ;(C)  D A LINGWOOD
0323               57  ; CALL-APPLE 1981
0323               58  ;IN DEPTH #1  P126
0323               59  ; = INPUTLINE A$
0323               60  ;
0323               61  ;
0323 20E3DF        62  ENTRY     JSR $DFE3
0326 A200          63            LDX #$00
0328 2075FD        64            JSR $FD75
032B A000          65            LDY #$00
032D 8A            66            TXA
032E 9183          67            STA ($83),Y
0330 C8            68            INY
0331 A900          69            LDA #$00
0333 9183          70            STA ($83),Y
```

```
0335 C8          71              INY
0336 A902        72              LDA #$02
0338 9183        73              STA ($83),Y
033A 4C39D5      74  EXIT1       JMP $D539
033D             75  ;
033D             76  ; END OF  & INPUT
033D             77  ;
033D             78  ;
033D             79  ;
033D             80  ;
033D             81  ;
033D             82  ;
033D             83  ;***  OUTLINE  ***
033D             84  ;
033D             85  ;     NIK  KELLY
033D             86  ;    30 JULY 1983
033D             87  ;
033D             88  ;0 A$="":REM 1ST VAR
033D             89  ;
033D             90  ;
033D             91  ;
033D A002        92  OUTLIN    LDY #$02
033F             93  ; COPY A$ LEN,LO,HI
033F B169        94  LOOP1     LDA (VARSP),Y
0341 99F800      95              STA ZP-2,Y
0344 C8          96              INY
0345 C005        97              CPY #$05
0347 D0F6        98              BNE LOOP1
0349             99  ;
0349            100  ; CATCH NULL STRING
0349 A5FA       101              LDA ZP
034B F011       102              BEQ SENDCR
034D            103  ;
034D            104  ; WORK THROUGH STRING
034D C6FA       105              DEC ZP
034F A0FF       106              LDY #$FF
0351 C8         107  LOOP2      INY
0352 206703     108              JSR READY
0355 B1FB       109              LDA (ZP+1),Y
0357 8DA5C0     110              STA ACIAD
035A C4FA       111              CPY ZP
035C D0F3       112              BNE LOOP2
035E            113  ;
035E            114  ; FINISH WITH EOL
035E 206703     115  SENDCR     JSR READY
0361 A90D       116              LDA #EOL
0363 8DA5C0     117              STA ACIAD
0366 60         118              RTS
0367            119  ;
0367            120  ; CHECK 6850 ACIA
0367 A902       121  READY      LDA #MASK
0369 2CA4C0     122  LOOP3      BIT ACIAC
036C F0FB       123              BEQ LOOP3
036E 60         124              RTS
036F            125  ;
036F            126  ;  END OF  OUTLINE
036F            127  ;
036F            128  ;
036F            129  ;
036F            130  ;
036F            131  ;
036F            132  ;
036F            133  ;***  INPUTLINE  ***
036F            134  ;      NIK  KELLY
036F            135  ;     30 JULY 1983
036F            136  ;
036F            137  ; A$ IS 1ST VAR
036F            138  ; BUT LEN() = 255
036F            139  ;
036F            140  ;
036F            141  ; COPY A$ LO,HI
036F A003       142  INLINE    LDY #$03
0371 B169       143              LDA (VARSP),Y
0373 85FD       144              STA ZP2
0375 C8         145              INY
0376 B169       146              LDA (VARSP),Y
0378 85FE       147              STA ZP2+1
037A            148  ;
037A            149  ; SET FOR QUICK ON/OFF
037A A000       150              LDY #$00
037C A255       151              LDX #SEROFF
037E            152  ;
037E A915       153  LOOP4      LDA #SERON
0380 8DA4C0     154              STA ACIAC
0383            155  ;
0383            156  ; CHECK FOR DATA
0383 A901       157              LDA #$01
0385 2CA4C0     158  LOOP5      BIT ACIAC
0388 F0FB       159              BEQ LOOP5
038A            160  ;
038A            161  ; ADD IT TO STRING
038A 8EA4C0     162              STX ACIAC
038D ADA5C0     163              LDA ACIAD
0390 C90D       164              CMP #EOL
0392 F005       165              BEQ EXIT
0394 91FD       166              STA (ZP2),Y
0396 C8         167              INY
0397 D0E5       168              BNE LOOP4
0399            169  ;
0399            170  ; STORE NEW LENGTH
0399 88         171  EXIT       DEY
039A 98         172              TYA
039B A002       173              LDY #$02
039D 9169       174              STA (VARSP),Y
039F 60         175              RTS
03A0            176  ;
03A0            177  ;  END OF  INPUTLINE
03A0            178  ;
03A0            179  ;
                180  END        END

***** END OF ASSEMBLY
```

Nik says he is prepared to tackle enquiries concerning the article, but please enclose an SAE of a sensible size. He is not equipped to discuss Pascal, Z-80, 6809, etc., as he is a 6502-er until the 68000 add-on for the BBC comes along.

Write to:

Nik Kelly B.Sc. AFBIS
[illegible]

Apple Magazines

by Keith Chamberlain

Several magazines on the market are dedicated to the Apple the best of which is NIBBLE, which is only available from the states. Next comes CALL A.P.P.L.E. which has been obtainable in the UK. and finally WINDFALL.

Other magazines such as PRACTICAL COMPUTING, PERSONAL COMPUTER WORLD, QUESTIONS AND ANSWERS, COMPUTER AND VIDEO GAMES and a host of others sometimes carry interesting articles and programs for the Apple, but should be viewed at your local newsagents before purchase as they may not be of much benefit at times.

At the top of my list is NIBBLE which has an amazing number of articles, reviews and good programs that actually work. The main problem is getting your copy from the states and the price. Also available are NIBBLE EXPRESS Vols 1, 2 and 3 which are splendid value and contain selected articles from all NIBBLES of 1980-81-82 which include all kinds of programs including utilities and games.

Next comes CALL A.P.P.L.E. the american apple user groups magazine which contains hints and tips, programs, reviews and for A.P.P.L.E. members, special offers on software and hardware, although not all of these are available in the UK. CALL A.P.P.L.E. can be received by subscription direct by becoming a member. It has been available in the UK but the company selling it has just folded up so its future availability is uncertain.

There are also special editions called Call Apple in Depth, of which at the moment there are three :-

1. All about Applesoft

2. All about Pascal

3. All about DOS

These are approximately £12 each and the editions I have are 1 and 3, which I have found to be extremely helpful especially number 1 whose cost was covered by one program alone.

Finally we have WINDFALL which, although it is apple dedicated, has its ups and downs as far as value is concerned, but this apart it does have some quite good hardware and software reviews, interesting articles and the occasional piece of information that makes its purchase worthwhile. This is another magazine that has started to appear in newsagents and may be worth thumbing through before purchase.

A new quarterly magazine called ORCHARD comes on the market in autumn which has exclusive rights from Nibble to market their editorial and software in the U.K. In other words they will be printing Nibble articles and programs. ORCHARD will be available from newsagents.

Basug Telephone

Your BASUG telephone contact is Fran Teo and you can phone her on weekdays and on weekday evenings until 8 p.m. on:
St. Albans [illegible]

Mainframe Interview

The idea of using Apple computers in the field of music isn't a new one. Most of you will have heard of Mountain Computer Music System, Alf boards and even Zapples! It is therefore surprising that there have been very few, if any, successful pop groups that have featured the Apple in any way. Having seen the Show Me Show on BBC 1 Wed 14th September, which often features computer items, I have a feeling that this situation may change soon. For those of you who missed it, the programme included a song by a band called Mainframe. This song was accompanied by the usual video mixing effects etc., also by what appeared to be Apple graphics. This would have been of interest anyway, but what caught my attention was that the keyboard player in the group was Herts area Basug regular John Molloy!

To acquire more information on this Apple-ication your intrepid reporter went into the depths of Hemel Hempstead, to the home of the other band member, Murray Munro. After a nice coffee we got down to some questions.

Were the graphics for the TV done by the BBC for you?

John:-"No. From the first of our concerts we have done all our own graphics. We had performed live 14 times before the Show Me Show, so by then we had created graphics worth appearing on the BBC. To be more accurate, we either use our own routines or use commercial software in our own way. Some examples of this are Apple World, Ceemac/Fire Organ and Graforth. We've even used the star routine in Cyber Strike!"

With the musical side of the band to concentrate on, how do you find time to do these routines?

John:-"Actually we have considerable help from Colin Holgate and David Green (both Basug members and regular Herts meeting supporters). Colin does most of the graphics for the shows, including the Show Me one and David helps out if Colin is stuck for ideas, also he is involved in special hardware projects we are working on."

More coffee. During this break Colin displayed some of the graphics used by Mainframe. These included many abstract patterns generated by the Fire Organ program (this, and Ceemac are reviewed by John and Colin elsewhere in this issue), sound to light programs, 3D Mainframe logos, ascending bubbles, Mainframe words flying past and many other graphics effects. Some of the sequences appeared to require several hi-res pictures to be stored in memory at once, more than there should be space for.

Colin:-"My original involvement with the band was to help them with some graphics for a video that accompanies the performance of their album Tenants Of The Lattice-Work. We needed to be able to do complex animations, but didn't have the computer speed or film animation facilities that the Tron creator used. An Apple doesn't quite compete with a Cray. This meant that any graphics had to be in real time. Normally this means using simpler graphics, or having 6 months to create arcade speed machine code routines. As we wanted to use 3D effects as well, even machine code wasn't quick enough. Bill Budge's 3D system is quick but doesn't include clipping or perspective, which were vital for our needs. The way round these problems was to have as many complete hi-res pictures in memory as possible and move them into the graphics area in the right order, using page switching to give smooth animation. Unfortunately there is only room for about 4 frames. By using a packing program an average picture could be reduced from the usual 34 sectors to about 6. When the reduced file is loaded and unpacked into the hi-res screen you end up with the original picture. One of our routines shows the Earth, then zooms into a close up of England. These pictures were generated on Bit-Stik, packed, loaded in sequence, and unpacked to give the animation."

The effect is quite good, with slight flicker noticeable. How many frames per second is the animation?

Colin:-"On a normal Apple the unpacking takes about 1/3 of a second, that's the price of saving so much space. Fortunately I have an Accelerator, so the Earth routine

runs at 7.7 frames per second and uses 15 pictures, so it lasts about 2 seconds."

We then adjourned to Murray's bedroom! Seriously. Unlike most of the bedrooms I've seen, this one had four track recording equipment in it. The band do their own recording on their own record label, MC 2 (as in E equals MC squared). I asked Murray if they had used the Apple for generating music at all.

Murray:-"Until the recording of the album, apart from the graphics in our show, we had only used the Apple computer for writing programs that we then put onto our singles. We do use it for storing our fan club names and addresses using Visidex, but that's unlikely to become a Mainframe product. The first single we released had a short Apple text program on it. This was a test really. To see what response we got from computer users, and to make sure that programs could be loaded from normal records. Since we did this two other music people have repeated the idea on their own records. Our second single included four programs on the B side. These were text and sound to light programs for the Apple, BBC, ZX Spectrum and text only for the ZX 81. A lot of our potential listeners have computers other than the Apple, so we try to include as many versions as there is time for. When we commenced the album, we started to use one of the special hardware devices John referred to earlier. This is a percussion machine board that plugs into one of the Apple slots. Dave Green designed it and offered it to us to use on the album. Normally we use a Moog Liberation for our percussion sounds, but when we heard David's board in action the Moog got locked away!"

Murray demonstrated the sounds this board could generate. He said that after 3 finger clicks he would hit various things, and that I should close my eyes and tell him which were real. After the clicks came a tambourine, a snare drum, a clap and a tom tom sound. I could see there wasn't a tom tom in the room, so I confidently said that the tom tom was on the Apple, and possibly the snare drum as well, the others being real. As you may have guessed, not only were all the 'test' sounds from the Apple, the finger clicks were too! Very impressive. I wanted to know more.

Murray:-"There is a good chance that we may produce these boards at a future date, so we're keeping the methods a secret for now."

John:-"The sounds available from the board have changed our percussion sound considerably. It's like having a cheap Fairlight attached to the Apple."

A Fairlight, they tell me, costs about the same price as 20 Apples.

John:-"Someday we may outgrow the graphics ability of the Apple and go onto more powerful machines, but not until we sell a few albums."

Well, must help them out. I'll be down at the record shop if you need me. John and Colin say they'll still be coming to the Herts meetings, between American tours that is!

Small Ads

Sony NTSC Colour Set 13" £100
will work on Apple without colour card.

Paul Tilling. [illegible]

******

Replay Copy Card and software £50

Ring Stuart Morley.
Office Hours [illegible]
After 6.30 [illegible]

****

QUOTE

"If Apple were selling no Apple IIs and no Lisas they would still be the sixth biggest computer selling company in the world."

Steve Holmes
Apple Marketing Manager

More on the 737

by Margaret Wood

Being a relative novice in the world of the byte and having recently had to battle with a newly acquired Centronics 737 printer, I have found the tips gleaned from the back issues of this journal to be invaluable. By a combination of a lot of trial and error and a little inspired guess work, I now have my printer doing things that only a few weeks ago I considered to be quite unattainable. The purpose of this article is to indicate how I solved the problems that I encountered and it includes tips from previous articles, for which I am indebted to the authors.

I have been working with an Apple //e processor but the solutions have been tested on an Apple II Europlus and found to be equally effective with one minor exception which is mentioned below.

My printer is connected through an Apple II Parallel Interface Card and I must say that I found the manual to this product less than helpful for my purpose. Firstly, there was confusion over the correct setting for switch SW6. Centronics 737 was not mentioned as suitable for the OFF position although other Centronics models were. However, the Centronix 737 was on the list. In my ignorance I still do not know if this is a misprint or another model of printer. From my programmed commands, I found it possible to control the printer from either setting. The ON setting was a little simpler and I made more progress with my later problems using this setting so I have opted to keep switch SW6 ON. Hidden deep in the Appendices is the information that CTRL-I;"O" is the code to return display to the screen. I struggled for a long time with CTRL-I;"40N". A tip that I found most useful was to define D$ as CHR$(13) + CHR$(4) and I$ as CHR$(13) + CHR$(9). This saves errors from omitted RETURNs.

The next problem was to obtain an 80 column program listing, hopefully without extraneous symbols. I tried T Tse's Nice List but it proved totally incompatible with my Centronics 737 and, with my limited knowledge, quite unalterable. All attempts at controlling the printer directly through the keyboard failed dismally. I could not break through the 40 column barrier and cryptic comments abounded at either end of the listing. The problem was solved by executing the following textfile with the program in memory:

