

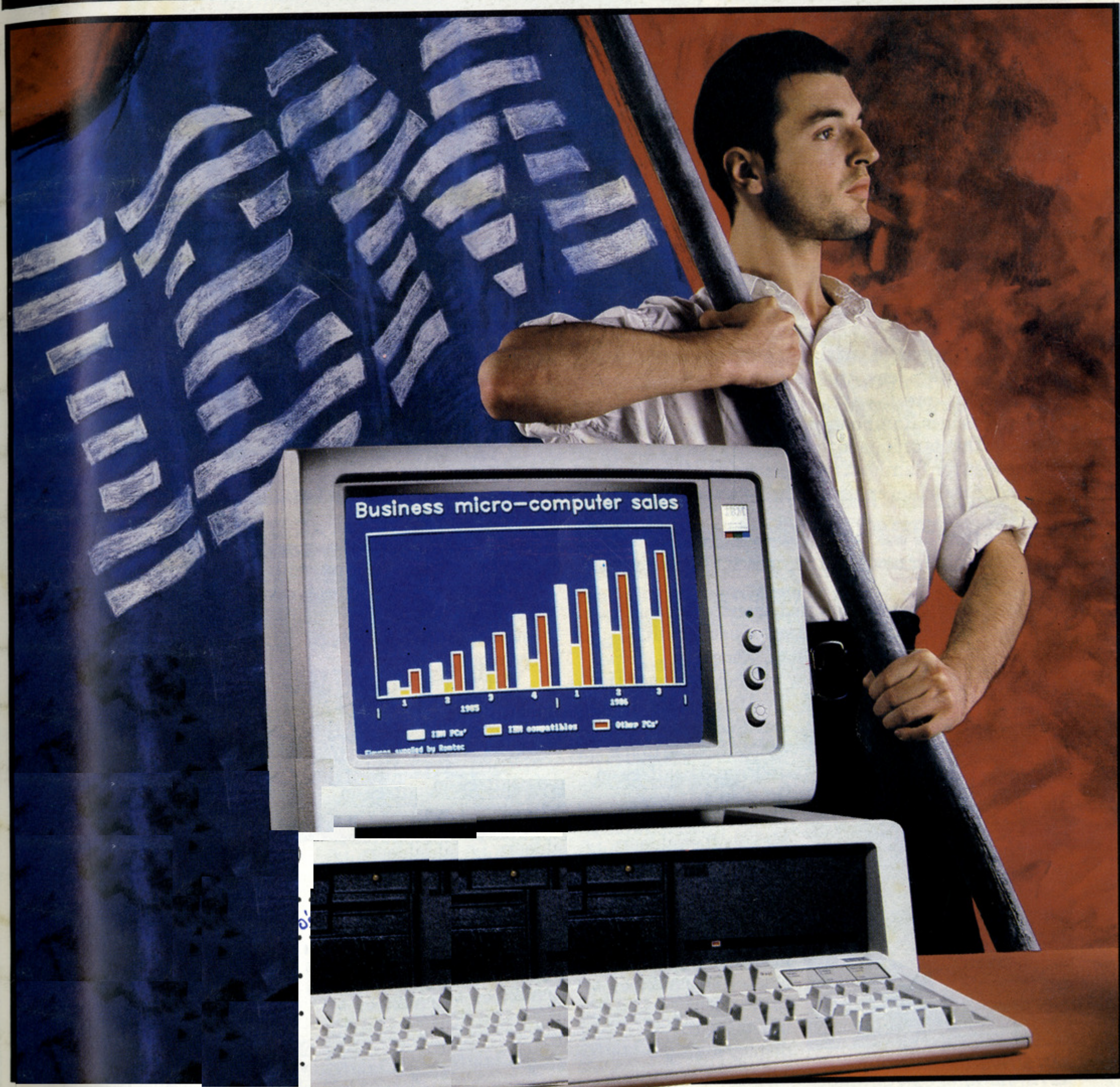
# Australian Personal Computer

ISSN 0725-4115 NZ \$3.50  
REGISTERED BY AUSTRALIA POST PUBLICATIONS VBP 3691

JANUARY 1987 \$3.50

RUN  
MACINTOSH  
SOFTWARE ON  
AN ATARI!

