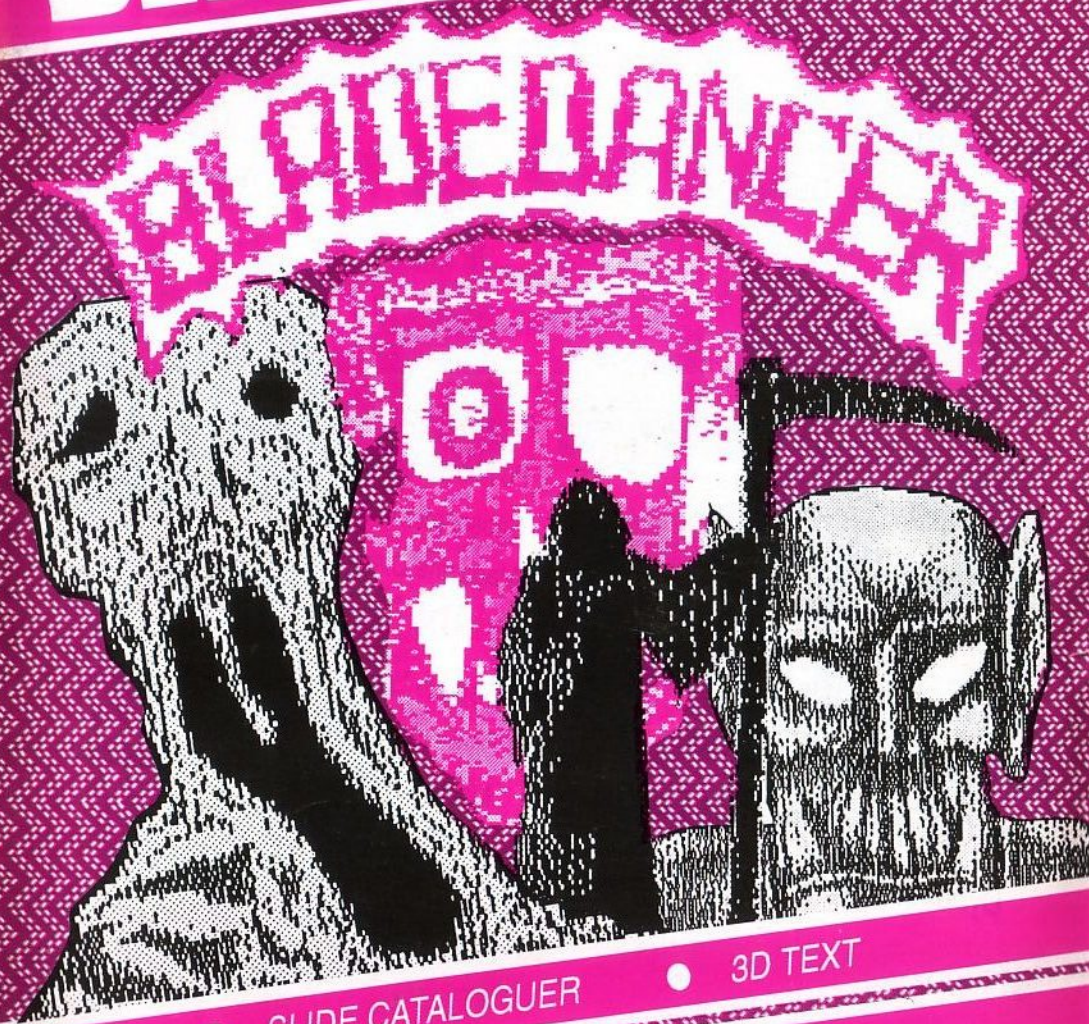


Vol 12 No 1 May 1993

BEEBUG

FOR THE
BBC MICRO &
MASTER SERIES



● SLIDE CATALOGUER

● 3D TEXT

● SOUND

● DECISION TREES

FEATURES

Slide Cataloguer (1)	
Robol - The Screen Designer	
Decision Trees	
3D Text	
BEEBUG Education	
Troubleshooting Guide (5)	
First Course: Sound (1)	
Public Domain Software	
512 Forum	
Gravity and Orbits (3)	
BEEBUG Workshop: Sorting (2)	
Mr Toad's Machine Code Corner	
Dynamic Free Memory Display	

REVIEWS

5	Bladedancer	22
---	-------------	----

REGULAR ITEMS

8		4
11	Editor's Jottings	17
15	Renewal Rates	18
19	RISC User	51
24	Hints and Tips	52
31	Personal Ads	53
34	Postbag	54
36	Subscriptions & Back Issues	55
	Magazine Disc	

HINTS & TIPS

41	Discrete Locks	
44	Bit Fields in BASIC	
46	Master Sideways RAM Banks Selection	
49	Date Checking	

PROGRAM INFORMATION

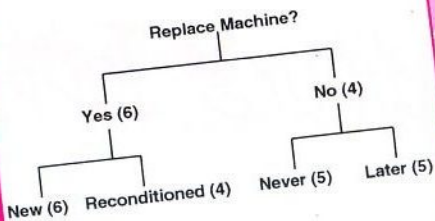
All listings published in BEEBUG magazine are produced directly from working programs. They are formatted using LISTO 1 and WIDTH 40. The space following the line number is to aid readability only, and may be omitted when the program is typed in. However, the rest of each line should be entered exactly as printed, and checked carefully. When entering a listing, pay special attention to the

difference between the digit one and a lower case l (L). Also note that the vertical bar character (Shift \) is reproduced in listings as |.

All programs in BEEBUG magazine will run on any BBC micro with Basic II or later, unless otherwise indicated. Members with Basic I are referred to the article on page 44 of BEEBUG Vol.7 No.2 (reprints



Robot - The Screen Designer



Decision Trees



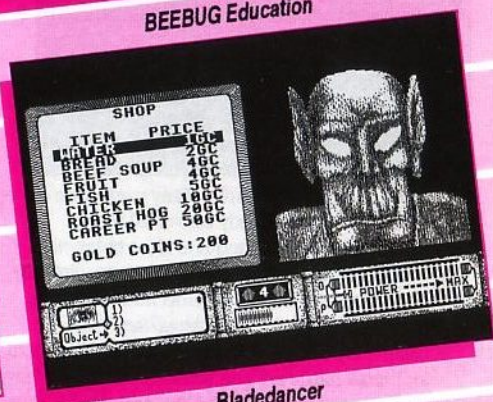
3D Text



BEEBUG Education

Note	Octave						
	1	2	3	4	5	6	7
C		5	53	101	149	197	245
C#		9	57	105	153	201	249
D	13	61	109	157	205	253	
D#	17	65	113	161	209		
E	21	69	117	165	213		
F	25	73	121	169	217		
F#	29	77	125	173	221		
G	33	81	129	177	225		
G#	37	85	133	181	229		
A	41	89	137	185	233		
A#	45	93	141	189	237		
B	1	49	97	145	193	241	

Sound



Bladedancer

available on receipt of an A5 SAE, and are strongly advised to upgrade to Basic II. Any second processor fitted to the computer should be turned off before the programs are run.

Where a program requires a certain configuration, this is indicated by symbols at the beginning of the article (as shown opposite). Any other requirements are referred to explicitly in the text of the article.



Program needs at least one bank of sideways RAM.



Program is for Master 128 and Compact only.

Editor's Jottings

Welcome to the first issue of volume twelve of BEEBUG, the only magazine devoted entirely to the needs of users with an 8-bit Acorn machine - the BBC Micro, the Master 128 and Master Compact. By and large all these systems run versions of the same operating system, and all use the same BBC Basic and 6502 assembler. There are differences, of course, and it is certainly true that some of the information and programs which we publish will be specific to perhaps one of these systems.

We always try to ensure a sensible mixture and balance of material so that each issue, and series of issues, will contain items of interest to all readers. Thus BEEBUG provides what it has always sought to do, to provide a valuable source of information and programs for users of these 8-bit systems.

However, readers will not be unaware that 8-bit technology has now been overtaken by 32-bit developments, of which Acorn's Archimedes range is a prime example. Many users with a BBC Micro have already converted to an Archimedes system, and it is clear today, that with a very few honourable exceptions, virtually all commercial activity is concentrated on the Archimedes range. Other magazines have completely or largely given up on the BBC Micro, and most hardware and software developers have long since concentrated on the Archimedes range for their new products.

This has been reflected in the readership of BEEBUG, which has declined steadily since the heyday of the BBC Micro seven or eight years ago now. We have fewer subscribers than was once the case, but despite this, we have maintained BEEBUG at 64 pages for a considerable time. However, the point has been reached where we

can no longer sustain this size of magazine, and thus with effect from this issue, we will be producing BEEBUG to 56 pages. Do bear in mind that BEEBUG now carries hardly any advertising at all compared with the situation even two and three years ago, and so the reduction in editorial content is less than that.

In addition, as we have done in the past, we will feature items from the early days of BEEBUG in this and future issues. While this will be only a quite small proportion of the magazine, many of the programs published seven, eight and nine years or more ago are often more innovative and well crafted than some of the offerings which we receive today. It really is quite staggering sometimes the quality of program which was achieved in those early days.