```
L = 66 - LEN(H$) - LEN(D$)

IF L<1 THEN ? "STRINGS TOO LONG -
RE-ENTER" : END

ES$ = CHR$(13) + CHR$(27)

DD$ = CHR$(13) + CHR$(4)

I$ = CHR$(13) + CHR$(9) : ? DD$;
"PR#1"; I$; "80N"; CHR$(15); ES$;
CHR$(20); ES$; CHR$(14); H$;
CHR$(14); : FOR Z = 1 TO L : ? " ";
: NEXT : ? CHR$(15); D$; CHR$(14);
ES$; CHR$(15); ES$; CHR$(19) : LIST
: ? I$; "O"; DD$; "PR#0" : END
```

The headings are optional and if required should be entered direct from the keyboard before executing the file. H$ was designed to be the name of the program and D$ to be the date but they can be used for any purpose as long as their combined length does not exceed 65 characters. The last line is intentionally long to avoid printing a selection of 1s at either end. Although I have not done it, I am sure that the file could be amended and two further variables input to give a partial listing.

A further improvement to my listings has been achieved by the use of (CTRL-O) at the begining of and (CTRL-N) at the end of any important remark. The underlining obtained is far neater than rows of asterisks and the program is unaffected.

Finally, I wanted to use a Word Processing Program and, already having access to Applewriter I and Applewriter II, wished to avoid further outlay if possible. Applewriter I does run on the //e processor but some of the rubbish that is displayed on the screen makes it almost impossible to check and amend certain types of text. It was therefore not acceptable to me to use this version with Ian Trackman's Go-Between Program. Despite scepticism from some

quarters, I find that Applewriter II works very nicely on a //e processor. The only compromise to be made is acceptance of the fact that it will not recognise my 80 Column Card and so I am stuck with a 40 column screen display, disappointing but not restricting. (Perhaps Father Christmas will have the //e version in his sack!)

Of course my first challenge was to release the 40 column chains. The answer to that was laid before me in this publication so my default constants were amended to contain a Top Line of: /(CTRL-I)132N/ . Even that took a while to achieve. There must be no extra spaces, no quotation marks and CTRL-I must be entered directly from the keyboard; no trying to be clever by entering CHR$(9)!

The next step was to investigate the use of the CTRL-V function. First results were disappointing. I wanted to underline. I could start underlining but it refused to turn off. At this point I hit upon an amendment to the Top Line command that solved the problem, although I have no idea as to why. The addition will work from within the text file but it is more convenient in the Top Line. My Top Line now reads:

/(CTRL-I)132N(CTRL-N)(CTRL-I)
(CTRL-N)(CTRL-I)(CTRL-N)/

No less will suffice but at least it can stay as a permanent default.

Acceptance of the remaining control codes through CTRL-V was easy once I had tumbled to the fact that ESC must be entered twice.

Having mastered the keystrokes required, I compiled a Glossary File with easily recallable symbols for definition. The following listing gives the actual keystrokes to enter:

F(CTRL-V)(CTRL-J)(CTRL-V)
{Full Line Feed Forward}

1(CTRL-V)(ESC)(ESC)(CTRL-A)(CTRL-V)
{1 Dot Space}

2(CTRL-V)(ESC)(ESC)(CTRL-B)(CTRL-V)
{2 Dot Spaces}

3(CTRL-V)(ESC)(ESC)(CTRL-C)(CTRL-V)
{3 Dot Spaces}

4(CTRL-V)(ESC)(ESC)(CTRL-D)(CTRL-V)
{4 Dot Spaces}

5(CTRL-V)(ESC)(ESC)(CTRL-E)(CTRL-V)
{5 Dot Spaces}

6(CTRL-V)(ESC)(ESC)(CTRL-F)(CTRL-V)
{6 Dot Spaces}

R(CTRL-V)(ESC)(ESC)(CTRL-J)(CTRL-V)
{Full Line Feed Reverse}

E(CTRL-V)(ESC)(ESC)(CTRL-N)(CTRL-V)
{Start Elongated Print}

e(CTRL-V)(ESC)(ESC)(CTRL-O)(CTRL-V)
{Stop Elongated Print}

p(CTRL-V)(ESC)(ESC)(CTRL-Q)(CTRL-V)
{Proportional Print}

n(CTRL-V)(ESC)(ESC)(CTRL-S)(CTRL-V)
{Normal Print}

c(CTRL-V)(ESC)(ESC)(CTRL-T)(CTRL-V)
{Condensed Print}

f(CTRL-V)(ESC)(ESC)(CTRL-\)(CTRL-V)
{Half Line Feed Forward}

r(CTRL-V)(ESC)(ESC)(CTRL-^)(CTRL-V)
{Half Line Feed Reverse}

<(CTRL-V)(CTRL-O)(CTRL-V)
{Start Underline}

>(CTRL-V)(CTRL-N)(CTRL-V)
{Stop Underline}

b(CTRL-V)(CTRL-H)(CTRL-V)
{Backspace}

Because of the use of the backslash in the command for Half Line Feed Forward, I cannot suggest a method of using this command with an Apple][Europlus unless the glossary file can be compiled on a //e processor.

There is just one disappointment. Although I can switch between proportional and condensed print from within Applewriter, I cannot alternate between these two types and normal print. Of course I can revert to normal by turning the printer off and then on but to switch to the other two styles I have to use the Printer Set Up routine that I have incorporated in the Hello Program on my DOS disks as a pre-boot style selector.

The use of enlarged print is a little tricky. Backspacing works but is not as I expected. I think that it is only a matter

of patience and simple mathematics to reach some formulae for its effective use. I hope to submit a further article covering the justification problems involved in the use of these two functions.

I am not sure if novices should be writing articles but hopefully this one may save a little midnight oil for someone.

Exec Tip

EXEC UTILITIES

by

Martin Rogers

In my previous article, BE AN APPLE EXECUTIVE, I promised you more applications for the EXEC command. Here is a useful patch which will present you with your program listings a page (or Screen) at a time. Just enter the following code into an EXEC file using your favourite Word Processor or using a program to write the commands to a text file called "PAGE LISTER". Just EXEC PAGE LISTER at any time to install, and subsequent LIST's will pause at the bottom of the screen until you press a key. The facility will remain active in your machine until the next time you re-boot.