AUSTRALIA'S TOP SELLING COMPUTER MAGAZINE



**POWER FOR THE PEOPLE**  
**IBM's new standard PC—the XT286**

RETURN

Remember:

```
Transfer whole of display window into screen% array
GET (0,0)-(486,322),screen%
Ask user for a file name. Save parameters followed by screen% array
map$=FILES$(0,"Save map as ...")
IF map$="" THEN PUT (0,0)-(486,322),screen%,PSET: RETURN
```

```
OPEN map$ FOR OUTPUT AS #1 LEN=1000
WRITE#1,map%,dimfix%,plot%,level%
WRITE#1,xlo,xhi,ylo,yhi,c,d
WRITE#1,xbase%,ybase%,offset%,extent%,square%,size%,corner%
FOR i%=0 TO 10015
  WRITE#1,screen%(i%)
NEXT
CLOSE #1
END
```

RETURN

Finished:

```
MENU 6,0,1
MENU ON
menunumber%=MENU(0)
WHILE menunumber%=0: WEND
```

RETURN

- a Temporary variable holding Real part of z during iteration
- action% Number returned by Dialog function showing type of last user action
- again% Flag - set to 1 if this is modified plot (initialisation already done)
- b Temporary variable holding Imaginary part of z during iteration
- box% Number of box containing buttons/edit fields currently selected in Parameter window
- c Real part of constant k
- click% Number returned by Mouse function indicating if mouse button pressed
- colour% 30=white, 33=black
- corner% Co-ordinate of centre pixel in current top-left square
- correction Correction factor to adjust zoom rectangle to screen format
- d Imaginary part of constant k
- dimfix% Dimension to be entered by user - 3=Real, 4=Imaginary
- done% Flag - set to 1 when OK button is pressed in Parameter window
- edfield% Currently selected edit field in the Parameter window
- extent% Half-size of current tiles
- g Step in value between adjacent pixels
- hibutton% Number of highest button in current group
- hold% Number returned by Mouse function to show state of mouse button
- i% General purpose loop counter
- level% Maximum number of iterations to be done
- lobutton% Number of lowest button in current group
- m Temporary variable holding square of Real part of z during iteration
- map% Type of plot - 1=Mandelbrot set, 2=Julia set
- map\$ Filename for saved plot
- menuitem% Number of item in menu selection

- menunumber% Number of menu bar selection
- n Temporary variable holding square of Imaginary part of z during iteration
- offset% Distance from centre pixel of current square to centre pixel of current tile
- oldmap% Flag - set to 1 if this is a partially completed plot loaded from disk
- plot% Plotting mode - 5=odd/even, 6=max value
- pointx% X co-ordinate where mouse button was clicked
- pointy% Y co-ordinate where mouse button was clicked
- press% Number of button last pressed
- r% Array holding co-ordinates of rectangle dragged with mouse
- screen% Array to hold whole of screen when saving/loading from disk
- size% Size of current squares
- small% Array to hold part of screen obscured by small window
- square% Size of current tiles (= 2\*extent%)
- x% X co-ordinate of centre pixel in current tile
- xbase% X co-ordinate of centre pixel in current square
- xhi High value of Real dimension displayed
- xlo Low value of Real dimension displayed
- xmid Value of Real dimension at centre of zoom rectangle
- xrange Extent of Real dimension of zoom rectangle
- xstart% X co-ordinate of centre pixel in square where plot was interrupted to save to disk
- y% Y co-ordinate of centre pixel in current tile
- ybase% Y co-ordinate of centre pixel in current square
- yhi High value of Imaginary dimension displayed
- ylo Low value of Imaginary dimension displayed
- ymid Value of Imaginary dimension at centre of zoom rectangle
- yrange Extent of Imaginary dimension of zoom rectangle
- ystart% Y co-ordinate of centre pixel in square where plot was interrupted to save to disk
- zimag Temporary variable holding new value of Real part of z
- zoomscale Ratio of Real:Imaginary extent of zoom rectangle
- zreal Temporary variable holding new value of Imaginary part of z



## Amstrad CPC Trace in Windows

by Jeffrey Weaver

This program allows the user to display the line numbers produced by the tracer to be printed in certain text windows or dumped to a printer. When the Control and Tab keys are pressed, the program being executed is halted until the keys are released.