There is one further consequence of what I have had to tell you. With the current and projected circulation of BEEBUG, we do not believe that it will be viable to continue with publication beyond April 1994. Of course, matters may change, and we would be among the first to welcome such good news. Therefore all renewals of subscriptions from now on will be calculated on a sliding scale to terminate with the April 1994 issue. Details will feature regularly in BEEBUG - see page 17.

As usual, we are distributing an index to all the issues of Volume 11 with this issue of BEEBUG.

We have been as honest as we can be with regard to the future of BEEBUG, and the market for the BBC Micro. We very much hope that you will stay with us for as long as BEEBUG continues to be published.

M.W.

Slide Cataloguer (Part 1)

Jim Phillips presents the first part of his ADFS slide catalogue

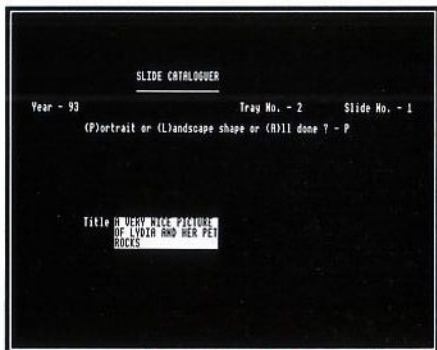
As a keen photographer I finish each year with a large number of slides which should have been sorted and labelled but, because it is a tedious and fiddly task, are still waiting for attention. Not only do I have to decide what each slide is, I also have to write or type a tiny label and then affix the marker to show the top of the slide. This program is of great use in the second part of the operation.

Designed for use in ADFS it uses one directory for each year (hence the need for ADFS) and keeps records in files of 50 entries each to match the GEPE trays which seem to have become the most common storage system. For my own purposes I have divided files into U** for U.K. slides and E** for continental ones. Others are easy to add if needed. The program could be adapted for DFS without much difficulty.

The program is written in Basic and is in two parts. The first, in this issue, creates the data files. The second, next month, displays the data, prints an index, prints the labels on the common 89mm X 36mm fanfold adhesive address labels and catalogues any directory.

Type in the program listed below, taking care with the line numbers that must fit in with the second part of the program, and save it. Now sort your slides as usual and then chain the program. You will be asked for the year and whether Europe or U.K. is required. The program will first check to see if the directory exists and, if not, will create it. It will then go to the directory and scan all the files with either the E or U prefix. It will then open the next in sequence and start the entry routine. First you will be asked if you want any standard phrases and if you do will accept up to 9 of these. These can subsequently be used when entering slide names by pressing f0 to f9 and enable you to enter automatically part of the slide title - very useful if you visit Wales!

The first entry number will now be displayed and you will be asked if the slide is 'Portrait' or 'Landscape' shape or if you have finished. When you answer either of the former a small reversed colour block will appear for you to enter the title. On pressing Return the next entry will be requested and so on until either you press 'A' for all done or you complete 50 labels. If the latter is the case a new file will be opened for the next tray of slides and so on. Pressing 'A' will terminate the process and return you to the menu.



Typing in the label

Later you can add the second part of the program and print your labels but for the moment I suggest you type in the program and try a few entries. To check the results, examine the file with *DUMP but remember the text of each string is in reverse order in the file.

HOW IT WORKS

The program makes extensive use of procedures which are as follows.

PROCsetup defines the address of OSFILE and its parameter block.

PROctitle puts a header on the screen and reduces the height of the text window to leave it on the screen.

Slide Cataloguer

PROCmenu sets the menu on screen and gets your choice.

PROCnew gets the year and file prefix information. It uses *PROCstate(dir\$)* to check if the directory is present and if not uses *CDIR to create it. Then it uses *PROCstate(file\$)* repeatedly to find the first unused filename. It opens the file and uses *PROCadd* repeatedly to enter the data. Finally, it closes the file and returns to the menu.

PROCstate(name\$) sets X% and Y% to the address of the parameter block. It puts the name of the directory or file in this and uses *FNisfile* to return the file state in A%.

FNisfile uses *OSFILE* with A%=5 to read the catalogue information for the file. If the file is not found A% is returned as 0.

PROCadd increments the entry number and prints a header. Checks for 'Portrait', 'Landscape' or 'All done' then uses *PROCgetitle* to get the title string. It repeats until 50 entries have been completed or 'All done' is selected.

PROCgetitle sets the inverse colour window then gets a text string from the keyboard, expands it to 40 characters, prefixes the slide number and suffixes the shape code (P or L).

PROCconv converts the offset part of the file name to a two digit string.

This should be enough detail to allow you to think about other applications for the program. Next month we'll get the labels printed.

```
10 REM Program Slide Catalogue Part 1
20 REM Version M1.0
30 REM Author Jim Phillips
40 REM BEEBUG May 1993
50 REM Program subject to copyright
60 :
100 MODE3:ON ERROR GOTO 180
110 PROCsetup
120 PROCtitle
130 REPEAT
140 PROCmenu
```

```
150 IF ans$="1" PROCnew
160 CLS:OSCLI"DIR $":UNTIL ans$="0"
170 VDU26:END
180 REPORT:PRINT" at line ";ERL
190 *CLOSE
200 *DIR $
210 MODE3:END
220 :
1000 DEFPROCsetup
1010 osfile=&FFDD:param%=&600
1020 ENDPROC
1030 :
1040 DEFPROCTitle
1050 CLS:PRINTTAB(25,0)"SLIDE CATALOGUE
R"
1060 PRINTTAB(25,1)" _____"
1070 VDU28,0,24,79,3
1080 ENDPROC
1090 :
1100 DEFPROCmenu
1110 PRINTTAB(22)"0 - Quit"
1120 PRINT
1130 PRINTTAB(22)"1 - Create new file"
1240 PRINT
1250 PRINTTAB(25)"Enter choice"
1260 REPEAT:ans$=GET$:UNTIL INSTR("01",
ans$)
1270 CLS
1280 ENDPROC
1290 :
1300 DEFPROCnew
1310 PRINTTAB(10)"what year?";:INPUTTAB
(25)"dir$
1320 PRINT
1330 PRINTTAB(10)"(E)urope or (U).K ?"
1340 REPEAT:pl$=GET$:UNTIL INSTR("EeUu",
pl$)
1350 PROCstate(dir$)
1360 IF A%=0 OSCLI("CDIR "+dir$)
1370 OSCLI"DIR "+dir$
1380 M%=0
1390 REPEAT:M%=M%+1
1400 PROCconv
1410 pl$=CHR$(ASCpl$ AND &5F)
1420 file$=pl$+offset$
1430 PROCstate(file$)
```

```

1440 UNTIL A%=0
1450 CLS
1460 N%=0:M%=VAL(RIGHT$(file$,2))-1
1470 REPEAT:M%=M%+1:PROCKey:PROCconv:file$=pl$+offset$:ch%=OPENOUT(file$)
1480 PROCadd
1490 CLOSE#ch%:N%=0
1500 IF shape$<>"A" PROCmore
1510 UNTIL shape$="A"
1520 ENDPROC
1530 :
1540 DEFPROCsetshape
1550 shape$=CHR$(ASCshape$ AND &5F)
1560 ENDPROC
1570 :
1580 DEFPROCKey:CLS
1590 PRINTTAB(10)"Preset phrases (Y/N)?"
"
1600 REPEAT:ans$=GET$:UNTIL INSTR("YyNn",ans$)
1610 IF INSTR("Nn",ans$) ENDPROC
1620 FOR G%=0 TO 9:OSCLI"KEY"+STR$(G%):NEXT
1630 G%=0
1640 CLS:REPEAT
1650 PRINT"Phrase "STR$(G%+1);
1660 INPUT phr$
1670 IF LENphr$>0 OSCLI"KEY"+STR$(G%)+"+phr$:G%=G%+1
1680 UNTIL G%=9 OR LENphr$=0
1690 ENDPROC
1700 :
1710 DEFPROCstate(name$)
1720 X%=&70:Y%<0
1730 ?&70=0:??&71=&60
1740 $&60=name$
1750 A%=FNisfile
1760 ENDPROC
1770 :
1780 DEFFNisfile A%=5:=(USR(osfile) AND &F)
1790 :
1800 DEFPROCmsg(mess$)
1810 CLS
1820 PRINTTAB(20)mess$
1830 delay=INKEY(400)