```
CALL-151
300: A9 03 A2 0B 85 37 86 36 4C EA 03 48 A5
25 C9 17 F0 04 68 4C F0 FD 98 48 8A 48 2C 00
C0 10 FB 2C 10 C0 20 58 FC 68 AA 68 A8 68 4C
F0 FD
3D0G
HOME:VTAB5:HTAB9:?"Page Lister Installed.":
VTAB12::?"All Listings will now display a
page at a time until you re-boot the System.":
VTAB21:HTAB11:?"An M.C.R. Utility.":VTAB23:
HTAB10:?"Any key to continue."
CALL768
```

Watch this space for more exciting utilities! Even better, write in with any that you have developed which you would like to share with BASUG members.

Education

by Norah Arnold

FOUR NEW BOOKS ON LOGO
AND TURTLE GRAPHICS

Three of these books are about Logo and one is a little different in that it deals with turtle-graphics only. I shall discuss that one first and then deal with the Logo books.

PICTURE THIS TOO!
subtitled An Introduction to Computer Graphics for Kids of all Ages.
Author David Thornburg.
Published by Addison-Wesley, 1982.
Price £9.00.

This book is fairly large, spirally bound, with an attractive cover. On first opening the book one gets the impression of well presented material, clearly set out and containing many black and white illustrations of the computer screen. There are versions of the book for other machines but the version I have used is written for those who have access to an Apple II with the language Super-Pilot, an extended version of Pilot incorporating turtle-graphics.

According to the author the book is aimed at 'kids' of all ages. I think that an intelligent nine year old could cope with working through this book with a bit of help here and there with the vocabulary. Equally well an adult new to Super-Pilot and turtle-graphics would find this a good book to begin with. One thing that children seem to like about the book is its 'chatty' style with a good spattering of Aha! Whoops! Ta-Daa! Wow! and other phrases terminating in exclamation marks. I can imagine some older people might find this style a little irksome.

I think that it is fair to say that this book is written in the immediate mode. By this I mean that it follows the pattern of 1. Let's type this. 2. Here is a picture of the screen. 3. This is what happened and why. 4. Now let's proceed by doing this..... throughout the book. This inevitably means that progress through the actual turtle-graphics material is slow but done in a reasonably detailed manner. Very simple drawings of lines and squares are

dealt with in Chapters Four and Five. In Chapter Six you learn to draw a stick figure of a person and not until Chapter Eleven do you learn to draw curves and circles. The last chapter, number twelve, ends with a simple drawing of a flower.

The content of the book does not actually take you very far into the realms of turtle-graphics but over the distance it does take you it holds your hand all the way. So if you are a whiz-kid just starting on Super-Pilot who wants to be doing marvellous lace-like turtle-graphics designs in a matter of hours then this book may not be for you. On the other hand, if you like a lot of help and encouragement and don't mind going slowly and surely through the material then this book would be a good buy. That is, providing you have Super-Pilot which I rather suspect that not many people have.

LEARNING LOGO ON THE APPLE II
Authors Anne McDougall, Tony Adams and Pauline Adams.
Published by Prentice Hall of Australia, 1982.
Price £11.00.

This book is aimed at people who know little of computing but who wish to use Logo. In view of this it starts with an explanation of how to set up the system and get going and then describes how it intends to deal with the differences between the two versions of Logo for the Apple II, that is, M.I.T. Logo (Terrapin and Krell) on one hand and Apple Logo on the other. When program listings or single procedures are given then the two versions are printed side by side, with M.I.T. Logo on the left and Apple Logo on the right. I found this approach far more satisfactory than the differences being shown only in an Appendix. Clear sub-headings to the chapters help you to find things easily and in general the presentation of the material is good.

Although this book may appeal to some teachers or parents who wish to interest their pupils or offspring in Logo programming, no help is given over the approach which should be used in presenting the material to the younger generation. For instance, in describing the 'total turtle trip' theorem, a table is given of the number of degrees necessary to produce triangles, squares, pentagons, hexagons etc. This, however, is precisely the kind of thing that a child should be encouraged to discover through experiment rather than having the list given as a foregone conclusion. One good point is that exercises giving suggestions for further work are given at appropriate points throughout the book.

The book does cover the basics of Logo well enough to enable a newcomer to the language to carry out their own projects in turtle-graphics including those using recursion. One chapter deals with the arithmetic one can do with Logo, again in a reasonably simple manner. Some of the most interesting material in the book appears in the last three chapters on list processing.

The authors of this book appear to have had sufficient involvement with Logo on the Apple II to know of some of the little pitfalls and problems which can arise. However, one thing that worries me is that most of the points covered by this book are dealt with equally thoroughly in the Logo manuals for Terrapin and Apple Logo. I can say nothing about the Krell manuals as I do not possess them. If you own the Terrapin or Apple manuals then to buy this book would be unnecessary in my opinion. If you are in a group situation where access to manuals is limited then a copy of this book would be extremely useful.

LEARNING WITH LOGO
Author Daniel Watt.
Published by McGraw-Hill, 1983
Price £16.50.

I had been awaiting the publication of this book with interest for some time. Daniel Watt worked with the Logo Group at M.I.T. for five years as an educational researcher and has had a great deal of experience in teaching Logo to adults and children.

It is a large book, approximately 11 x 9 inches and weighing almost two pounds. It has soft covers and is spirally bound. A great deal of thought has gone into the way the material is presented in this book. The book is divided into three parts. Chapters 1 to 6 are for children between the ages of ten and thirteen to read and work through themselves, although personally I would think that intelligent eight to nine year olds could be included here if they were proficient at reading. Chapters 7 to 9 could be worked through by any child who has satisfactorily completed Part 1, who has understood the concepts involved and had time to experiment and practise using them. The

rest of the book, Chapters 10 to 14 is meant for youngsters from 13 or 14 upwards or for children younger than this to work through with an adult to give a hand here and there.

Much effort has gone into making the book attractive to children, for instance the trickier concepts are explained with the help of cartoon characters; the turtle himself, the Logo wizard and Logo's helpers who are shown as robots whose tasks are linked to the shape of their heads. I could imagine most children would be grabbing for their crayons and felt-tips to colour the cartoon characters when they take a break from seriously perusing the content of the book.

The left hand margin of the book is used to display cartoon symbols indicating 'powerful ideas' (ie. important concepts), 'exploration' sections giving ideas for further experimentation with a concept and finally 'helper's hint' sections which are given in smaller print and are directed at any adult, teacher or parent, who is working with the child.

This book is directed mainly at users of the Terrapin and Krell versions of Logo for the Apple. However, pages 289 to 358 are given over to appendices covering several subjects including using the book with Apple or TI Logo.

I do have several criticisms of this book. First, I found it rather floppy and difficult to handle because of its size, weight and soft covers. A child using it would need to have enough table surface next to the computer to support the book as it is far too heavy and floppy for a child to hold. This raises the question as to whether a slightly less extravagant format with smaller symbols and cartoons would have led to a book which was easier to handle and therefore more pleasant to use - a debatable point. Secondly, although no-one can dispute Daniel Watt's high degree of involvement with Logo over many years, I was disappointed that so much of the content, although presented in a highly original way, echoed material already published by other members of the M.I.T. Logo Group. Thirdly, I was sad that the price will put the book out of reach of many primary school teachers in this country, whose capitation allowances might run to one copy being purchased for reference, but would certainly not allow a reasonable number to be bought for use by the children.

Having said all this, if you want your children to enjoy becoming proficient at Logo, you could not do better at the moment than to buy this book and enjoy using it together.

LOGO PROGRAMMING
Author Peter Ross
Published by Addison-Wesley, 1983
Price £7.95.

I had been using my own copy of this book for several weeks before the review copy arrived. It is different from the previous two in that it is 'home grown' and originates from the work of the Logo group in the Department of Artificial Intelligence at the University of Edinburgh. It is also different in that it does not have a picture of a hand placing a disk in a drive. In other words it is not a book for complete beginners to computing and it is not aimed at children. Neither does it cover the same material as the manuals which come with the different versions of Logo. Although Terrapin Logo is used throughout the book, notes are given on other versions where this is necessary. Appendices cover Apple Logo, Research Machines Logo for the 380Z, TI Logo for the TI99/4 and 4a, and Radio Shack Logo for the Tandy 32K Color Computer.

The content is presented in a straightforward and scholarly fashion and takes you into Logo at a fairly rapid speed. There are some good technical points made and I particularly liked the section on the problems of animation in Chapter 3. Suggestions for further exercises are given at appropriate points within the text and where necessary answers are given in Appendix F.

This book would not be suitable for anyone who just wants a superficial knowledge of Logo in order to play around with the graphics. It seems suitable for anyone who is conversant with their machine, who has some knowledge of other languages and who wishes to be able to program efficiently in Logo. It would probably be a suitable book for teachers in Computing Departments in secondary schools.

Criticisms? Well, it is a bit wordy in places and perhaps there could have been a few more illustrations, but I have found it accurate and a worthwhile buy at the price.

IN CONCLUSION I am still waiting for a book on Logo which is directed at parents and teachers and which shows the numerous ways

(continued on page 19)

Book Review

Title : ProStar Training Guide.
Author: Jane Davis
Price : £29.00 inc p&p.
Published by JD Publications, 38 Ethelden Road, London W12 7BG.

A Review by Jim Panks.

Manuals on software are either good, bad or indifferent. Jane Davis has decided to sit down and make the MicroPro range of software much easier to understand by writing an easy to understand manual. The manual contains all you need to know on the following packages : Wordstar, Spellstar, Datastar, Calcstar, Mailmerge and Supersort.

The manual adopts the tutorial form of learning, with all steps being explained down to the last detail, no over technical words have been used and the book is aimed directly at the user who wishes to get on and use his software. The ability to link the various packages has been explained and although this may sound daunting, this manual does the job in an excellent fashion.

The first chapter contains the general CP/M copying and setup data to enable you to go on to chapter two where making back-ups, tutorial disks and the printing of files are discussed.

The next chapter is devoted to Wordstar and goes through all the basic commands explaining how to enter data, store files and print documents. I found the style very pleasing, by the end of the chapter I was using Wordstar with confidence.

The following chapters in section one explain the DataStar and Mailmerge packages. The basics are covered well and help you to understand the ways in which you can put the packages to use.

Section two covers all the packages and takes you in depth through all the more advanced features. The interlinking of packages is explained in great detail, but still does not overwhelm the user. Many useful tips are given, and from this a good working relationship with the software can be obtained.

There is a glossary of terms to round off this excellent manual.

The price is very reasonable and is value for money. The manual comes in A4 size and the 200 pages are placed in a strong ring binder. The manual has large type with bold letters being used to emphasise user input; each page has been laid out so that the page is uncluttered and easy to read; the layout has been designed with thought. There is a very comprehensive index.

This book is ideal as a reference manual after the tutorial has been completed. It is anticipated that all the packages could be learnt in three weeks, this time scale assumes that you split the lessons up into two or three hour stints. I think that this is a reasonable time scale. This will probably suit the user in an office enviroment where a chapter a day in between other work would not interfere with normal routine.

I would recommend this manual to any user of Micropro Software, the author has devoted her wide knowledge of this software to produce one of the best software manuals I have seen. Maybe the software publishers will take heed and make manuals to this standard.

Lonely Apples

The following member would like to contact others in his area:-

J. Howard Wright
[illegible]

Tel: Alloa [illegible]

Education cont.

in which Logo projects can be linked with the rest of the school curriculum. After all, why program in Logo once you know how unless the programs can be shown to have a usefulness in the general process of education.

Fast Circles

by Ray Harris

Do you tire of plotting circles so slowly that you can count the points as they appear?

Here is a small set of programs that will at least keep you awake, although I don't guarantee that they are the fastest possible. Circles are plotted at the rate of 10 in about 3 seconds. I am grateful to Derek Ball of Leicester University School of Education for the basic idea, suggested for the RML 380Z.

The centre of a circle may be anywhere within the screen boundary, and the radius from 1 up to well over half the screen width. I wrote the programs for the ITT 2020, which involves some scaling to create a true circle, as does my printer. You may enter your own required scale factor in the 'Circle Creator' program. To adapt the code for the Apple requires only two changes to the hex listing, both noted. Relocating the code will require just one change, in line 178.

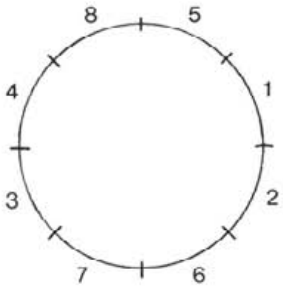
You should enter the hex from $94E0 to $95D5 and BSAVE CIRCLE.B,A$94E0,L$F6. Next RUN CIRCLE SETUP, which will write data from $9400 to $949F (the coordinates of points on one eighth of a circle of radius 128 units) and then save the whole memory from $9400 to $95FF as 'CIRCLE.M/C'. Only 'CIRCLE CREATOR' is required in future for producing circles, together with the new hex program.

'CIRCLE CREATOR' has no built in error checking, because you are likely to adapt it for your own purposes. However, there is nothing wrong with drawing a circle which goes partly off the screen: that part will not be plotted.

In drawing each circle, with the flag holding 0, the sections 1 to 4 are plotted, each point in 1 being followed by its images in sections 2, 3 and 4. When the flag holds 1 sections 5 to 8 are plotted similarly. The two plotting halves correspond effectively to different radii in the X and Y coordinate directions, so changing SC will produce ellipses. See what you can do with it!

```
10  REM     ********************
20  REM     *  CIRCLE SETUP    *
30  REM     *   RAY HARRIS     *
40  REM     ********************
50  HIMEM: 37888
60 BASE = 148 * 256
70 AD = BASE
80  FOR I = 0 TO 79
90 X =  COS(I / 100)*128:Y =  SIN(I / 100)*128
100 X =  INT (X + 0.5):Y =  INT (Y + 0.5)
110  POKE AD,X: POKE AD + 1,Y
120 AD = AD + 2
130  NEXT I
140 D$ =  CHR$ (4)
150  PRINT D$"BLOAD CIRCLE.B"
160  PRINT D$"BSAVE CIRCLE.M/C,A$9400,L$200"
```

```
10  REM     ********************
20  REM     *  CIRCLE CREATOR  *
30  REM     *   RAY HARRIS     *
40  REM     ********************
50  HIMEM: 37888
60  PRINT  CHR$ (4)"BLOAD CIRCLE.M/C"
70  POKE 235,0: POKE 236,148: REM  BASE IN $EB.EC
80 CIRCLE = 38112:FLAG = 6
90  TEXT : HOME
100 SC = 96 / 79: REM  SCALING FACTOR FOR ITT & MY
MONITOR
110   REM  FOR APPLE  SC IS CLOSE TO  1; CHANGE SC
FOR SOME PRINTERS
120  INPUT "WHAT POINT IS THE CENTRE?(X,Y) ";X,Y
130   POKE 227,Y:  POKE 237,X  - 256 *  INT  (X /
256): POKE 238, INT (X / 256)
140  PRINT
150   PRINT "WHAT  IS THE RADIUS?  (<" INT  (255 /
SC)") ";: INPUT "";R
160  HGR2 : HCOLOR= 3:R1 = R * SC
170  POKE 213,R1: POKE 215,R
180  POKE FLAG,0: CALL CIRCLE: REM  LEFT & RIGHT
190  POKE 213,R: POKE 215,R1
200  POKE FLAG,1: CALL CIRCLE: REM  TOP & BOTTOM
210  REM  TO RUN AGAIN TYPE 'RUN 80'
```




```
                1  ********************************
                2  *                              *
                3  *          CIRCLE.S            *
                4  *                              *
                5  *         RAY HARRIS           *
                6  *                              *
                7  *           5/7/83             *
                8  *                              *
                9  ********************************
               10
               11  FLAG     =   $06        ;flag for halves
               12  XCOORD    =   $19
               13  XPRODUCT  =   $1B
               14  XOFFSET   =   $1D
               15  TEMP1     =   $1F
               16  R1STORE   =   $D5
               17  RSTORE    =   $D7
               18  YCENTRE   =   $E3
               19  BASE      =   $EB
               20  XCENTRE   =   $ED
               21  YSTORE    =   $EF
               22  YCOORD    =   $F9
               23  YPRODUCT  =   $FB
               24  YOFFSET   =   $FD
               25  TEMP0     =   $FF
               26  HPLOT     =   $F498
               27  *
               28  *HPLOT IS $F498 for ITT and $F457 for APPLE
               29  *
               30          ORG  $94E0
94E0: 48       31          PHA
94E1: A0 00    32          LDY  #$00
94E3: B1 EB    33  GETX    LDA  (BASE),Y
94E5: 85 19    34          STA  XCOORD
94E7: A5 D5    35          LDA  R1STORE
94E9: 85 1F    36          STA  TEMP1
94EB: A5 D7    37          LDA  RSTORE
94ED: 85 FF    38          STA  TEMP0
94EF: A9 00    39          LDA  #$00
94F1: 85 1A    40          STA  XCOORD+1
94F3: 85 1B    41          STA  XPRODUCT
94F5: 85 1C    42          STA  XPRODUCT+1
94F7: 85 1D    43          STA  XOFFSET
94F9: 85 1E    44          STA  XOFFSET+1
94FB: 85 FB    45          STA  YPRODUCT
94FD: 85 FC    46          STA  YPRODUCT+1
94FF: 85 FD    47          STA  YOFFSET
9501: A2 08    48          LDX  #$08      ;to count shifts
9503: 46 1F    49  MULTR1  LSR  TEMP1
               50  *
               51  *calculate X-coord*R1/128
               52  *
9505: 90 0E    53          BCC  XLEFT
9507: 18       54          CLC
9508: A5 19    55          LDA  XCOORD
950A: 65 1B    56          ADC  XPRODUCT
950C: 85 1B    57          STA  XPRODUCT
950E: A5 1A    58          LDA  XCOORD+1
9510: 65 1C    59          ADC  XPRODUCT+1
9512: 85 1C    60          STA  XPRODUCT+1
9514: 18       61          CLC
9515: 06 19    62  XLEFT   ASL  XCOORD
9517: 26 1A    63          ROL  XCOORD+1
9519: CA       64          DEX
951A: D0 E7    65          BNE  MULTR1
```