To use the program, type it in as listed and save it; then run it. The program will set HIMEM and then POKE-in the machine code. It will ask

for a stream number in the range 0-8. 0-7 are text windows and 8 is the printer. The computer then asks for the size of the window and its position. When the window size has been entered, the Basic program deletes itself and the computer can be used as normal; but when TRON or TROFF are typed, they act on the new trace program.

```

10 PAPER 0:PEN 1:MODE 1:INK 0,0:BORDER 0
20 MEMORY 39999:RESTORE:FOR F=40000 TO 40255:READ A$:POKE F,VAL("&"+A$):NEXT F
30 PRINT "PLEASE SELECT WINDOW NUMBER (0-8)":PRINT:PRINT"0-7 ARE TEXT WINDOWS AN
D 8 SENDS THE":PRINT:PRINT"LINE NUMBERS TO THE PRINTER.":PRINT:INPUT "NUMBER ":S
T
40 IF ST<0 OR ST>8 THEN ST=8
50 PRINT:INPUT "LEFT COLUMN OF WINDOW ":L
60 PRINT:INPUT "RIGHT COLUMN OF WINDOW ":R
70 PRINT:INPUT "TOP ROW OF WINDOW ":T
80 PRINT:INPUT "LOWER ROW OF WINDOW ":W
90 POKE 40248,ST:POKE 40189,L:POKE 40191,R:POKE 40193,T:POKE 40195,W
100 CLS:CALL 40000:PRINT "NEW TRACE ROUTINE IS NOW READY.":NEW
110 DATA 3E,C3,21,4C,9C,32,5A,BB,22,5B,BB,C9,CD, 8F,9C,CD,9E,9C,CA,B1,9C,3A,3D,9
D,FE,0,C2,BE,9C,3A,31,9D,FE,5B,C2,B1,9C,3A,38,9D,FE,8,D2,87,9C,3A,3A,9D,4F,CD,11
,BC,B9,C4,F5,9C,CD,7B,BB,22,3B,9D,3A,38,9D,CD,B4,BB,32,39,9D,3E,1,32,3D,9D,C3,A4
,9C,32,31
120 DATA 9D,ED,43,32,9D,ED,53,34,9D,22,36,9D,C9,3A,38,AE,FE,0,C9,3A,31,9D,ED,32,
9D,ED,34,9D,2A,36,9D,C9,3A,31,9D,CD,BA,9C,C3,A4,9C,CF,0,94,C9,3A,38,9D,FE,8,D2,1
2,9D,3A,31,9D,FE,5D,C2,B1,9C,3E,20,32,31,9D,CD, B1,9C,3A,39,9D,CD,B4,BB,2A,3B,9D
,CD,75
130 DATA BB,3E,0,32,3D,9D,3E,44,CD,1E,BB,28,4,CB,79,20,F5,C3,A4,9C,3A,38,9D,CD,B
4,BB,F5, 26,1,16,14,2E,1,1E,3,CD,66,BB,F1,CD,B4, BB,CD,11,BC,32,3A,9D,C9,3A,31,9
D,FE,5D, 28,B,3A,31,9D,CD,2B,BD,30,F8,C3,A4,9C,3E,20,CD,2B,BD,3E,0,32,3D,9D,C3,A
4,9C,0,0
140 DATA 0,0,0,0,0,7,0,3,0,0,0,0,0

```



## BBC Interrupt-Driven Clock

**MICROTEX  
666**

by Alan Potton

This program is available electronically through Microtex 666's software downloading service. It is accessed through Viatel page \*66637#