```

```

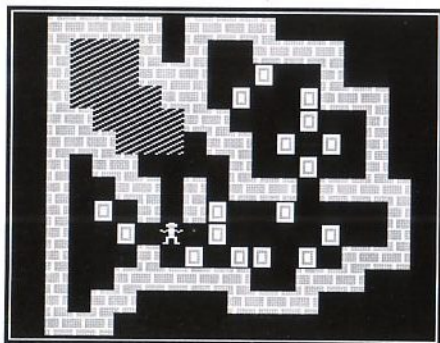
1840 ENDPROC
1850 :
1860 DEFPROCmore
1870 CLS:PRINTTAB(10)"(A)ll done or (M)ore?"
1880 REPEAT:shape$=GET$:UNTIL INSTR("AaMm",shape$)
1890 ENDPROC
1900 :
1910 DEFPROCgettitle
1920 PRINTTAB(15,11)"Title"
1930 VDU28,21,16,40,14,17,0,17,135
1940 CLS
1950 INPUT" title$
1960 IF LENTitle$>40 title$=LEFT$(title$,40) ELSE IF LENTitle$<40 title$=title$+STRING$(40-LENTitle$," ")
1970 VDU20,28,0,24,79,3
1980 IF N% <10 N$=" "+STR$(N%)ELSE N$=STR$(N%)
1990 IF M% <10 M$=" "+STR$(M%)ELSE M$=STR$(M%)
2000 YR$=RIGHT$(dir$,2):pl$=LEFT$(file$,1)
2010 line$=N$+ "/" +M$+pl$+YR$+ "/" +title$+shape$
2020 ENDPROC
2030 :
2040 DEFPROCconv
2050 offset$=RIGHT$("00"+STR$(M%),2)
2060 ENDPROC
2070 :
2080 DEFPROCadd:REPEAT
2090 N%=N%+1
2100 CLS:PRINTTAB(5,0)"Year - ";dir$;TAB(40)"Tray No. - ";M%;TAB(60);"Slide No. - ";N%
2110 PRINTTAB(15,2)"(P)ortrait or (L)andscape shape or (A)ll done?";
2120 REPEAT:shape$=GET$:UNTIL INSTR("AaPpLl",shape$)
2130 PROCsetshape
2140 IF shape$<>"A" PRINT" - ";shape$:PROCgettitle:PRINT#ch%,line$
2150 UNTIL N%=50 OR shape$="A"
2160 ENDPROC

```

Robol - The Screen Designer

Miroslaw Bobrowski gives you power over the mighty Robol.

So, how did you get on last month? Some of those screens are *hard*, aren't they! This month we present the screen designer so that you can create some really difficult ones for kith and kin. First type in and save the listing below, then run it.



A complete level

There are three options when it comes to choosing screens. You can design one from scratch, edit one that already exists in a separate file, or edit one that already exists in the *Robol3* file. A new screen has to be given the number after the last screen in the *Robol3* file and an original screen must already exist. There is room for 64 screens in one *Robol3* file.

Lines 1040 and 1050 could cause you problems depending on how many screens you have in your original *Robol3* file. These lines include a system of error traps, some of which might actually get in the way of accessing certain screens. While testing at BEEBUG we took out the error checks in line 1050 to give us access to all screens. This is done by

changing the UNTIL command to UNTIL (N%>0 AND N%<65). This will still make sure you are within the total range of screens. If you do this you must be sure about the screen number you are using, though nothing too drastic will happen if you get it wrong.

The instructions for the actual designer are shown below.

```

                                ROBOL DESIGNER
First choose the number of screen to
be designed.
Use arrow keys to move cursor box and
numeric keys to put character on the
screen, in particular:-
1 for a piece of wall
2 for a target (hatched) box
3 for a pack
4 for the Robol
5 for a pack at the target area.
To wipe a character use DELETE or 'o'
key.
The number of packs must be equal to
the number of hatched areas, and there
should be only one Robol character on
the screen.
When the screen is ready press COPY to
save screen file.

Press SPACE to continue
```

Main menu

When you have finished your design press Copy for the save menu. Here you can save the screen as a separate file, very useful if it isn't finished, or put it straight into the *Robol3* file. Enjoy!

```

10 REM Program RobDes
20 REM Version B 1.2
30 REM Author M.Bobrowski
40 REM BEEBUG May 1993
50 REM Program subject to copyright
60 :
100 ON ERROR CLOSE#0:OSCLI"FX4,0":REPO
RT:PRINT" at line ";ERL:END
110 MODE 7:HIMEM=&5800
120 PROCspritedata:PROCassemble
130 PROCinstr:PROCOptions
140 MODE 5:VDU 23;8202;0;0;0;:*FX4,1
```



```

150 DIM M% 320,B% 160:PROCclear:IF O%=
2 OSLI"LOAD "+R$+" "+STR$-B%
160 IF O%=3 PROCget_data
170 PROCgrid:IF O%>1 PROCexpand
180 ready%=FALSE:X%=10:Y%=10:PROCcurso
r(X%,Y%)
190 REPEAT
200 PROCmove_cursor
210 PROCset_sprite
220 UNTIL ready%:*FX4,0
230 PROCstore
240 END
250 :
1000 DEF PROCOptions
1010 REPEAT:ok=TRUE
1020 CLS:REPEAT:PRINT" Do you want
"" 1. Design a new screen"" 2. Edit
a screen designed by you and"SPC5;"stor
ed as a separate file"" 3. Edit an ori
ginal screen"CHR$130"Choice : ";:*FX15,
1
1030 O%=GET-48:UNTIL O%>0 AND O%<4:PRIN
T;O%
1040 PRINT"CHR$131"Now enter the number
of the screen"CHR$131"to be edited ";:
IF O%=3 PRINT"(1-50)"; ELSE PRINT"(51-64
)";
1050 x%=POS:y%=VPOS:REPEAT:PRINTTAB(x%,
y%)SPC(39-x%)CHR$7:INPUTTAB(x%,y%) "N%:
UNTIL (O%<3 AND N%>50 AND N%<65) OR (O%=
3 AND N%>0 AND N%<51)
1060 IF O%=2 ok=FNis(N%)
1070 UNTIL ok=TRUE:ENDPROC
1080 :
1090 DEF PROCclear
1100 FOR I%=0 TO 320:M%?I%=0:NEXT
1110 ENDPROC
1120 :
1130 DEF PROCget_data
1140 F%=OPENUP("Robol3"):PTR#F%=(N%-1)*
160
1150 FOR I%=0 TO 160:B%?I%=BGET#F%:NEXT
:CLOSE#F%:ENDPROC
1160 :
1170 DEF PROCgrid
1180 GCOL 0,1:FOR I%=63 TO 959 STEP64:M
OVE 0,I%:DRAW 1279,I%:NEXT
1190 FORI%=64 TO 1279 STEP64:MOVE I%,0:
DRAW I%,1023:NEXT
1200 ENDPROC

```

```

1210 :
1220 DEF PROCexpand
1230 V%=0:FOR I%=0 TO 159:H%=B%?I%:J%=H
% DIV 15:K%=H% AND 15
1240 M%?V%=J%:M%?(V%+1)=K%:V%=V%+2:NEXT
1250 V%=0:FOR I%=0 TO 30 STEP2:FOR J%=0
TO 19
1260 H%=M%?V%:PROCdisplay(J%,I%,H%):V%=
V%+1
1270 NEXT:NEXT:ENDPROC
1280 :
1290 DEF PROCcursor(x%,y%)
1300 GCOL 4,0:MOVE 64*x%,1023-y%*32:PLO
T 33,64,0:PLOT 33,0,-64:PLOT 33,-64,0:PL
OT 33,0,64:ENDPROC
1310 :
1320 DEF PROCmove_cursor:*FX15,1
1330 REPEAT:C%=X%:D%=Y%:sprite%=FALSE:K
%=-GET
1340 IF K%=136 AND X%>0 X%=X%-1
1350 IF K%=137 AND X%<19 X%=X%+1
1360 IF K%=138 AND Y%<30 Y%=Y%+2
1370 IF K%=139 AND Y%>0 Y%=Y%-2
1380 IF K%>135 AND K%<140 PROCcursor(C%
,D%):PROCcursor(X%,Y%)
1390 IF K%>47 AND K%<54 S%=K%-48:sprite
%=TRUE
1400 IF K%=127 S%=0:sprite%=TRUE
1410 IF K%=135 ready%=TRUE:IF NOT FNche
ck ready%=FALSE
1420 UNTIL sprite% OR ready%:PROCcursor
(X%,Y%):ENDPROC
1430 :
1440 DEF PROCset_sprite
1450 GCOL0,0:PROCdisplay(X%,Y%,S%)
1460 M%?(X%+10*Y%)=S%:PROCcursor(X%,Y%)
1470 ENDPROC
1480 :
1490 DEF PROCdisplay(X%,Y%,A%)
1500 CALL &B00
1510 ENDPROC
1520 :
1530 DEF FNcheck
1540 C%=0:D%=0:E%=0:FOR I%=0 TO 320:IF
M%?I%=2 C%=C%+1
1550 IF M%?I%=3 D%=D%+1
1560 IF M%?I%=4 E%=E%+1
1570 NEXT:IF E%=1 AND C%=D%=TRUE
1580 SOUND 1,-10,10,20:=FALSE
1590 :