```
951C: 06 1B    66            ASL  XPRODUCT
951E: 26 1C    67            ROL  XPRODUCT+1
9520: A5 1C    68            LDA  XPRODUCT+1
               69  *
               70  *use the high byte to effect the division
               71  *
9522: 85 1D    72            STA  XOFFSET
9524: C8       73            INY
9525: B1 EB    74  GETY      LDA  (BASE),Y   ;do the same for Y
9527: 85 F9    75            STA  YCOORD
9529: A9 00    76            LDA  £$00
952B: 85 FA    77            STA  YCOORD+1
952D: A2 08    78            LDX  £$08
952F: 46 FF    79  MULTR     LSR  TEMP0
9531: 90 0E    80            BCC  YLEFT
9533: 18       81            CLC
9534: A5 F9    82            LDA  YCOORD
9536: 65 FB    83            ADC  YPRODUCT
9538: 85 FB    84            STA  YPRODUCT
953A: A5 FA    85            LDA  YCOORD+1
953C: 65 FC    86            ADC  YPRODUCT+1
953E: 85 FC    87            STA  YPRODUCT+1
9540: 18       88            CLC
9541: 06 F9    89  YLEFT     ASL  YCOORD
9543: 26 FA    90            ROL  YCOORD+1
9545: CA       91            DEX
9546: D0 E7    92            BNE  MULTR
9548: 06 FB    93            ASL  YPRODUCT
954A: 26 FC    94            ROL  YPRODUCT+1
954C: A5 FC    95            LDA  YPRODUCT+1
954E: 85 FD    96            STA  YOFFSET
9550: 84 EF    97            STY  YSTORE
9552: A5 06    98            LDA  FLAG
9554: F0 0C    99            BEQ  PLOT1
9556: A5 1D    100           LDA  XOFFSET
9558: A6 FD    101           LDX  YOFFSET
955A: 85 FD    102           STA  YOFFSET
955C: 86 1D    103           STX  XOFFSET
955E: A9 00    104           LDA  £$00
9560: 85 1E    105           STA  XOFFSET+1
               106 *
               107 *HPLOT needs X-coordinate in reg.Y,X
               108 *and Y-coordinate in reg.A
               109 *
9562: 18       110 PLOT1     CLC
9563: A5 ED    111           LDA  XCENTRE
9565: 65 1D    112           ADC  XOFFSET
9567: AA       113           TAX
9568: A5 1E    114           LDA  XOFFSET+1
956A: 65 EE    115           ADC  XCENTRE+1
956C: F0 0B    116           BEQ  XOK        ;X-coord <256
956E: C9 02    117           CMP  £$02
9570: B0 28    118           BCS  PLOT3      ;X>511
9572: E0 68    119           CPX  £$68
               120 *
               121 *(£$68 for ITT, £$18 for APPLE)
               122 *
9574: B0 24    123           BCS  PLOT3      ;X>359 or 279
9576: A8       124 XOK       TAY
9577: 18       125           CLC
9578: A5 E3    126           LDA  YCENTRE
957A: 65 FD    127           ADC  YOFFSET
957C: B0 07    128           BCS  PLOT2      ;Y coord>255
957E: C9 C0    129           CMP  £$C0
9580: B0 03    130           BCS  PLOT2      ;Y coord>191
9582: 20 98 F4 131           JSR  HPLOT
9585: 18       132 PLOT2     CLC
9586: A5 ED    133           LDA  XCENTRE
9588: 65 1D    134           ADC  XOFFSET
958A: AA       135           TAX
958B: A5 1E    136           LDA  XOFFSET+1
958D: 65 EE    137           ADC  XCENTRE+1
958F: A8       138           TAY
9590: 38       139           SEC
9591: A5 E3    140           LDA  YCENTRE
9593: E5 FD    141           SBC  YOFFSET
9595: 90 03    142           BCC  PLOT3      ;Y<0
9597: 20 98 F4 143           JSR  HPLOT
959A: 38       144 PLOT3     SEC
959B: A5 ED    145           LDA  XCENTRE
959D: E5 1D    146           SBC  XOFFSET
959F: AA       147           TAX
95A0: A5 EE    148           LDA  XCENTRE+1
95A2: E5 1E    149           SBC  XOFFSET+1
95A4: 30 24    150           BMI  NEXT       ;X<0
95A6: A8       151           TAY
95A7: 18       152           CLC
95A8: A5 E3    153           LDA  YCENTRE
95AA: 65 FD    154           ADC  YOFFSET
95AC: B0 07    155           BCS  PLOT4      ;Y>255
95AE: C9 C0    156           CMP  £$C0
95B0: B0 03    157           BCS  PLOT4      ;Y>191
95B2: 20 98 F4 158           JSR  HPLOT
95B5: 38       159 PLOT4     SEC
95B6: A5 ED    160           LDA  XCENTRE
95B8: E5 1D    161           SBC  XOFFSET
95BA: AA       162           TAX
95BB: A5 EE    163           LDA  XCENTRE+1
95BD: E5 1E    164           SBC  XOFFSET+1
95BF: A8       165           TAY
95C0: 38       166           SEC
95C1: A5 E3    167           LDA  YCENTRE
95C3: E5 FD    168           SBC  YOFFSET
95C5: 90 03    169           BCC  NEXT       ;Y<0
95C7: 20 98 F4 170           JSR  HPLOT
95CA: A4 EF    171 NEXT      LDY  YSTORE
95CC: C8       172           INY
95CD: C0 A0    173           CPY  £$A0
               174 *
               175 *80 points, 2 bytes per point
               176 *
95CF: B0 03    177           BCS  END
95D1: 4C E3 94 178           JMP  GETX
95D4: 68       179 END       PLA
95D5: 60       180           RTS


--End assembly--

246 bytes

Errors: 0
```

Permanent & Temporary Strings

IN BASIC OR

HOW TO AVOID GARBAGE COLLECTION

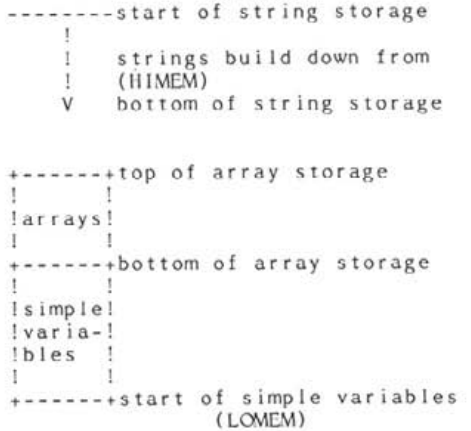
by Richard Teed

Garbage collection is something that seldom bothers a user of BASIC: this is because memory generally takes a lot of time to become exhausted, but when it does and a garbage collection is invoked then the computer may hang-up for about five minutes and I believe the theoretical limit is somewhere in the region of four hours!

If one thinks about it, it is possible to extract enough information from a program to tell BASIC when and where to garbage collect so that a full scale garbage collection is not necessary. The problem is that no facilities exist at a high level to make this possible.

Programs that use strings fall into two groups when creating strings: the first is constantly creating temporary strings as in copying one file to another, and the second has a number of permanent strings from which temporary strings or further permanent strings are created.

Let us examine how to avoid garbage collection on the two classes of strings, but first here is a memory map of BASIC's variable space to make the explanation easier:



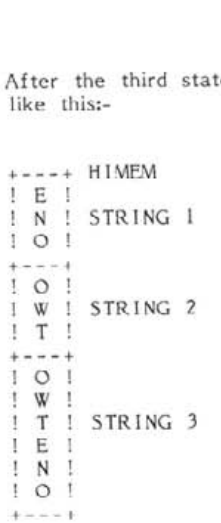
What causes a garbage collection is when there are so many strings that they have come down from HIMEM and are about to over-write the array section of memory, the interpreter then sees which strings can no longer be accessed and removes them moving valid strings up to take their place.

Consider the following program and suppose that "A$" and "B$" are to be permanent (by permanent I mean available to the program throughout its execution), while the value assigned to "C$" is temporary; that is it is only valid within the subroutine after which its value may be lost:-

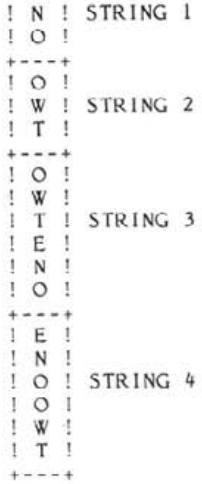
```
100 A$="ONE" : B$="TWO" : C$=A$+B$ :
  GOSUB 1000 : C$=B$+A$ : GOSUB 1000 :
  C$=A$+A$ : GOSUB 1000 : END

1000 PRINT C$ : RETURN
```

The first two statements produce the following strings in memory:-

After the third statement the memory looks like this:-



After the fifth statement however things look like this:-



STRING 3 will no longer be accessable and would be cleared by a garbage collection.

In order to make string "A$" and "B$" permanent and string "C$" temporary a little code must be added to the subroutine and just after all permanent strings have been created.

What needs to be known is the memory location at which the permanent strings stop and this is given in locations $6F and $70 (111 and 112). Now each time before exiting from the subroutine (any temporary strings having being finished with) we must put the bottom of string storage back to the bottom of the permanent strings so that the next temporary string overwrites the last. Here then is the complete program:-

```
100 A$="ONE" : B$="TWO" : PL=PEEK(111) :
  PH=PEEK(112)