This program provides a real-time clock and alarm facility on the BBC Micro. The program should be typed-in as it stands, substituting all pound signs (£) for hashes (#). There are four possible alterations. The user could change the loop in line 1040 to run from 1 to 3 rather than 0 to 2, in order to check for errors in the assembly language the first time the program is run, and then change it back when all is OK. Secondly, the machine code can be relocated by changing the value of P% in line 1050 and the two old values in line 2740. The current setting of &1700 is for use with a disk system; for a cassette system, use &B00. Thirdly, the coordinates on the screen where the time is to be printed may be altered in line 1350. Finally, if the user has other events enabled, remove the two slashes in line 1880.

When the program has been typed-in, save it then run it. The machine code will be installed and a prompt for the time will be given: it should be entered in the format

hh:mm. Then the alarm time is requested and given in the same format. The Basic program is erased using the NEW command. There are three \* commands to use the new clock:

\*ZINITCL — this *must* be entered to initialise and start the clock running. It need only be typed once, and to ensure correctness, must be typed at the time entered at the prompt.

\*ZDISP ON/OFF — this command switches the display of the clock onscreen on and off.

The default is off.

\*ZALM ON/OFF — this command switches the alarm on and off. If it's on and the alarm time is reached, it will ring. When off, the alarm will not sound.

The program is made up of two parts of machine code and one of Basic. The first machine code routine in lines 1080 to 1840 adds three new OS commands. Lines 1080 to 1190 provide the new service routine to access the new commands.

The clock works using the count-

down timer (using the OSWORD routine with A=4) on a one-minute cycle. When the interrupt signal is produced, control jumps to the User Interrupt Service Routine. The

routine increments the clock, displays the time and checks for the alarm. The address of the routine is given to the operating system in lines 1330 and 1340.

```

1000 MODE3
1010 w=&FFF1
1020 b=&FFF4
1030 a=&FFE3
1040 FOR LX=0 TO 2 STEP 2
1050   P%=&1700
1060   [oot LX
1070
1080   .cli
1090   pha:txa:pha:tva:pha
1100   stx &B0:stx &B1
1110
1120   ldy £1:lda (&B0),Y:ora £32:tax
1130   ldy £2:lda (&B0),Y:ora £32:tav
1140   cpx £ASC"z":bne mark0
1150   cpv £ASC"i":beq zinitcl
1160   cpv £ASC"d":beq zdisp
1170   cpy £ASC"a":beq zalm
1180   .mark0
1190   jmp zdef
1200
1210   .zalm
1220   jmp za
1230
1240   .zinitcl
1250   lda £0:sta &75
1260   lda £0:sta &78
1270   lda £&86:sta &79
1280   lda £&EB:sta &7A
1290   lda £&FF:sta &7B
1300   sta &7C:sta &7D
1310   lda £4:ldx £&79:ldy £0:isr w
1320   lda £14:ldx £5:jsr b
1330   lda £int MOD 256:sta &220
1340   lda £int DIV 256:sta &221
1350   lda,£70:sta &7E:lda £00:sta &7F
1360   lda £01:sta &88
1370   lda £00:sta &89
1380   lda £&F1:sta &8A
1390   lda £&FF:sta &8B
1400   lda £&CB:sta &8C
1410   lda £00:sta &8D
1420   lda £&B0:sta &8E
1430   lda £00:sta &8F
1440   jmp ex
1450
1460   .zdisp
1470   ldy £8:lda (&B0),Y:ora £32
1480   cmp £ASC"n":bne dpoff
1490
1500   .on
1510   lda £1:sta &78
1520   jsr print
1530   jmp ex
1540
1550   .dpoff
1560   lda £0:sta &78
1570   lda £134:jsr b:stx &76:stx &77
1580   lda £31:jsr a
1590   lda &7E:jsr a:lda &7F:jsr a
1600   lda £32:jsr a:jsr a:jsr a:jsr a:jsr a
1610   lda £31:jsr a
1620   lda &76:jsr a
1630   lda &77:jsr a

```