```

Robol - The Screen Designer

```
1600 DEF PROCstore
1610 VDU22,7:PRINT""Do you want to""
1. Save screen data as a separate file"
" 2. Incorporate screen data into the'S
PC4;"robol3' file"CHR$130"Choice : "CHR
$129;
1620 REPEAT:O%=GET-48:UNTIL O%=1 OR O%=
2:PRINT;O%
1630 IF O%=1 PROCsave ELSE PROCput_data
1640 ENDPROC
1650 :
1660 DEF PROCsave
1670 FOR I%=0 TO 160:Q%=M%?(2*I%):R%=M%
?(2*I%+1):B%?I%=Q%*16+R%:NEXT
1680 OSCLI"SAVE rob"+RIGHT$( "00"+STR$N%
,2)+" "+STR$~B%+" +A0 "+STR$~(&3000+(N%-
1)*160)+" "+STR$~(&3000+(N%-1)*160)
1690 ENDPROC
1700 :
1710 DEF PROCput_data
1720 F%=OPENUP("Robol3"):PTR#F%=(N%-1)*
160
1730 FOR I%=0 TO 160:Q%=M%?(2*I%):R%=M%
?(2*I%+1):B%?I%=Q%*16+R%:BPUT#F%,B%?I%:N
EXT:CLOSE#F%:ENDPROC
1740 :
1750 DEF PROCinstr
1760 CLS:PRINTTAB(9,1)CHR$131CHR$157CHR
$129"ROBOL DESIGNER "CHR$156
1770 PRINT"First choose the number of
screen to""be designed."
1780 PRINT"Use arrow keys to move curso
r box and""numeric keys to put characte
r on the""screen, in particular:-"CHR$
131" 1"CHR$134"for a piece of wall"CHR
$131" 2"CHR$134"for a target (hatched)
box"CHR$131" 3"CHR$134"for a pack"
1790 PRINTCHR$131" 4"CHR$134"for the R
obol"CHR$131" 5"CHR$134"for a pack at
the target area""To wipe a character us
e DELETE or '0'""key."
1800 PRINT"The number of packs"CHR$129"
must be equal"CHR$135"to""the number of
hatched areas, and there""should be"CH
R$130"only one"CHR$135"Robol character o
n""the screen.""When the screen is rea
dy press COPY to""save screen file."
1810 PRINT'TAB(5)"Press SPACE to contin
ue ";:REPEATUNTILGET=32:ENDPROC
1820 :
```

```
1830 DEF PROCspritedata
1840 P%=&C00
1850 [OPT 0
1860 EQUUS STRING$(32,CHR$0)
1870 ]
1880 FOR I%=1 TO 192 STEP 4
1890 READ a$
1900 [OPT 0
1910 EQUO EVAL("&"+a$)
1920 ]
1930 NEXT
1940 ENDPROC
1950 :
1960 DATA FF2F2F2F,F0F0F0F,FF0F0F0F,2F2
F2F2F,FF0F0F0F,4F4F4F4F,FF4F4F4F,F0F0F0F
1970 DATA 22118844,22118844,22118844,22
118844,22118844,22118844,22118844,221188
44
1980 DATA 43707000,53535343,43535353,70
7043,2CE0E000,ACACAC2C,2CACACAC,E0E02C
1990 DATA 1212110,1155BB11,20301001,602
020,8484880,88AADD88,40C08008,604040
2000 DATA BC8F8FFF,ACACACBC,BCACACAC,FF
8F8FBC,D31F1FFF,535353D3,D3535353,FF1F1F
D3
2010 DATA 1212110,1155BB11,20301001,602
020,8484880,88AADD88,40C08008,604040
2020 :
2030 DEF PROCassemble
2040 scr_addr=&70:temp=&72:row=&74
2050 temprow=&75:column=&76:index=&77
2060 x=&78:y=&79:addr=&7A
2070 FOR pass=0 TO 2 STEP 2
2080 P%=&B00
2090 [OPT pass
2100 STX x:STY y
2110 ASL A:ASL A:ASL A:ASL A:ASL A:STA
index
2120 LDA #&58:STA addr+1:LDA #0:STA add
r
2130 LDY y:BEQ zero
2140 .loop1
2150 CLC:LDA addr:ADC #&40:STA addr
2160 LDA addr+1:ADC #1:STA addr+1
2170 DEY:BNE loop1
2180 .zero
2190 LDX x:BEQ store_addr
2200 .loop2
2210 CLC:LDA addr:ADC #16:STA addr
```

Continued on page 30

Decision Trees

It's Make Your Mind Up Time with your very own G. T. Swain.

These days almost everybody is an expert. There seem to be no limits to available specialities and sometimes the sheer diversity of experts leaves me breathless. All these experts seem to have one thing in common. They are always eager to proffer opinions on every subject under the sun, but when it comes to making decisions it is always left to somebody else. The one speciality lacking is that of an expert decision maker.

I said most people claim to be experts, and I am as guilty as the next person when it comes to making decisions. There is however a method for arriving at 'inevitable' decisions, when the alternative is waiting an eternity for somebody else to get there first.

The tool I use for the job is called a *Decision Tree*. Anybody can apply the same methods, and they can be easily adapted for the computer. They do not require any specific piece of software, the method can be adapted for use in any word processor or text editor, or a spreadsheet. It is also possible to write simple programs in almost any of the popular computer languages.

A decision tree in its simplest incarnation is a list of simple questions with either/or answers; binary choices. The way it works is to apply values to each of the options. By totalling or *compounding* these values it is possible to determine the relative suitabilities of different available courses of action. The trick is to ask and answer the right questions. To determine that one or other alternative is most suitable it is

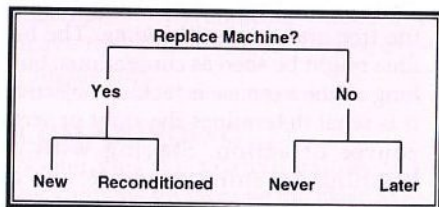
essential to limit the scope of the answers to the choices on offer.

Let's try an example using an ordinary word processor (I am using View). We will try a situation that most people can readily understand. The question is whether or not to replace an elderly washing machine. It is still serviceable, but the service engineer has been called in several times and it is becoming increasingly costly. There are good reasons for both replacing and not replacing the machine.

The first entry on our decision tree is the question of whether or not to buy a replacement machine. There are only two possible answers. If we decide to buy, then our next dilemma might be whether to buy a new machine or a reconditioned model. If we decide not to buy, we might like to reconsider at a later date or to give up on the idea altogether.

Thus far the choices have been fairly simple, but if we are being methodical, then there will be several subsequent questions, so we need to have a recognisable structure.

Our decision tree will start off looking like this:



Decision Trees

This brings into play two essential concepts about decision trees. These are *Terminators* and *Loops*. These are both important elements within the tree structure. The purpose for which the decision tree is being constructed is as an aid to making some kind of decision. A terminator is a decision; in this case the word 'never' is applied as a terminator. The loop is a combination of a delay factor and a means of re-direction. The word 'later' in the context of our example gives us the opportunity of reconsidering the options at a later stage. There should be at least one terminator and one loop at each level of the decision tree.

Although I opted to use a word processor I could easily have constructed a program along similar lines. Using pseudocode, rather than any specific computer language, our program might be constructed as follows:

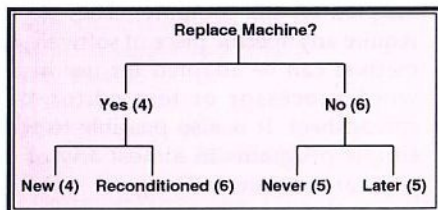
```
10 REM Decision Tree
20 INPUT "First Question (Yes/No)"
30 IF yes INPUT "Next Question
(Yes/No)"
40 ELSE INPUT "(Terminate/Loop)"
50 IF terminate STOP
60 IF loop GOTO first question
70 IF option1 THEN .....
80 IF option2 THEN .....
```

I have used the taboo word 'GOTO', but this is not about good or bad programming practices, more about the decision process itself.

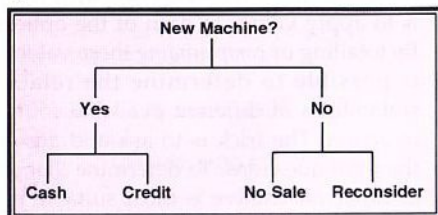
There is another essential ingredient to the tree and this is *Weighting*. The basic idea might be seen as contentious, but as long as the exercise is tackled objectively it is what determines the right or wrong course of action. Staying with our Washing Machine example: we can assign values according to known

factors. For instance if the cost of repairing the machine is becoming increasingly expensive, the answer 'Yes' to the first question makes more sense than answering 'No'. We can realistically assign a higher value to 'Yes' than 'No'. Just how much value to apply is subject to personal interpretation. The actual values or range of values does not matter that much, it is more important to be consistent throughout. If you start with a range of values, say 1 to 10, then the same range of numbers must be employed on all lower levels.