110 C$=A$+B$ : GOSUB 1000 : C$=B$+A$ :
  GOSUB 1000 : C$=A$+A$ : GOSUB 1000 :
  END

1000 PRINT C$ : POKE 111,PL : POKE 112,PH
  : RETURN
```

The constant re-setting of the bottom of string storage means that it will never reach the top of array storage which would cause a garbage collection.

You can of course update the number of permanent strings simply be PEEKing location 111 and 112 just after the new permanent strings have been made.

Modifying BASIC's pointers like this can be dangerous if you are not careful, but basically the only thing that you should not do is to produce a new string from temporary strings that have just been released by re-setting location 111 and 112, because the new string will be created over the top of the strings that are to comprise it resulting in corruption.

Software Library

I have not received one program for inclusion in the software library since the last magazine in August. I would hope that this is due to holidays etc.

I would ask anyone with material that they think may help other people or would be suitable for the software library, especially Pascal, CP/M or programs for the younger user (Christmas is fast approaching), to send it in as soon as possible so that I can realease some more disks into the library. It's all up to you!

I will be writing to all those that have helped in the past to see if they are still willing to do the same work in the future. There is much work that needs to be done in respect of updating the software library to 3.3. DOS.

Our new Special Software is ready with Business Basic by Richard Teed, GR Pad by Mike Siggins and four programs by Ian Trackman. These are really good value and come complete with a manual. So send your orders in. The programs cost £11.50 each.

Last but not least, we are still looking for good quality program material for our Special Software Library. If you think that you have anything, please contact me for details of the standards we expect. Remember we pay for these.

Communications Update

by Q. REIDFORD

Following articles in the previous two issues of 'HARDCORE' and with the knowledge of two months use of 'MICRONET' I feel that it is time to put my experiences to BASUG members and poll their reactions.

As I see it there are two roads that we as a group can go down;

A) Micronet, via Prestel

B) Bulletin boards and data exchange similar to 'The Source'

A. PRESTEL (Micronet)

This system uses a Modem linked to the Apple via an RS232 interface to recieve data at 1250 Baud and transmit at 75 Baud. The initial cost of the HARDWARE is £75.00 (special offer to BASUG) plus Prestel fee of £5.00 per quarter plus the Micronet fee.

The Modem for the Apple will be a direct connect version with a switchable Baud rate of either 1250/75 OR 1200/1200, however the 1200/1200 baud rate is HALF-DUPLEX i.e. if two people are communicating then one must agree to be the sender and the other the recipient. This protocol is set by the use of a switch on the modem, software switching will come, but I am unsure whether it can be envoked without disconnecting.

The most important aspect however has got to be the content, the information to be gleaned or exchanged. Prestel has an extremely good service called 'Viewfax' which has Micro-news, software, and gossip. It is well laid out and interesting, and although it is not Apple specific, this 'magazine' could be emulated to a great extent by BASUG on our own pages.

Software that is available is either free or relatively cheap (BBC), however I have found that a lot of the programs are poor quality (free ones!!) and can take a long time to down-load, particularly if the built-in checksums don't and the page in question has to be taken down a second or third time - however that is all part of the initial fun of it!!

Letters which are sent in to the magazine are put up on a 'bulletin board' as are the answers, but this interaction is only as good as the management of the magazine.

Information on club events and meetings are available as are dates for computer shows etc. The gossip section of Viewfax, rejoicing under the name 'Micrognome', is both interesting and fairly uncompromising.

However there are several problems that I feel members should be aware of and depending on their requirements, consider carefully before parting with their cash.

1. Telephone costs can become very high, particularly if your nearest Prestel computer is not a local call, mine isn't..!

2. Telephone line noise can be very troublesome, again if you have to use long-distance. This will be helped by using the direct connect modems. The acoustic version is distracted by kids shouting and requires care in positioning the handset.

3. Unless you live in the London area, you cannot use the 'Mailbox' facility to any great advantage. The local computers only hold one message at a time and do not transmit that message to the other computers. I cannot send a message to a Prestel user unless he is registered on the same computer, OR I call London....Bad lines etc.

4. Particular Prestel facilities such as colour and graphics, while attractive, are not essential in transferring information, programs etc. But they are a nightmare to produce and take precious time to transmit.

5. Unless you particularly want to know what films Quantas are showing on their long-haul flights, or you frequently book theatre seats or hotels/flights, you will tend to use the system to collect 'mail', with perhaps a quick excursion to particular pages of interest.

B. BULLETIN BOARDS.

These are public access data bases which are run by enthusiasts such as ourselves. They use a standard 300 baud FULL-DUPLEX modem connected via an RS232 interface.

I know of seven of these bulletin boards and have accessed four of them using a borrowed Modem, Apple communications card, and a standard communications package. I believe that all these data-bases are run with TRS-80 computers with about 200K disk storage. I understand that the Tandy is used because of the host software, bought in the States, but there is little, apart from the Auto-answer modems, that many BASUG members do not already have.

The principal feature of this type of data-base is the ability to leave messages, either to a particular person, or as an open letter, for example asking for help.

There is also a section for down-loading programs, mainly communications, and mainly for TRS-80, Apple takes a back seat.!! However the programs that I have taken down for a BBC using the Apple seemed worthwhile, and free.

There are SIG pages and sales and wants, all good stuff, and easy to access.

A feature which is particularly friendly is the 'chat' mode. In this mode, if the system operator is near his machine he can break in for a chat, on request, and help out immediately with information or whatever. Interaction and direct communications are very much more enjoyable than the passive role of Prestel.

The Bulletin-boards can emulate any terminal and you configure the system on initial entry by answering a few questions about screen width etc. You then choose a password and from then on you will get messages etc. on signing on and of course the system will remember your terminal configuration.

It is particularly nice to see a few users from Finland and Denmark using the system. In the Hull Bulletin board there are many callers from the States, which makes it VERY interesting.

For those members with access to the necessary equipment, here are some numbers and instructions:

1.Liverpool (24 hours) 051-428-8924
This is an auto-answer system.
If using an Apple press return when asked for terminal type.
2.Hull (9pm-1am) 0482-859169
I get a poor line so access limited, but it's the most interesting.
3.North-East (9pm-1am) 0207-43555
4.Stourport (???) *0299-32853
5.London (8pm-1am) *01-348-6518

* Dial number, allow 2 rings, hang up and re-dial for connection to computer.

All systems are 300 Baud, 7 bit even +1 parity.

PROBLEMS:

1. Phone costs still high, but information better.
2. No Apple users, yet.....

My own view is that there is much more scope for the bulletin board approach to communications, than the Prestel/Micronet route. The Apple has a wealth of commercial hardware and software for 300 baud data-bases, there are many articles on the subject in American magazines, and as we are all using the same machine we can happily exchange information Modem to Modem with each other without an intermediate stage.

I want to be able to look through messages and problems, make contact with other members, exchange programs etc. etc.

I do not believe that Micronet can adequately provide this service, particularly as it depends on intermediate editing. Members will pay, through Prestel, for a service that they may never use, and unless they really want to USE Prestel then I believe that BASUG should look to the interactive 300 Baud bulletin board.

In terms of hardware and costs, we know that the Micronet package will give us basic communications as already described, for £75.00

What I would propose is that members would buy a freely available modem kit from MAPLIN at £39.95 and a serial card such as an RS 232 interface. The Maplin modem does NOT have BT type approval, although it will comply when the wheels grind into motion: Many people use them and they are reputed to be excellent.

The main problem is that it comes as a kit, on top of this you must know something about electronics to get it to work. I certainly could not build one without help, and I imagine that I am not alone !!!

However it sufficient people were interested we would hold a series of regional or local seminars where members could come for the day, armed with a soldering iron, and, under the guidance of an expert, build their own - assuming known numbers we could even arrange for the kits to be available for purchase at the seminar.

With the communications card we have the problem of cost again, the Apple comms. card costs about £100 and that's a lot!

Hardware designers, here is your opportunity! Can we please have some-one offer their services (paid!) to design a basic 300 Baud comms. card. This could be available either as a kit or as a built unit, and if the cost could be kept down to around £30.00 then the whole package of Modem/card would be in the same ball-park as the Micronet offer.

On the telephone costs aspect then I think that we may well have a network of volunteers throughout the Country dedicating their machines (probably on a rota). BASUG would supply the software and all you would have to do, if the modem was auto-answer, would be to leave your machine switched on for a few hours in the evening.

I would think it essential that BASUG had a central data-base, auto-answer, 24 hour service which would be the principal point of reference. This central system could contain chunks of the software library, ordering service and 'help' centre.

It is essential if we are to continue with this excercise that we can get an accurate indication of our members wishes and therefore I would appreciate it if every-one could send answers to the following questions as soon as possible so that we can then arrange either that the Micronet connection is enhanced OR that we organise the kits, software and hardware to get our own Bulletin-board under way without delay.

COMMUNICATIONS SURVEY:

---

1. Are you intereseted in using your system for communications?

2. Are you interested in purchasing the MICRONET system?

3. Are you interested in purchasing a 300 Baud system?

4. Would you wish to attend a construction seminar?

To assist in arranging suitable seminar centres could you please give your address and telephone number.

If you can contribute either in software or hardware design please include details.

There is NO obligation to purchase anything by answering these questions.

We have already had a very good response to the extract from Quentin's article that appeared in the Update. Here are some quotes to give you an idea of what has been said so far.

Tony Game of Felixstowe, Suffolk.

I am very much concerned with the whole subject as owing to a disability which prevents me from travelling much, I have been unable to partake in the affairs of BASUG as much as I would have wished. I see a responsive and versatile communications system as being very much the answer to this.

I heartily agree with you that a system whereby one has to establish a verbal

contact before sending anything is far from ideal. It seems to me that if the Call Apple style of hotline is to be established it could most easily be done via an Apple somewhere being available to record a problem or request, the answer when possible being sent back to the waiting Apple at the other end. I would expect that in a very short time any system which is not really flexible will be obsolete.

John Stevenson of East Hendred, Oxon.

How long will it be before we can use one universal/programmable modem that will serve all the various options for data transmission. I am all for flexibility if it can be achieved at not too great a cost but I am concerned at the number of promises of availability or changes in the not far distant future that come out from many organisations.

Colin Madge of Croydon.

Yes. I think a bulletin board is a natural and useful extension to the Apple and BASUG but as an adjunct to not a replacement for a Prestel/Viewdata system. My own personal wish is also to have the ability to download broadcast teletext into the Apple as well.

With speed, 300 Baud is the accepted standard but 1200 Baud with error correction is possible. Whilst any decision taken to ensure a quick service would I think revolve around 300 Baud and a cheapish modem, I hope this would be only an interim decision to be extended later if necessary.

Ewen Wannop of Box, Wiltshire.

I have been using both bulletin boards and Micronet 800 for some time now, the former with a 300 baud modem andd the latter with the acoustic modem provided with the Micronet package for the Beeb. The 300 baud modem is based on the Maplin kit and runs to the European standards. The two systems are really completely different. Micronet and its parent Prestel are mainly one-way information sources. It is of course possible to upload/download programs and leave messages, but as the system grows it will rapidly be swamped by the Beeb and other users. It will then be difficult for Apple users to find what they want quickly, unless the BASUG pages are to become fairly complex. However the 300 baud boards are for information exchange on a level that is not swamped by other data. The mailbox facilities on these boards allow fast and easy access to information transfer and ease of directing these messages to anyone as needed. Program exhange is easily implemented, the limitation simply being the amount of disk space that can be put on-line.

Roland Saam of London SW7.

I want to do reliable user to user text transmission ideally with mailbox facility. 300 baud sounds fine - my volumes of text will not be large.

Donald MacLean of Chorleywood.

I think:

1. That the Apple presence on Micronet should be expedited (i.e. that the cheap modem and card should become available pdq), AND

2. That BASUG should establish a 300 Baud Bulletin Board, AND

3. That direct Apple to Apple communication at 300/300 should be actively fostered.

****

IMPORTANT

Please read the update.

****

Ceemac

by John Molloy and Colin Holgate

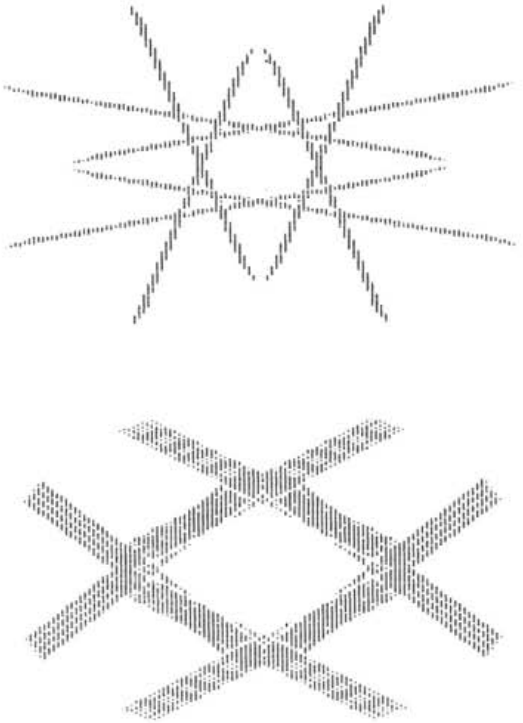
Ceemac is an abstract graphics language created by an American called Brooke Boering. It looks a little like Pascal when written down, and it has two major sections: the editor and the interpreter. All the variables are predefined and are prenamed. Some for example, read the keyboard or give the state of the paddles or the state of the cassette input port, others are for tables and giving random numbers over specific ranges. It also has a library of almost 30 macros which includes a trace function and the ability to send sound out through the speaker.

Mr Boering's concepts are simple to follow but Ceemac does have some strange eccentricities. For example instead of writing a program you compose a score regarding the graphic displays as visual scores and not as programs. However if you also ignore the fact that everything else connected with the disk has a name (edgar the editor and cindy the proofreader etc). It is surprisingly easy to write impressive abstract graphics.

All the scores from the Fire Organ demonstration disk are supplied with the language in editing form and this gives the best start to writing your own routines, the suggestion being to start with something which is "sorta close" to what you would like to see on the screen and then experiment until you a) get what you want or b) (the more likely) end up with something very impressive but not what you were aiming at in the first place.

When the disk is first booted, it comes up with the Fire Organ logo. Then the program jumps into the fire-organ routine. In fact it looks the same as Fire Organ except that none of the keys apart from the two arrow controls seem to work. By pressing ctrl-A the screen jumps to text and it is in editing mode. The current score is the fire-organ score. To execute it again enter ctrl-A this is how to move between the editor and the interpreter. Once in the editor ctrl-C jumps Ceemac into DOS and to run another score you simply brun it. To re-enter Ceemac if you jump out accidentally then just type *800G.

Each score is limited to 256 statements. These include title, end statement and any comments. You can only put one statement on a line. For example one of the scores which comes with the disk is called ko (Standing for key O on the Fire Organ disk).

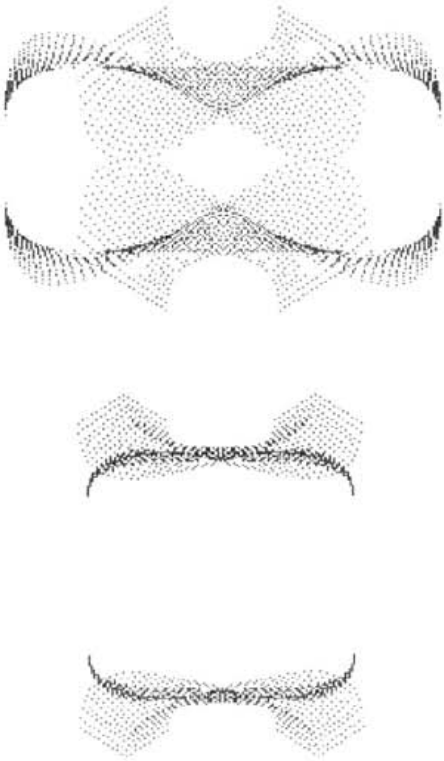