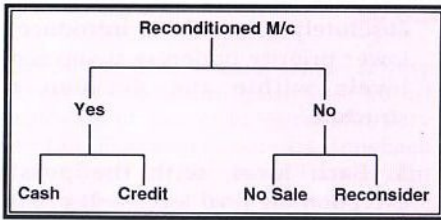
In this example we decide to use a range of values between 1 and 10 where 1 is least favourable and 10 most desirable. We will apply 7 to 'Yes' and 3 to 'No'. If we were looking at an empty bank account then our criteria might be 1 for 'Yes' and 9 for 'No'. Then again the need to replace the machine might outweigh the state of the bank account so the values used will fall somewhere between the two sets of values above, eg. 4 and 6.



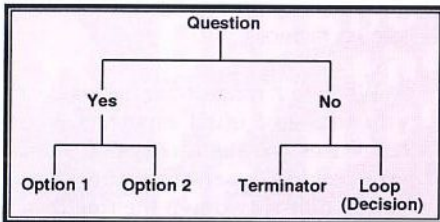
This takes us on to the next set of questions:



and:

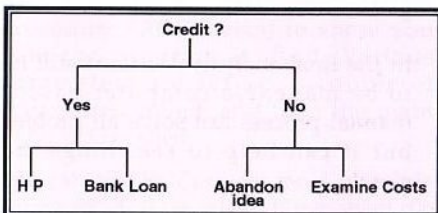


Each question and its immediate responses can be described as a level, and the levels usually conform to a regular pattern:



The options from this layer will become the questions for the next level.

It is the nature of Decision Tree structures that most readily lends itself to computerisation. Within a word processor the basic pattern can be copied or moved quite easily. New levels can be introduced at any point as and when new factors need to be considered. To illustrate the point, using the washing machine example and the options of cash or credit, we might begin to establish a good case for credit. It could occur that a bank loan decision might be introduced into the decision tree structure as follows:



The same principle of two options, a terminator and a loop back to earlier level(s) is maintained.

The principle of looping needs further explanation at this point. As more levels are created it becomes easier to loop back not just one level, but any number of levels including back to the top of the tree. The principle of looping also illustrates a variation on the decision tree principle, as you approach each higher level you are asking the question, "This level or its parent?"

WEIGHTING

Although the values to be applied are a matter of personal choice the way in which they affect the final set of choices is critical. Obviously the washing machine example is relatively trivial, other decisions may depend on considerably more options. In real life the values applied to different options will need to be regularly reviewed. For example, if making investments (buying shares) the acquisition price will vary and the decision to buy will seem more or less favourable according to the reasons for purchase and the prevailing price. This is where the computerised Decision Tree comes into its own.

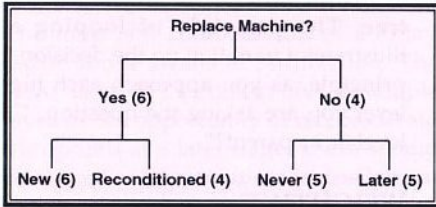
In its simplest form the word processed tree can be edited. If you devised some kind of computer program or perhaps used a spreadsheet then it might be possible to automate the input and the calculation of Weighting values. I will leave that possibility to a more expert programmer.

COMPOUNDING

Sometimes the relative values of any two options prove to be too close to each

Decision Trees

other to make either choice more attractive. When this happens you need to look at the weighting factors of at least one succeeding level. In our example the Yes/No options are rated as 6 and 4 respectively. The offspring options also produce close values as follows:



Pursuing the 'Yes' option gives two possible compounded values of 12 and 10. The 'no' option also gives two compounded values of 9 and 9. In this way one of the four possible choices begins to appear more attractive.

There is one disadvantage to the process of compounding in that the more levels that are added together the greater the range of options that will exist. If taken too far, the final compounded values will tend to become unreliable.

Before leaving the reader to experiment it is worth re-stating the basic principles.

1. To obtain a sensible solution careful thought must be given to both the questions asked and the answers obtained.
2. Break down the questions into straight 'Either/Or' format. If the original problem offers more than two options then prioritise the options and

make your starter question a direct choice between the first two options. If absolutely unavoidable introduce the lower priority option(s) at succeeding levels within the decision tree structure.

3. Each level, with the possible exception the final level, will comprise two straight options, a terminator and a loop.

4. Each option will be assigned a value or percentage depending on one or more known factors.

5. Whatever range of weighting values you use you must ensure that each level uses the same range of values. If you decide to change the range of values half-way down the tree then it is essential to alter the preceding levels using the new range of values.

6. If it is found that the values applied do not indicate either most or least favourable options, then it is obvious that more or different information is required about the choices available. Possibly you have been answering the wrong questions.

7. Although the use of a computer makes frequent changes possible, the problem may be complex enough to justify the production of hard copy, either as a visual aid or as the basis of a questionnaire which can be used to gather more data.

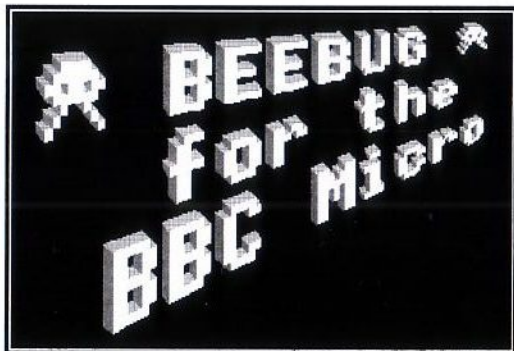
In the final analysis decisions still have to be made. No computer-based or manual process can solve all problems, but it can help to see things more clearly.

B

3D Text

by O.R. Thomas

Text on the BBC micro's screen is very versatile - you can print just about anything, in lots of colours, and very fast too. However, a serious drawback of the Beeb for the more flamboyant amongst us is the decidedly two-dimensional nature of the screen display. This program will give you the next best thing to a holographic TV.



Sample 3D text display

Any text or user-defined characters can be displayed mode 1, in a 3-D style, and with different colours used for the different faces of the letters. The examples shown here will give you the general idea. You can use the 3-D text for title screens, posters, or anything requiring an extra dimension.

The program consists mainly of the procedures that actually plot the 3-D characters (lines 1000 to 1570). All these must be included in your own program. Also listed here is a short demonstration procedure (*PROCdemo*) to show you how the program is used. Various parameters are set up, a couple of procedures called, and then the main *PROC3D* is called.

The call to *PROCprepare* must be made before *PROC3D* is called. If you want the

text to appear centred on the screen then *PROCcentre* should also be called, as in the demonstration procedure. If the text is to be placed in any other position across the screen then spaces should be added to the END of the text string as it is this end that is normally positioned hard against the right hand side of the screen. Indeed, it is just this action in that *PROCcentre* performs automatically when it centres each line of your text.

There are several styles of text that the program can produce. These differ in the relative proportions of the letters. Some are double height, some double width and some both. Naturally, the larger the letters, the less will fit onto the screen. You should choose the style suitable for your text and assign the style number to the variable *style%*. These are the styles available.

Style	No. of lines on screen	No. of letters per line
1	3	13
2	3	6
3	2	13
4	2	6
5	1	13
6	1	6

The number of possible lines of text actually used to display words should be assigned to the variable *lines%* and the text, or characters, required in these lines assigned to elements of the string array *line\$()*, up to the value of *line%*. The demonstration procedure shows just how all this is done. At this point *PROC3D* can be called to produce the display.

When using the 3-D procedures in your own program you could also make a

3D Text

call to a screen dump routine (such as those in BEEBUGSOFT's Dumpmaster or Computer Concept's Printmaster) after the PROC3D call, if a hard copy of the 3-D text is needed.

As well as the six letter styles provided you can also design ones of your own. The main characteristics of the styles are contained in the data statements in lines 1450 to 1500. Each data statement corresponds to one style and each style requires seven data items. By altering one or more of these parameters you can create a 'new' style.

The seven data items are read into the seven variables *height*, *mletters%*, *mlines%*, *add*, *change%*, *alt%* and *mult%*. These control, respectively, the vertical distance between elements of the characters, the maximum number of letters in a line, the maximum number of lines on the screen, the width of the characters, the height of the elements, the vertical position of the right hand end of the lines, and the distance between lines.

Not all values for all the variables will give exactly what you might expect. For example substituting a value of 2 for *height%* in the data for style 1 will double the vertical distance between elements, leaving a gap between each. However, using a value of 3 will not produce a smaller gap but the same sized one between elements 3 and 4, and 6 and 7 only!

To start you off, try any of the styles with this data:

2,7,2,0.5,1,0,20

A bit of experimenting can not only keep you amused for hours but produce some interesting, and even useful results.

PROGRAM NOTE

The program determines the shape of the character to plot it onto the screen, not by way of the OSWORD call with the accumulator set equal to 10 as you might expect, but by writing the characters in the normal way at the bottom of the screen and using POINT to see which pixels are lit.