```
SCORE: KO-21ST CENTURY MAN (SLOW)
  :
  : INITIALIZE
  :
  SPEED [0;0]
  CLEAR [0;0]
  V2 = 1
  :     V2 = INCR. FOR Y CORDINATES
  :
  V3 = 2
  :     V3 = INCR. FOR X CORDINATES
  :
  V4 = 7
  :     V4 = SEMMITRY
  :
```

```
:
: MAIN CHAIN
0
:
: INIT. FOR NEW PATTERN
SKY [$0F;0]
XY2 = RANDOM;$80
COLOR = NXTCOL
:       SET COLOR FOR NEW PATTERN
:
: DRAW PATTERN ERASE AFTER 8 LINES
FOR $10
  BLINE [0;V4]
  Y2 = Y2 + V2
  X2 = X2 + V3
  Y1 = Y1 + V2
  X1 = X1 + V3
  BRASE [8;0]
AGAIN
CEEMAC  REL 1.0
```

From its listing; the top line is the title line; the colon is the comment symbol. SPEED {0;0} is the macro which overrides the paddle setting. CLEAR {0,0} sets the screen to black. V# are the free variables and can be set as required. 0 in the line under the comment : MAIN CHAIN is where the program loops back to when the left arrow is pressed. All ceemac programs loop back on themselves so that when it hits the last statement (CEEMAC REL 1.0) it jumps back to the top of the score. SKY {$0F;0} fills the screen with stars but due to the $0F in the first parameter it fills the screen with 15 stars. XY2 = RANDOM;$80 is the same as typing X2 = RANDOM Y2 = $80 and is the end position for the BLINE macro below. RANDOM is a random variable between 0-$FF. COLOR= NXTCOL gets the next colour off of the hires colour table. COLOR is the variable which holds the present plotting colour. FOR $10 is the start of a loop which ends at the AGAIN. It is in hex. BLINE {0;V4} draws a straight line XY1 holding the start position XY2 holding the end and COLOR holding the colour of the line. The parameter set to V4 in the macro is the symmetry (if it was set to 0 it would draw a single line but V4 is set to 7 which reflects in all eight sectors of the screen. Y2 and Y1 are incremented by V2 which is set to 1 while X2 and X1 are incremented by V3 which is set to 2. BRASE {8;0} erases the BLINE plotted eight generations later (so after it has plotted the eighth line it erases the first, after the ninth the second etc.). After this loop has been executed sixteen times the computer jumps to the top of the score.

After a short while I found it quite easy to understand what was going on and to write simple but effective graphic displays. Spline runner was one of my early attempts to write a Ceemac score, along with it I have included some graphics dumps of the routines.



```
SCORE: SPLINE RUNNER
  :
  :     AUTHOR JOHN MOLLOY
  :
  : A CEEMAC SCORE ILLUSTRATING
  : THE USE OF THE JORDAN CURVE
  : KNOWN TO CEEMAC AS SPLINES
  :
  : IF I CAN DO IT ANYONE CAN
  :
  : SELECT THE FASTEST SPEED
  : THIS IGNORES THE PADDLES
  SPEED [0;0]
  :
```

```
: NOW CLEAR THE SCREEN BLACK
CLEAR [0;0]
:
: SET WHITE AS THE COLOUR FOR
: PLOTTING
COLOR = $FF
:
: SELECT TO RANDOM START POSITION
: BETWEEN 0-127
XY1 = RND7;RND7
XY2 = RND7;RND7
:
:
: THIS IS WHERE TO LOOP BACK TO
F
:
: GIVE A RANDOM FIGURE WITH WHICH
: TO WEIGHT THE SPLINE
: IE THE AMOUNT OF BENDING APPLIED
: TO THE SPLINE
XY0 = RND7;RND7
XY3 = RND7;RND7
:
: ADD OR SUBTRACT UNTIL THE SPLINE
: HITS A WALL
X1 = X1 % 2
Y1 = Y1 % 2
:
: NOW WE GET TO DRAW THE SPLINE
SPLINE [8;6]
:       ^ ^
:       I I
:       I VALUE OF SYMETRY
:       MAXIMUM FORCE ON SPLINE
:
: ERASE THE 20TH LEVEL SPLINE
SPRASE [$14;0]
:
: LOOP BACK ROUND TO F
CEEMAC  REL 1.0
```

As you can see it is quite varied even from a program as short as this (12 lines if you take out all the comments. A SPLINE is a curved line (as opposed to a BLINE which is a straight line) and the mathematical formula for this macro was supplied by Mik Jordan (hence the other reference to them as Jordan's curves) The manual gives a two and a half page explanation of how they work as the concept is quite hard to grasp but the effect is stunning. The graphics dumps don't give any impression of movement which is how the scores 'evolve' in real time.

On the whole Ceemac is an interesting graphics package. If you are into fast moving hi-resolution abstract visuals, then it is not to be missed. Anything omitted from the manual is covered in notes left in BRUN-able files on the disk, and are easy to access. But if you would just like to see what Ceemac can do then the package Fire Organ is for you.

Fire Organ

No amount of describing Ceemac can really show what it can do for you, as per the formula '1 picture = 1000 words'. Fortunately the makers of Ceemac have put together a 'visual album' of Ceemac 'scores'. This analogy to music is a good way of putting things. The 'Fire Organ' disk (available from the software library) is a collection of visual 'tracks' created on Ceemac and put together with another program called Maestro. By combining many scores together a variety of effects can be triggered without having to load in another effect file. This is done by assigning a separate key for each score. Altogether there are 34 different Ceemac files, operated by hitting the keys 1 to 9 and A to Y.

In general the different effects are either moving lines, spline dot lines, kaleidescope patterns or actual shapes. Many of the parameters can be altered during these routines, for example the rate at which the lines are drawn/erased can be adjusted by moving paddle 0, or the next lap of the effect can be triggered by pressing button 0. Also paddle 1 may affect how many lines appear at once, and button 1 changes direction or colour of the lines. This depends on which of the 34 scores you are watching. In any case the left arrow duplicates the role of button 0, and the right arrow does the same for button 1. By pressing, hitting or moving these controls you have dynamic control over the score as it is being shown. Effectively this means you 'play' the score. This can be very impressive if the playing is done to music, at least we hope it's impressive, Mainframe has used Fire Organ extensively over the past 10 concerts!

The Ceemac scores on the Fire Organ disk were written to be accompanied with music. Apart from the main display program, the disk also has several binary files on it labelled 'About...'. These are to give the user an insight into the thinking behind

Ceemac and the scores themselves. When run these files display a text account of what inspired the writing of the different scores. This seems to vary from a favourite piece of rock music (Mainframe probably), to having a drink with a girl friend! Some of the text is of no particular value, but is part of a friendly approach that Ceemac has throughout. This is a bit hard to explain. Put simply, Ceemac is written in a familiar way, the sort of program you would write for your own use. Things are often explained in an amusing way. We have heard criticism of this from someone whose sense of humour is on a different wavelength to Brooke Boering and his friends, but we get on well with the friendly method used.

Other files explain the different parts of Ceemac, such as the program editor, the text editor and other Vagabondo products.

Altogether then, Fire Organ would be worth buying as a separate product, just to watch while listening to music. As it is Fire Organ is effectively free, to help sell the Ceemac language itself. Ceemac doesn't appear to be available in Britain, but can be bought in America. Maestro is also available from Vagabondo. Further details are given in the afore mentioned binary files on Fire Organ.

hardcore

Contributions to Hardcore are always welcomed. Without them it would not exist. Whether you have a learned article, a 'quick tip', a problem or a solution, let us know.

If you have more than a short letter, it is helpful if you could send it on a disk, with a hard copy if possible. If you want to send printed copy, please do not cut it up but fill justify on a 9 cm (3.5 ins) column. We prefer copy on disk, either as a text file or an Applewriter I file. Pascal text files and others such as Wordstar files can probably be accommodated. Please use the minimum of embedded printer commands.

Remember, contributors receive credits for cheap Software Library or blank disks.

DOS EOF

AN ALTERNATIVE DOS END OF FILE

by Richard Teed

One of the problems with reading data files in BASIC is that it is very difficult to deal with an end of file, either data has to be put in the file to show the end of file or an "ONERR" command has to be used.

Although the "ONERR" command is usable it has a number of problems the worst of which is that file reading cannot be done in a subroutine because the stack is upset.

The following is the DOS readbyte routine which is used for reading text files from disk:-

```
                ORG     $A626
                MSB     ON
                LST     OFF
;
ISBASRUN        EQU     $A65E
CLSWARM         EQU     $A679
CSWSTATE        EQU     $AA52
RDTEXT          EQU     $A68C
CLOSFREE        EQU     $A2FC
RTS5            EQU     $A60D
DOERROR         EQU     $A6D2
ASAVE           EQU     $AA5C
XSAVE           EQU     $AA5A
IN              EQU     $0200
DOSXIT          EQU     $9FB3
;
;       DOS 3.3 READ BYTE ROUTINE
;
;*** READ A BYTE
;
READBYTE        JSR     ISBASRUN        ;Is BASIC running ?
                BCS     CLSWARM         ;If not then shut shop
                LDA     #6              ;Set DOS to state 6 (no output
RDB.1           STA     CSWSTATE        ; of prompt)
                JSR     RDTEXT          ;Read a byte from the file
;                                       ; into the accumulator
                BNE     TSTLWR          ;If not an end of file terminator
;
;*** DEAL WITH END OF FILE
;
EOF             JSR     CLOSFREE        ;Close the file
                LDA     #$03            ;Is DOS in state 3 (EXEC)
                CMP     CSWSTART
                BEQ     RTS5            ;If so then exit to caller
RD.2            LDA     #$05            ;Get theerror number for:
                JMP     DOERROR         ;An "END OF DATA ERROR"
;
;*** CHECK AND DEAL WITH LOWER CASE
;
TSTLWR          CMP     #$E0            ;Is the character lower case ?
                BCC     LL53            ;If not then skip
                AND     #$7F            ;Else remove the MSB
LL53            STA     ASAVE           ;Save A
                LDX     XSAVE           ;Get our position in the input buffe
                BEQ     LL63            ;If at the start then skip
                DEX
                LDA     IN,X            ;Set the MSB on the character
                ORA     #$80            ;(last character planted may have
                STA     IN,X            ;been fixed for lower case)
LL63            JMP     DOSXIT          ;EXIT
```

As can be seen each byte read is placed in the input buffer until a null byte is returned; then an end of file condition exists and all text files are closed and an "ENDOFDATAERROR" is output. The exception to this is if the text file is being EXECed, in which case the file is simply closed.

What is needed is for a special character to be returned on the end of file being reached and for this I have chosen CHR$(127).

Ideally a check would be made on the number of characters read into the input buffer. If the present one was the first then a CHR$(127) ($7F) would be returned. This would satisfy the GET command. If the present character were the second then the routine would assume that the command was an INPUT and would return a <RETURN> character: thus a CHR$(127) would be returned for both GET and INPUT.

Unfortunately things aren't that simple and, instead of the "X" offset being zero for a GET, it's a one, so the <RETURN> character must be returned on the third character (ie when "X" is 2). This will result in the single character returned to GET being the desired CHR$(127) but the string returned to INPUT will be a double CHR$(127). Still, this should not cause any real problems.

Here, then, is the modified DOS for our improved end of file condition (n.b. on end of file you have to do the closing yourself).

```
                ORG     $A626
                MSB     ON
                LST     OFF
;
SKIP            EQU     $2C
RETURN          EQU     $8D
ISBASRUN        EQU     $A65E
CLSWARM         EQU     $A679
CSWSTATE        EQU     $AA52
RDTEXT          EQU     $A68C
CLOSFREE        EQU     $A2FC
RTS5            EQU     $A60D
DOERROR         EQU     $A6D2
ASAVE           EQU     $AA5C
XSAVE           EQU     $AA5A
IN              EQU     $0200
DOSXIT          EQU     $9FB3
;
;
```

```
;
;       IMPROVED DOS 3.3 READ BYTE ROUTINE
;
;*** READ A BYTE
;
READBYTE        JSR     ISBASRUN        ;Is BASIC running ?
                BCS     CLSWARM         ;If not then shut shop
                LDA     #6              ;Set DOS to state 6 (no output
RDB.1           STA     CSWSTATE        ; of prompt)
                JSR     RDTEXT          ;Read a byte from the file
;                                       ; into the accumulator
                BNE     TSTLWR          ;If not an end of file terminator
;
;*** DEAL WITH END OF FILE
;
; This routine returns a CHR$(127) for a GET
; And a two character string both CHR$(127) for an INPUT
;
EOF             LDX     XSAVE           ;Get our current position in the input
buffer
                CPX     #$02            ;Is this the second character ?
                BNE     GET             ;If not then this is a GET not an INPUT
                LDA     #RETURN         ;Terminate string with a <RETURN>
                DFB     SKIP            ;Skip the next sequential word
GET             LDA     #$FF            ;End of file character
                NOP
                NOP                     ;Pad out to the same length as the
                                         original
                NOP
;*** CHECK AND DEAL WITH LOWER CASE      (as above)
```

There are two things to be remembered with the above routine: The first is NEVER EXEC a file with it because if you do the result will be an unstoppable infinite loop, and the second is that it will not work for INPUTting to numeric variables.

A big advantage of this method for dealing with the end of files is that it will work with compiled programs as well as normal interpreted programs.

In order to use the routines easily, here is the data in poke form. You can put this into your programs at strategic places.

NORMAL DOS
==========

```
100 FOR N=42534 TO 42589:READ A:POKE N,A:NEXT N

1000  DATA
32,94,166,176,78,169,6,141,82,170,32,140,166,208,15,32,252,162,169,3,205,82,
170,240,206,169,5,76,210,166,201,224,144,2,41,127,141,92,170,174,90,170,240,
9,202,189,0,2,9,128,157,0,2,76,179,159
```

IMPROVED DOS
============

```
100 FOR N=42534 TO 42589:READ A:POKE N,A:NEXT N

1000  DATA
32,94,166,176,78,169,6,141,82,170,32,140,166,208,15,174,90,170,224,2,208,3,1
69,141,44,169,255,234,234,234,201,224,144,2,41,127,141,92,170,174,90,170,240
,9,202,189,0,2,9,128,157,0,2,76,179,159
```

WORDSTAR

Do your written presentations look quality ?

DO YOUR BASIC

ARE YOU FED UPP WITH SPEELING

MISTOKS AND GRUBBY, ##IPPEXED

TYPPING ??

YOU NEEED THE BAASUG

WRODSTAR COURSE !!

The first BASUG Wordstar course will be held in Bracknell on the 8th October 1983. There are some places left, but HURRY. No experience or technical knowledge knowledge is needed. For further details, and to reserve your place, ring Fran on 0727-67450. The cost is £17.50 to members, £25.00 to non-members.

Readers' Letters

/Ed. -Here is the reply to Jeremy Lee-Potter's letter published in the August edition of Hardcore./

Apple Computer (UK) Ltd.
Hemel Hempstead, Herts

Dear Sir,

A lower quality black and white output is usually due to the 'colour killer' switch being in the incorrect position. All Apple //e's have this switch, although on later models it has moved to the right hand edge of the motherboard. The computer will produce black and white when this switch is in the 'On' position. A quick check is to load the 'Apple presents Apple' disk and observe the screen whilst moving the switch: May I also suggest a check of the monitor cable to see if this is of the correct type? Some coax cables are only for use with VHF signals and this attenuates the video band causing the over use of contrast.

The character change on the 7x5 dot pattern was to enable the use of descenders in lower case, a situation not encountered on the Apple][+ of course.

World-wide sales of Apple][+ and //e have increased dramatically over the last year causing great strain on the production of things like the Monitor ///. Apple are trying to resolve this problem by early Autumn. As you can appreciate few monitor companies are in the position to produce the quantities we require to the quality standards we insist on.

In conclusion I am surprised that you find the Apple //e video inferior and hope this is due to the switch position. In the two and a half years it took to develop, the video output section was one of the improvements we sought. Tidying of the signal quality on-board was our main objective to achieve this.

Yours faithfully,

Stephen Holmes
Apple][Product Manager

Finchley, London

Dear Sir,

G. H. Davies, in the June edition of Hardcore, enquires about the hi-res graphics mappings. I shall try to answer his question.

The Apple hi-res screen is split into 192 rows of 280 dots. Each row of 280 dots is stored in forty bytes. Each group of seven dots belongs to one byte, the remaining bit (the eighth) in each byte being used for colour description. Each byte is represented back-to-front on the screen i.e. the least significant bit is on the left of the seven bit group instead of being on the right as is the usual convention. As you move from left to right, increasing the X co-ordinate, you move from bit to bit going in increasing significance until the next byte in memory is accessed.

The ITT, though, has 192 rows of 380 dots: There are still forty bytes to each row, but each byte now has nine bits, all visible on the screen. The reverse mapping system is still used (i.e. lsb on the left).

So the X co-ordinate can be converted into an offset address and a bit number by the following simple calculation:

```
for Apple:  OFFSET = 280 DIV 7
            (integer divide)
            BITNUM = 280 MOD 7

for ITT:    OFFSET = 360 DIV 9
            BITNUM = 360 MOD 9
```

The Y co-ordinate is not so easy. Both machines use the same mapping technique. Try this BASIC program:

```
10 HGR : FOR X = 8192 TO 16383 :
   POKE X,255 : NEXT
```

This fills hi-res page 1 in logical sequence of bytes. You can see the extraordinary way the memory has been mapped into the graphic

page. There seems no logical reason why this method was chosen. It is made worse by the fact that at the end of each row there are two bytes which are not visible on the screen and so are not used. This is evident from the following proof:

Each page is 8k long = 8192 bytes.

Each row on Apple and ITT contains 40 bytes.

Each page contains 192 rows.

=> Number of visible bytes per screen = 40 x 192 = 7680

-> 8192 - 7680 = 512 bytes are unused.

=> 512 / 192 = 2.667 = 2 bytes per row are wasted.

In fact, we do not have to bother about the technicalities of the relationship of the Y co-ordinate and the left-hand base screen address since both Applesoft and Palsoft ROMs provide a machine code routine which calculates the left-hand base address for the Y ranges 0 to 191. These can be used as follows:

```
For Apple: LDA YCOORD
           JSR $F411

For ITT:   LDA YCOORD
           JSR $F458
```

The base address is returned in locations $26 and $27. The page number ($20 = page 1, $40 = page 2) must be placed in location $E6.

So, the actual location of the byte to be chosen can now be calculated:

add the offset calculated first to the base address in locations $26 and $27 just calculated.

The bit position can, for the Apple, be easily calculated:

```
VALUE = 2
```

Thus 'VALUE' will now be a number which can be directly inserted into the location. Of course, in assembler, this value can be calculated as follows:

```
      LDX VALUE
      LDA #0
      SEC
SHIFT ROL
      DEX
      BPL SHIFT
```

The resulting value can be stored in the calculated address or it might be used as a mask for a logical operation to achieve EOR, AND or OR as well as REPLACE.

The ITT, because of its extra bits, uses a slightly more complicated system. The first eight bits of each byte can be addressed in the same way as the Apple, using the above method; but the ninth bits cannot.

To write to the ninth bits you actually write to the eighth bit of the specified location as follows:

```
LDY #0
LDA (HPTR),Y ;GET GRAPHICS BYTE
               POINTED TO BY HPTR
PHA
ORA #$80     ;SET EIGHTH BIT
STA $C05E    ;ENABLE NINTH BITS
STA (HPTR),Y ;SET EIGHTH AND NINTH
               BITS
STA $C05F    ;DISABLE NINTH BITS
PLA
STA (HPTR),Y ;RESET REST OF BYTE TO
               ITS OLD VALUE
```

The above routine will set the ninth bit of a location pointed to by 'HPTR' without changing any other bits. The extra processing required to handle the ninth bits is what makes ITT graphics that much slower than the Apple's.

To read the value of the ninth bits there is a simpler way:

```
LDA (HPTR),Y ;SELECT WHICH BYTE TO
               READ
LDA $C063    ;GET SELECTED NINTH BIT
BMI BITON    ;IT'S ON
BPL BITOFF   ;IT'S OFF
```

The ninth bit switches used by the ITT are, in fact, the Apple switch 3 and paddle 3 locations. Hence the ITT has only three paddles and two switches.

I hope that the above throws some light on the Apple and ITT graphics.

Yours,

Dave Miller

Accra, Ghana

Dear Sir,

The BOLDPRINT routine on library disk 72 is a useful extension to the limited facilities of the Silentype printer.

Switching it on from within a BASIC program with a DOS BLOAD and a CALL 769 is no problem. But can anyone tell me how to switch it off?

Yours sincerely,

Manu Herbstein

Sample printout:

NORMAL SILENTYPE PRINTOUT.

THIS IS BOLDPRINT.

Sauchie, Clacks.

Dear Mr. Panks,

I have not yet got an Apple, but intend to buy a //e later this year, so I wonder if you could answer one or two questions for me?

1. Are there any Apple User Groups near me? I live near Stirling which is roughly midway between Edinburgh and Glasgow.

2. Does the Apple //e suffer from overheating if it has several add-ons like the Z80 card, extra memory, Accelerator, etc? If so, would the Monitor Stand just not make things worse?

3. Can you please recommend a magazine which is devoted entirely, or mainly, to the Apple, and how would I get it?

Yours sincerely,

J. Howard Wright

/Ed. -1. We don't seem to have any local group near you on our records, but I have put your address in our lonely Apples column.
2. We haven't heard any complaints about overheating. The most power-hungry boards on the][have been made less so on the //e and there are fewer chips, so it shouldn't be a problem. If anyone knows to the contrary, please let us know.
3. See the article by Keith Chamberlain in this issue./

East Hendred, Oxon.

A new boy seeks help.!!

As a member for 2 months may I make a suggestion that has possibly been aired before. Is it possible for new members to get hold of back copies of Hardcore for a fee. Maybe someone has a cumulative index or could master disks for each issue be put in the library. On the same theme, is there any place where one can find out more about the various programs listed in the Software Library other than suck it and see?

I hesitate to send this on a disk since I note that in the current Hardcore you ask for comments via Applewriter I. This is II.

John Stevenson.

/Ed. -Back copies are available from the administrator at the cover price of £1. We are considering doing a special offer for a batch of back issues, so watch this space! It wouldn't be worth putting the issues on disk as the disk would cost more than the back issue. Also the files are very specific to my printer. As to the software library, yes, for that price you must expect to take a chance on whether the disk is what you want. However, disks are frequently mentioned in Hardcore with comments, so perhaps the back issues might help you. I believe Jim Panks is currently working on a new catalogue.

It's sensible!
It's compact!
It's easy on the eye!
It's *inexpensive!*

THE ADVANCED DESK-TOP PRINTER STAND

Patent Pending.

Avoids all the problems normally associated with printer paper management

- Stands neatly on a desk top
- Printed paper stacks neatly in the tray
- Supply paper locates underneath – can conveniently remain in its carton
- Sturdy design – finished in brandy white
- Electric cables run underneath – out of harm's way

Accommodates most makes of 80-character printers including:
- **EPSON MX80**
- **MICROLINE 80**

ORDER DIRECT FROM ADVANCED RESOURCES ON THE ORDER FORM BELOW.

Discounts available. Please enquire for details.

Advanced Resources reserve the right to improve the design without notice.

To: Advanced Resources, St. Gabriels, Much Birch, Hereford HR2 8HY.
Telephone: (0981) 540 262

Please supply the ordered quantity of Advanced Printer Stands at a price each of £28.50 which includes VAT and delivery (UK mainland only).

Order No.	Quantity	Price each	Total

Cheque enclosed for £ _____

My make and model of printer is: _____

Name _____
Position _____
Organisation _____
Address _____

Post Code _____
Telephone Number _____
Address for delivery (if different from above)

Name _____
Address _____

Post Code _____

Applewriter II files are text files and exactly what we like. In any case, we are prepared to accept any type of file as there is usually someone who can read it for us./

Liverpool 16.

Dear Sirs,

I was most interested in your Update No. 12 that BASUG is venturing North of Watford and coming to Liverpool. Please could you forward details.

Whilst writing, I would be pleased if you could answer how FRE(0) clears the memory of 'unwanted strings', how FRE(0) is used in garbage collection. I have found a time lag whilst this function is used, can vouch that my 'wanted strings' remain intact. However, I am still mystified.

Belatedly, may I congratulate you on the Machine Code Course I attended in December last which I thought was first class.

Thanking you for an informative magazine.

Yours sincerely,

Paul Rubin.

/Ed. -Check with the diary for details of the Liverpool workshop.

Would anyone like to explain FRE(0)? You might find Richard Teed's article on strings in this issue of interest in the meantime./

Belfast.

Dear Sir,

In the August edition of Hardcore Dougal Hendry mentioned shiftkey bounce with the Vision-80 card. Until then I had thought that there was a fault with my Apple II+. The dealers who supplied the Vision-80 card even went so far as to replace my keyboard in an attempt to cure the fault but it returned within a week. Now I find that it occurs with other Vision-80 cards!

When corresponding with Pynwon about configuring the card for Wordstar, I mentioned this problem, but all they said was that they had no problems with theirs, so implying that the fault must be with my machine.

Does anyone know how I can fix the shiftkey bounce, and also is there any way of adjusting the card so that the bottom line does not disappear off the bottom of the screen (without having to adjust the monitor)?

Yours sincerely,

B. P. Cooper.

Sunbury on Thames.

Dear 'Sir',

In the August 'Epson Pages', Norah Arnold refers to Mike Glover and Trevor Roper's 'bypass code' on page 57 of June 'Windfall', pointing out that the version given for the Grappler works just as well for the Digitek Printmaster.

May I point out that the version given for the Epson 8132 card will work just as well for the Apple and ITT Parallel Printer (and for all I know, other) cards? What it does is to store the character to be sent in $C090 and check the printer 'busy' signal in $C1C1, exactly as these cards do. The firmware on the cards is what is bypassed, so the fact that the Epson firmware differs from the Apple's is irrelevant!

Yours faithfully,

R. Teale.

Kingswood, Surrey.

Dear Sirs,

Re Program Difficulty on ITT 2020

I understand that you may be able to help me on a particular 'bug' in my program. This is the final problem on a program which has taken me two years to complete.

I am using the HGR2 graphics mode to plot sales figures and a line of least squares regression. However, to give the axis any meaning, I would obviously like to label

them (both X and Y axis). I have consulted the 'Applesoft II Basic Programming Reference Manual' which for my purposes suggests the two following POKE commands:-

POKE -16298,0 High Resolution Graphics 2 to Low Resolution Graphics 2, whilst not clearing the screen.

POKE -16300,0 Low Resolution Graphics 2 to Low Resolution Graphics 1, whilst not clearing the screen.

All that is displayed are a number of lines horizontally across the screen, together with 'blobs' where the narrative should be. It seems that I need either another POKE command to go from Low Resolution Graphics 1 to Text Page 1 without clearing the screen, or a totally different approach.

I have the original type of ITT 2020, but it has been uprated to DOS 3.3, it has 48k of memory with two disk drives.

I attach a copy of the relevany part of my program annotated as necessary.

Many thanks in anticipation of your help.

Yours faithfully,

G. L. Taylor.

/Ed. -If anyone can help, get in touch with me and I will pass on the listing./

Thank you to Ian Shipley of London WC2 and David Wilshere of Blonay, Switzerland for your letters on the Epson and Applewriter. I have passed them on to Jim Panks who will hopefully have replies or an appropriate article for the next issue.

St. Mary Cray, Kent.

Dear Tony,

I've noticed the odd 'small ad' in past issues of Hardcore. Is it possible for the magazine to run 'box numbers' for replies? This might sound silly for a publication like Hardcore, but I know that many people are very sensitive about having their address published as it can attract burglars.

On a wholly separate note, I have a huge collection of Eamon disks, which are all public domain and I'd be happy to arrange to copy them at cost.

Finally, I've heard nothing from Micronet since first registering my interest about a year ago. Has anyone at BASUG had greater luck getting a modem link into Prestel? I'm beginning to wonder whether Micronet have lost me from their records. If anyone can help with a contact address & phone number for them I'd be grateful as I'd certainly like to chase them up.

Always looking forward to the next Hardcore.

Yours sincerely,

Selwyn Ward.

/Ed. -Small ads are free to members. BASUG does not have the resources or the administrative time to run box numbers. If you do not wish to have your address published, then just quote a phone number.

Thank you for your offer of Eamon disks. I have passed the list on to Jim Panks as I feel it might be more sensible for him to arrange to have them copied once and then add them to the software library.

On the last point, I suggest you read Quentin's article in this issue. The address you wanted is:

Micronet 800
Telemap Ltd
Bushfield House
Orton Centre
Peterborough PE2 0UW
Tel:(0733) 237111./

Hampton, Middx.

Dear Mr. Panks,

I enclose herewith a disk with a machine code version of 'Haunted Caves'. I had been translating the game from Basic into binary for a bit of intellectual amusement, when I saw, in the August copy of Hardcore, Philip Colmer's comment on the lack of a 'Help' facility. I took the opportunity to write one in, and hope that the resulting game may be of some interest to other members.

Yours sincerely,

W. Anderton.

London E13.

Dear Yvette,

I have just solved the ITT/Pascal Upper/Lower case problem, as mentioned by Mike Siggins in Hardcore, August '83. Chances are that everyone already knows about it, but just in case - here goes.

Reading through back issues of 'Windfall', I found in the January '83 issue, a note in Appletips (by J.P.Lewis) on implementing an adapted monitor to enable a shift-key mod.

This made it clear that the shift key routine in SYSTEM.APPLE is located at the paddle button 3 location ($C063). A search of the ITT2020 manual elicited the fact that this location is used for the hi-res graphics and is hence not available. The solution seemed to be to point this routine at something more innocuous. The cassette-in location at $C060 presented itself as a suitable candidate, and the following program (adapted from that in the above article) has solved the problem for me. (Incidentally, other keys are now in the right place ([@,[,]]).

```
program SHIFT;

(* Resets the shift vector at $C063 in the
file SYSTEM.APPLE to $C060 for use with the
ITT2020 where $C063 is not available *)

var
   BUF:packed array[0..31,0..511] of 0..255;
   F:file;
   I:integer;

begin
   reset(F,'SYSTEM.APPLE');
   I := blockread(F,BUF,32);
   close(F);
   BUF[4,61] := 96; (* To get BIT $C060. *)
   reset(F,'SYSTEM.APPLE');
   I := blockwrite(F,BUF,32);
   close(F);
end.
```

Hope that this is of some use.

Yours sincerely,

Andy Holderness.

ADVERTISERS

Advertising rates

Full page	£50.00
Half page	£27.50
1/4 page	£17.50
Flysheets	£75.00

Preparation of artwork from £ 5.00

Copy Dates

Date	Edition
October 28th	December
January 6th	February
March 2nd	April
May 4th	June
July 6th	August

Please send complete camera-ready artwork in monochrome. If the original is in A4, then the typeface must stand photographic reduction to A5. We can undertake minor alterations to copy.

DIARY

October

4th	Herts Group - Communications and Prestel
8th	BASUG General Meeting in Liverpool (see update or ring John Rogers on Harpenden 5100). Subject Communications.
8th	Wordstar Course
10th	Hants and Berks Group
	Croydon Group
11th	East Kent Group, Margate
13th	South West London Group - Technical Software
15th	Advanced Visicalc Course

November

1st	Herts Group
8th	East Kent Group, Margate
10th	South West London Group - Peeks, Pokes & Calls
12th	BASUG National Meeting in Maidstone,Kent.
14th	Hants and Berks Group
	Croydon Group
24th-26th	North West Computer Show, Belle Vue, Manchester
27th	Visicalc Course in Manchester

December

6th	Herts Group
8th	South West London Group - Games
10th	BASUG National Meeting
12th	Hants and Berks Group
	Croydon Group
13th	East Kent Group, Margate

If you would like your events in the diary, please write in and tell us about them.

ELITE SOFTWARE COMPANY.

S-C Macro Assembler: Full macro assembler with 100 page manual. Operates in any II or //e. Source files may be in core or on disc. Object files may be assembled directly to memory, or to disc. Can read or write standard text files. Editor and assembler are co-resident, all Apple DOS and monitor commands are available.
Price £60.00 + VAT.

Appli-Kit: Applesoft utility. **Appli-Kit** is a package designed to aid the writing and debugging of Applesoft Basic programs. All commands are typed in as if they are part of the standard Applesoft set.
Price £25.00 + VAT.

Wildcard: plugs into any slot and copies the program in core onto a standard DOS disc. Copies 48K core resident programs. No parameter list needed. Copies most programs requiring RAM card. Copies made with the **Wildcard** can be copied with standard copy programs. Copies even the bit copiers. Works with any RAM card - plugs into any slot.
Price £99.00 + VAT. (Now works with //e.)

Wild-Word: a 40 column word processor for the II or //e. True SHIFT key operation. Built-in mailing list and merge. Core-resident.
Price £35.00 + VAT.

Format-80: 80 column word processor for II and //e. What you see is what you get. Editing and formatting achieved with single keystrokes. Supports all Apple compatible printers. Mailing list and merge feature. Core-resident.
Price £129.00 + VAT.

Ramview: 80 column board for the //e, upgradeable to 64K.
Price £80.00 + VAT.

All Elite Software Company products are <u>not</u> copy protected.

Elite Software Company, 2 Almorah Road, Heston, Middlesex, TW5 9AD. Tel. 01-572-0453

SAVES TIME

Imagine the time, energy, and frustration you could save by boosting your Apple's speed from 1 Mhz to 3.58 Mhz. That's 3½ times faster than normal, making the Apple II Plus arguably the fastest Micro on the market.

How is it possible? It's all down to ACCELERATOR II. This new plug-in board from Pete & Pam Computers contains a 6502C Processor and 64K of memory. The board runs all native Apple II software, including programs written in Applesoft, Integer, Machine Code, Pascal, Apple Fortran 77 and Forth.

Amongst the many thousands who could benefit from ACCELERATOR II are users of Visicalc, DB Master, Micro Modeller, Multiplan Tabs, and Systematics.

SUPER FAST

In November 1982, PCW published a bumper round up of all the Benchmark Timings since PCW began. The Olivetti M20 came out top of the 'league' with an average Benchmark timing of 11.5. Running the same Benchmark test programs, the Apple II Plus with Accelerator II averages a timing of 8.58 – that's an incredible 25% faster than the Olivetti M20.

We have reproduced some of PCW's findings, incorporating Benchmark Timings for the Apple II Plus with Accelerator II.

Machine	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8	Average
Apple II Plus with Accelerator II	0.3	2.4	4.5	5.0	5.5	8.2	12.9	2.98	8.6
Olivetti M20	1.3	4.0	8.1	8.5	9.6	17.4	26.7	1.6	11.5
IBM Personal Computer	1.5	5.2	12.1	12.6	13.6	23.5	37.4	3.5	17.6
Osborne 01	1.4	4.4	11.7	11.6	12.3	21.9	34.9	6.1	19.9
Intertec Superbrain	1.6	5.2	14.0	13.9	14.8	26.3	43.2	5.6	21.9
Apple III	1.7	7.2	13.5	14.5	16.0	27.0	42.5	7.5	24.7
ACT Sirius 1	2.0	7.4	17.0	17.5	19.8	35.4	55.9	4.3	24.8
Xerox 820	1.7	5.5	15.5	15.1	16.2	28.9	46.1	8.0	26.1
Apple II	1.3	8.5	16.0	17.8	19.1	28.6	44.8	10.7	30.4
Commodore CBM 8032	1.7	10.0	18.4	20.3	21.9	32.4	51.0	11.9	34.3

So don't wait – start to save time now. Contact your local dealer, or call us on (0706) 212321, or, in London on 01-769 1022.
A faster, easier computing life is on its way.

– AVAILABLE NOW FROM –

Pete & Pam Computers

Head Office:
NEW HALL HEY ROAD,
Rossendale, Lancs. BB4 6JG
Tel: Ross (0706) 212321 & 227011
Telex: 635740 PETPAM G

London Office:
103-5 BLEGBOROUGH RD,
London, SW16 6DL
Tel: 01-769 1022/3/4
Telex: 923070 PPCOMP G