This program was first published in BEEBUG Vol.4 No.3

```
10 REM Program 3D TEXT
20 REM Version B0.2
30 REM Author O.R.Thomas
40 REM BEEBUG May 1993
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO 1710
110 :
120 DIMline$(3)
130 MODE 1
140 VDU19,3,2;0;19,2,7;0;
150 PROCdemo
160 END
170 :
1000 DEF PROC3D
1010 FOR L%=lines%-1 TO 0 STEP -1
1020 ang%=alt%-(L%*mult%)
1030 PROCword(line$(L%+1),1008-(32*L%))
1040 NEXT L%
1050 ENDPROC
1060 :
1070 DEF PROCword(line$(L%),cy%)
1080 COLOUR2
1090 PRINTTAB(20,31)line$(L%);
1100 FOR py%=1 TO 33 STEP4
1110 angle%=(py%/height)+ang%
1120 a1=TAN(RAD(angle%)):a2=TAN(RAD(ang
le%-change%))
1130 sub=4:lp=1309:rp=1359
1140 FOR px%=(LEN(line$(L%))+1)*32 TO 0
STEP-4
1150 IF POINT(px%+640,py%)=2 PROCbox
1160 rp=lp:sub=sub+add:lp=lp-sub
1170 NEXT px%
1180 NEXT py%
1190 PRINTTAB(20,31) SPC(18);
```



```

1200 ENDPROC
1210 :
1220 DEF PROCbox
1230 PROCface(2,lp,rp,lp,rp,((1579-lp)*
a2)+oy%,((1579-rp)*a2)+oy%,((1579-lp)*a1
)+oy%,((1579-rp)*a1)+oy%)
1240 PROCface(1,lp-sub,lp,lp-sub,lp,((1
579-lp)*a2)+oy%+sub,((1579-lp)*a2)+oy%,(
(1579-lp)*a1)+oy%+sub,((1579-lp)*a1)+oy%
)
1250 PROCface(3,lp-sub,rp-sub,lp,rp,((1
579-lp)*a1)+oy%+sub,((1579-rp)*a1)+oy%+s
ub,((1579-lp)*a1)+oy%,((1579-rp)*a1)+oy%
)
1260 ENDPROC
1270 :
1280 DEF PROCface(col%,x1,x2,x3,x4,y1,y
2,y3,y4)
1290 GCOL0,col%
1300 PLOT4,x1,y1
1310 PLOT4,x2,y2
1320 PLOT85,x3,y3
1330 PLOT4,x2,y2
1340 PLOT4,x4,y4
1350 PLOT85,x3,y3
1360 ENDPROC
1370 :
1380 DEF PROCprepare
1390 RESTORE1450
1400 FOR L%=1 TO style%
1410 READ height,mletters%,mlines%,add,
change%,alt%,mult%
1420 NEXT
1430 IF lines%>mlines% THEN CLS:PRINT "
Too many lines for style ";style%:END
1440 ENDPROC

```

```

1450 DATA4,13,3,.13,1,-13,8
1460 DATA4,6,3,.65,1,-13,8
1470 DATA1.95,13,2,.13,2,-16,15
1480 DATA1.95,6,2,.65,2,-16,15
1490 DATA1.33,13,1,.13,3,-29,8
1500 DATA1.33,6,1,.65,3,-29,8
1510 :
1520 DEF PROCcentre
1530 FOR L%=1 TO lines%
1540 IF LEN(line$(L%))=mletters% GOT015
60
1550 line$(L%)=line$(L%)+STRING$(mlett
ers%-1-LEN(line$(L%))*.5," ")
1560 NEXT L%
1570 ENDPROC
1580 :
1590 DEF PROCdemo
1600 VDU23,230,60,126,219,126,36,66,129
,0
1610 style%=1
1620 lines%=3
1630 line$(1)=CHR$230+" BEEBUG "+CHR$23
0
1640 line$(2)="for the"
1650 line$(3)="BBC Micro"
1660 PROCprepare
1670 PROCcentre
1680 PROC3D
1690 ENDPROC
1700 :
1710 ON ERROR OFF
1720 MODE 7
1730 IF ERR<>17 THEN REPORT:PRINT " at
line ";ERL
1740 END

```

B

Renewal Rates for BEEBUG magazine and magazine disc subscriptions

See this month's editorial for further explanation.

The table below shows the renewal rate applying after the May issue 1993 according to the first issue of the renewal period. For joint BEEBUG/RISC User subscriptions add half the appropriate BEEBUG renewal rate to the full RISC User renewal rate (£18.40).

Renewal Issue	Issues to go	Mag UK	Mag Europe	Mag Mid-E	Mag Am+Af	Mag Else	Disc UK	Disc O'Seas
Jun	9	16.56	24.75	30.15	32.85	35.55	45.00	50.40
Jul	8	14.72	22.00	26.80	29.20	31.60	40.00	44.80
A/S	7	12.88	19.25	23.45	25.55	27.65	35.00	39.20
Oct	6	11.04	16.50	20.10	21.90	23.70	30.00	33.60
Nov	5	9.20	13.75	16.75	18.25	19.75	25.00	28.00

For example, someone who applies for a new joint subscription starting in August and who lives in the Middle East would be charged: £35.00 + £12.25 = £47.25

New RU + Half BB

All renewals should be sent to BEEBUG Subscriptions Dept., 117 Hatfield Road, St Albans, Herts AL1 4JS, tel. 0727 840303.

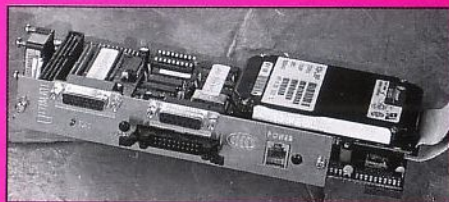
New Generation!

RISC

user

RISC User, the highly popular magazine for Archimedes users, is bigger and better. The new RISC User is now B5 size which offers a sophisticated design, bigger colour illustrations and bigger pages with more information. Altogether better value and no increase in price.

RISC User is still a convenient size to assemble into an easy-to-use reference library, containing all the information you need as an Archimedes user. Every issue of RISC User offers a wealth of articles and programs with professionally written reviews, lively news, help and advice for beginners and experienced users, and items of home entertainment. Altogether RISC User has established a reputation for accurate, objective and informed articles of real practical use to all users of Acorn's range of RISC computers.



SUBSCRIPTION DETAILS

As a member of BEEBUG you may extend your subscription to include RISC User for only £10.50 (overseas see table).

Destination	Additional Cost
UK, BFPO & Ch Is	£ 10.50
Rest of Europe and Eire	£ 15.40
Middle East	£ 19.60
Americas and Africa	£ 21.90
Elsewhere	£ 33.00

WORKING WITH IMAGES

A new series on the use of images on the Archimedes, which looks this month at two image translation and processing packages.

KALEIDOSPRITE

A program offering a fascinating never-ending sequence of kaleidoscope patterns.

S-BASE: A PROGRAMMABLE DATABASE

A review of this advanced programmable database from Longman Logotron.

THE ULTIMATE EXPANSION

A new concept in hardware expansion.

PROFESSIONAL SCOREWRITING

A look at PMS, a professional quality music typesetting package.

WORDZ OF WIZDOM

A new easy-to-use wordprocessing package from Colton, aimed at the general market.

THE NEW INTEGRIX COLOURJET PRINTER

The Colourjet Series 2 printer is out and is very cheap.

WRITE-BACK

The readers' section of RISC User for comment, help, information - a magazine version of a bulletin board.

INTO THE ARC

A regular series for beginners.

TECHNICAL QUERIES

A column which answers your technical queries.

THE DOS SURVIVAL GUIDE

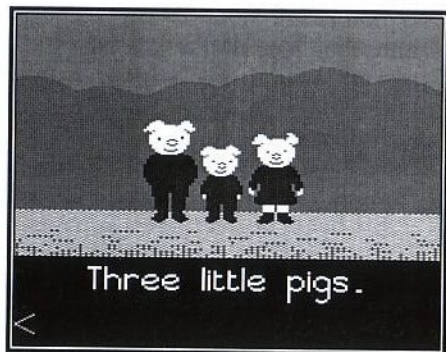
A series of articles on how to use the PC Emulator.

BEEBUG Education

Mark Sealey assesses an excellent new software suite for infants.

It is not by chance that BEEBUG Education turns this month to the second offering from Sherston Software in less than a year. In Volume 11 Issue 5 two suites for infants in the *Peek-a-boo* series were looked on very favourably.

Now more material for children at Key Stages 1 and 2 comes under the spotlight: *Three Little Pigs at Home*. This is a series of computer-based activities for children of all abilities designed to address most areas of the curriculum. It is particularly strong in mathematics, including the often neglected yet vital area of estimation.



THE PACK

Three Little Pigs is available in 5 $\frac{1}{4}$ " DFS and 3 $\frac{1}{2}$ " ADFS formats for the entire 8 bit range. With these come a teacher's booklet and documentation, a children's booklet, 5 work cards, 2 worksheets and particularly colourful A3 and A4 overlay sheets for Concept Keyboard. Given its price, this would in itself represent good value for money so late in the day for 8-bit software (this is likely to be the last

Sherston 8-bit release) even if the quality of the product were nothing special. But it is.

THE STORIES

These are in two parts. The users have to help the pigs construct a rather wobbly and insecure house - a beehive hut like those common in many parts of the world today; it is blown down by you-know-whom and a better one constructed of wood - only to meet the same fate. Counting activities are employed, tallying, estimation and number patterns. Finally the bricks make it but only after some logical thought, use of simple ratios (sand:cement) and intriguing pattern work.

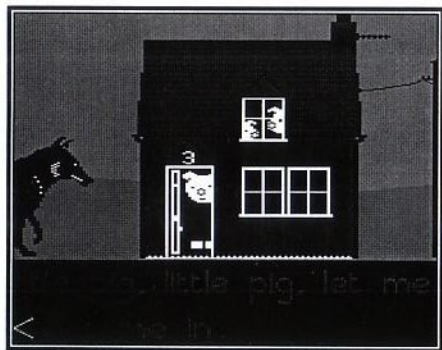
By now the children are well involved, having used additionally a further variety of number, spatial and problem-solving skills. The graphics, text and use of sound are just right, and neither interfere with the action nor detract from the learning exercise.

But what of the wolf? He's prowling around outside waiting to get his own back. But the door is locked, the fire lit and - vegetarians or lupophiles turn away - a pot is set to boil him in.

BOXES AND TILES

Having sorted the wolf out, the pigs can settle down in their new home. A delivery of boxes arrives and the little pigs need help to carry them in and unpack them. More of what we adults know as 2D and 3D geometry; to the pigs (sorry, the children working at the

computer) it is more juggling with broken plates, stacking boxes and telling colours apart, nesting the boxes, putting them in order and comparing the dimensions of various familiar household objects.



In fact, everything here can be directly related to the National Curriculum. Simple skills are exercised to some purpose in best Primary education fashion. The teacher's book suggests that 30 minutes would be needed for this two-player phase. In fact, this figure can vary according to the ability of the players, but not their patience: involvement is high. This timing also depends on how long the pupils spend discussing the activity on which they are engaged - a good thing that can promote greater understanding.

Tiles is the only game that is designed for one player only - with the others there is the option for one or two. Consequently, it takes a little less time. There is nothing new in the activity, indeed some of the earliest educational software ten years ago exploited the ease with which graphics can be manipulated in real time on screen, and combined to produce mathematically

pleasing effects (known as tessellations). Triangles, squares, hexagons and octagons are used, and the results can be printed out straight away on an Epson compatible printer

Here, rotational as well as reflective symmetry is dealt with, although it isn't introduced in the National Curriculum until beyond the level at which this software is aimed. That's no bad thing: some children find symmetry quite hard and this part of the package might just draw attention to early difficulties.

IN THE DARK

To bed at last, tired pigs, only, in best adventure game tradition, one of the piglets is missing! Since it's dark, a torch is pressed into service to go and look for him. This part of the suite bears some resemblance to Peek-a-boo in that it takes place in a familiar, domestic, environment.

Although the written form of language of the home is given great prominence as the pupils need to explore it, so too is extending their skills of spatial awareness. What is the geographical relationship of one room to another? This too can be extraordinarily hard for even quite old children (and for adults too - think of all those maze games that kept us up well beyond bedtime when we first got our computers). Opportunities to practice this are well thought out and levels of difficulty carefully graded (see later). Another nice touch is that the teacher can determine the location of the lost piglet if they wish - otherwise it is randomly placed each time the game is started.

USING TLP IN THE CLASSROOM OR AT HOME

There you have it, as they say. The pig is safe, the poor wolf casseroled, the house left tidy and the pigs asleep. Although the package is not a single adventure but split into 4 or 5 separate parts, the material is coherently presented both in terms of content and on-screen cues.

Its recommendation extends beyond schools: parents wanting a simple item of software suitable for their own children to use will be pleased with this compact and well-presented package. It is easy to use, very attractive visually and capable of being adapted for use by children of varying abilities and familiarity with the computer. But that is not all.

DOCUMENTATION AND SUPPORTING MATERIAL

Everything you as an adult are likely to want is contained in the well written and indexed 32 page A5 manual: from National Curriculum references and a topic web, through word lists and ideas of further use to booklists for children and teachers as well as places to visit that support the theme of houses and homes.

This is a popular topic for children entering school for the first time. Sherston's excellent pack could well be used at its centre... indeed, it is described and sold as a 'topic based package'. Like all the best resources to support topic work - and don't listen to John Patten - The Three Little Pigs at Home can be used selectively, is adaptable and, most important of all, works with the child and at their level.

One of the ways of achieving child-centredness is to be able to adapt your material to the experience, ability and even the mood or predisposition of the

child using it. This is eminently possible with Three Little Pigs thanks to the facility within the Teacher Control menu (and indeed during the game itself by 'secret' key combinations) not only to set the level of difficulty but the number of players (see above).

But this same menu also leads to something quite rare for software of this type - or indeed for any educational software. You can compile up to eight sequences of children's names to play any combination of the component parts of the package, specifying any of the levels of difficulty (1-3). When it is run, the names are displayed on screen and the children come to the computer to do their bit. The point of this is not so much to automate use of the computer as to have a set of appropriate activities ready for any occasion.

CONCLUSION

Three Little Pigs is a winner. The graphics are simple, the text easy to cope with and the options for use are flexible and carefully thought out. Although the medium of the pack is a game, it also teaches something. For example mortar is the correct term for what we use to join bricks together, not the word at first used by one of the children working on it for the purposes of this review: "paste" (!). Its pace and variety keep children on their toes. A familiar story is used to put them there and the whole thing works well. Even now, go out and buy it.

Product Supplier	Three Little Pigs at Home Sherston Software Swan Barton, Sherston, Malmesbury, Wiltshire SN16 0LH. Tel. 0666 840433 Fax 0666 840048
Price	£31.67 inc. VAT and p&p



Bladedancer

Marshal Anderson goes Orc bashing.

Product	Bladedancer
Supplier	Omicron Technologies P.O. Box 37, Daventry, Northants NN11 4UG
Price	£11.95 inc VAT

A new game for the BBC series of computers is rare enough and, no matter what the content, we all have to be pleased to see a new company taking the machines seriously - Omicron Technologies are taking their work very seriously.

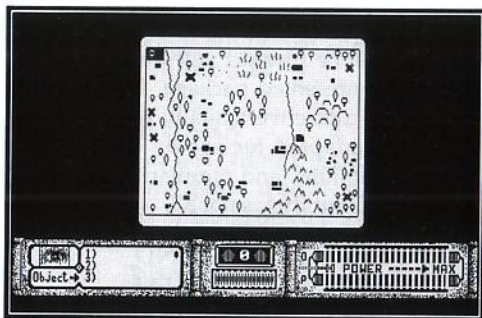
Let's look at what we're dealing with here. This isn't another shoot 'em up variation, it's not Pacman or platforms, this is a full blown role playing adventure. Not only that, we're talking graphics digitised from video and sampled sound, though you do need sideways RAM to get all the goodies.

WHAT'S GOING ON?

The basic story has a Dungeons and Dragons feel, as you would expect, and I won't go into it here. Simply, you have to collect nine pieces of a pentagram to prevent the end of life as we know it. Let's get going.

The game arrives on five discs. This is not as irritating as you might imagine and a couple of swaps will get you going. We start with the Home Town disc and our first task is to decide who we are. You can choose from sorcerer, archer and warrior - each comes in both genders - and that decides the graphic that represents you in the game. On selection you get a breakdown of the character and a picture. I must admit,

technically clever though the digitised pictures are, I should have preferred drawings - the sorcerer looks like a rather surprised Tom Hanks. Here, you also get the 'Vegetarian Option'; I'll leave the vegetarians to try that one out. Finally you get to select sideways RAM and off you go to the first map.



Here be dragons (probably)

There are four maps or counties, each on a different disc and, as each one contains more than 150 locations, it will be a while before you need to swap. You move around the map using the cursor keys but Omicron have not made life easy by letting your cursor do the walking - you can only move to adjacent locations. Pressing Return puts you into a location which may contain a puzzle, a monster or nothing at all. Each location is a 'widescreen' scrolling area.

OKAY, MONSTER, MAKE MY DAY

If you find yourself in a combat zone, and you'll soon know, you get to fight the nasty monster. There are seventeen varieties to choose from and you use a combination of hacking, shooting, parrying and throwing to beat the beast - negotiation is not an option. As you

fight, you get to see your's, and the respective monster's, power drain away; when all the power is gone it's 'Game Over' for the loser. At the end of the combat zones you reach the shop where you can use some of the 200 gold pieces you started off with to replenish your power.

Each character type has things called career points. What this indicates is the state of your particular ammunition. You start with ten points and each time you cast a spell or loose an arrow you use one point. So, you can't go out, Arnie style, with your twenty-rounds-a-second repeating crossbow; you need to take care. Each time you kill a monster you gain a career point - you can buy more but they're very expensive.



Don't ask for credit!

PUZZLED? YOU WILL BE

The combat, however, is not what it's all about. Strewn about the place are various objects that form long and sometimes complex linking puzzles. You know the sort of thing; you must get past the bull but you need a rope to tie it down, a trader has a rope but wants a gem for it and so on. Sometimes you not only need objects but there will be something to fight as well. There are also word puzzles, some of which are dead straight forward while others are cryptic.



The rewards of helping others

At the end of these strings of puzzles you will find bits of the pentagram, once you have found all the bits on a map you will be given the password to take you to the next. This continues until you have completed them all at which point you will be treated to something rather special.

SO, IS IT ANY GOOD?

Of its type this package is at least very good. Of its type on the BBC nothing I have seen comes close and it's very sensibly priced. You certainly need to take the game quite seriously; the puzzles are long and involved, you'll need to take notes. You can only re-enter the game at the start of a given map, so you need to be able to complete each at a sitting. As moving around the map can take some time you need to think carefully about the order in which you do things. You can only carry up to three objects and, as there are 150 individual puzzles to solve don't expect to do it the first time. This approach is helped by the fact that puzzles are always in the same place and, as far as I could tell, so are the objects needed to solve them.

Technically it's a great achievement, although the sound doesn't contribute much and, if the art work is a bit dull, well, let's see what they come up with next. **B**

Troubleshooting Guide (Part 5)

Gareth Leyshon wraps up his series with a look at the ADFS and Boot Files.

In my final guide to solving problems, I'll look at the Advanced Disc Filing System (ADFS) supplied with the Master series and which may have been installed in some Beebs. I'll also look at how to write a !BOOT file.

To select the ADFS, simply type *ADFS; to get back to DFS, use *DISC. When ADFS is selected, the computer will immediately try to read data from the ADFS disc in the drive - if you haven't put one in you'll have a long wait staring at a computer that seems to be tied up. Pressing Break will not cure this as the computer also reads the ADFS disc on reset: Ctrl-D-Break will do the trick. To select ADFS 'passively', use *FADFS. You must then use *MOUNT to initialise a disc after inserting it. When changing discs, use *DISMOUNT before removing the one you have finished with, change discs and *MOUNT the next one. If using dual drives, use *MOUNT 0 or *MOUNT 1 as appropriate. The rear side of a 40 track disc cannot be used by a standard ADFS; a double-sided 80-track disc is treated as if the two surfaces are two parts of the same drive. Floppy drives 0 and 1 are known as drives 4 and 5 if you also have a *Winchester* (hard disc) system.

DISC FAULTS

On ADFS these are generally similar to those on DFS as discussed last time, although the error numbers could be different. Investigate the error as explained for DFS discs: consider the possibility of having the drive set to 40 rather than 80 tracks (or vice versa) or putting a DFS disc in by mistake. The test technique for ADFS is to *MOUNT it, and the rules given for DFS hold: an instant error message suggests something is seriously wrong with the disc, success in mounting a disc which

later gives trouble suggests a wrong track setting or that a particular file is corrupt, while a pause - possibly a very long pause - suggests that a DFS or other format disc has been put in by mistake.

If you get an error message saying 'Disc changed' or 'Broken Directory' after changing a disc, put in the disc you just took out, *MOUNT it, then immediately *DISMOUNT it, put in the disc you want to use and finally *MOUNT that. If you still get 'Broken Directory', try *DIR \$ on the old disc. If you have accidentally selected ADFS when you wanted DFS, hold down the Ctrl and D keys, and press Break. If these fail, try the following cures for each particular error message:

'Bad Sum' - press Break.

'Bad FS Map' - try *DIR \$, if that fails, Break.

'Broken Directory' - see above, or try *DIR \$, if that fails, Break.

'Can't delete CSD' - you can't erase the Currently Selected Directory, i.e. the directory you are in.

'Can't - File Open' - type *CLOSE.

'Data Lost' - press Break; an error has caused data to go astray.

'Disc Changed' - replace the old disc, *DISMOUNT it, then *MOUNT the new disc.

'Map Full' - compact the disc (see *COMPACT, below)

'No Directory' - you haven't *MOUNTed following *FADFS.

FORMATTING

To prepare a blank disc for use with the ADFS, it must be formatted. The ADFS formatter is on the Welcome disc supplied with the Master 128 (it is in ROM on the Master Compact): start it up using Shift-A-Break. ADFS utilities are the third option on the main menu; the program you want from the sub-menu is called AFORM. You will be asked for the 'size' of the disc. If your drive is 40 track, your size is S(mall). With 80 track: for single-sided use M(edium), and for double-sided use L(arge). The disc as well as the drive must be up to the required standard of tracks and sides, and both must be of 'double density' capacity.

A file name in ADFS may be up to ten letters long, compared with seven in DFS. The big difference - the one that earns the title 'advanced' - comes in the use of directories. Before use, a directory must be created: this is done using *CDIR <name>, e.g. *CDIR LETTERS. A directory name, like an ordinary filename, can be up to ten characters long.

DIRECTORIES

When you catalogue a disc, you get a list of the files in the top-level or *root* directory, named '\$', and any directories contained in it (directories normally have DLR after their name). To find out what a directory called <dir-name> contains, you can type *DIR <dir-name> to select it and then *CAT, or you can use *CAT <dir-name> to examine it without selecting it. Any such directory may in turn contain directories as well as files.

A given directory will hold a maximum of 47 references to other files or directories. It is easy to get lost in a maze of directories, but *DIR ^ will get you up one level from where you are, *DIR \$ gets you to the top level and *BACK takes you to the last directory you visited.

There is a program called CATALL on the Welcome disc that catalogues everything in every directory on an ADFS disc.

Of the ADFS utilities supplied on the Welcome disc, the following will be particularly useful:

AFORM - to format a disc

BACKUP - to backup an entire disc

CATALL - to list everything on the disc

COPYFILES - to copy files between the ADFS and other filing systems, e.g. from DFS to ADFS

DIRCOPY - copies files and directories to other ADFS discs

If you boot (Shift-Break) the Welcome disc having chosen the ADFS, you will be presented with a menu that has a built-in help facility: press Shift together with the number for the utility you want, then Return to get the HELP text for that utility. The number without Shift will choose the actual utility.

A disc may reach capacity in two ways. If 47 different files are stored on it, the catalogue space reserved for storing information about the files is used up, and there is no room to list any more, regardless of how much space on the disc is unused. Attempting to put a new file on the disc will result in a 'Cat full' error. If you can delete one file and create a directory instead, you will create 47 more slots within the new directory which can be used to relieve the problem.

On the other hand, you may not have enough disc memory capacity left for the data. In this case you will get a 'Disc full' message. It may be possible to reclaim some space tied up between files by compacting the disc (see Free Space,

Troubleshooting Guide

below) but, as always, remember that using *COMPACT will corrupt the current memory contents.

BACKING UP

It is a good idea to make regular backup copies of your discs, especially if they are regularly updated rather than just being read (e.g. working discs in a database system). You will need the BACKUP program from the utility disc. Make sure you have a good supply of formatted discs, of the correct format, and have both drives of a dual-deck switched to the right number of tracks. If you have only a single drive, the backup program will prompt you to change discs.

Note the following configuration commands, which only apply to the ADFS: *CONFIGURE HARD or *CONFIGURE FLOPPY tells the ADFS which type of drive it should use, if you have a hard disc installed. You can tell it whether or not you want it to read the catalogue of the disc's root directory when initialised using *CONFIGURE DIR (to read) or *CONFIGURE NODIR (to not read).

ADFS files can be locked with a *ACCESS command but in a more complex way than with DFS. Each file can be Readable, Writeable and Locked against erasure/amendment as denoted by R, W and L after its name when the disc is catalogued. Trying to read a file without R, amend a file without W or delete one with L will result in the computer complaining with the message 'Access Violation' or 'Locked'. The symbol D may also appear - this indicates that the name is that of a directory. There is another setting, E, which irreversibly sets a file so that it can only be executed (i.e. cannot be examined, only used) but such a file can be deleted - never set E unless you're certain of what you're doing. To lock or unlock a file use *ACCESS <name> RWL

to lock or *ACCESS <name> RW to unlock; using the latter command should cure 'Access Violation' too.

FREE SPACE AND COMPACTION

The command *MAP gives information on how the regions of free space are spread over the disc. The top line says 'Address : Length'. If nothing appears below it, there is no space left. If a single line of numbers appears, all the free space is in a single block. If more than one line appears, the spare area is fragmented, with different portions between the various files. The computer will not let the number of entries here exceed 80 (it will complain with 'Compaction Required') and when it approaches that number it will begin to say 'Compaction Recommended'.

The compaction command significantly differs from that used in DFS. It always applies to the currently mounted drive, and you must specify which part of the main memory can be used to store data during the process. The command *COMPACT 0E 72 will use what is normally the entire usable memory, but you must be in a shadow mode first (type MODE 135 if you are unsure). You might have ROMs installed which require some memory, so from Basic type PRINT ~PAGE/&100 to find what the first value for the *COMPACT should be, and use:

```
PRINT ~(HIMEM-PAGE)/&100
```

to find the second. If the first value is a single digit you *must* include a zero before it. The *COMPACT command, of course, corrupts anything already in memory. If you know the position and length of some free memory you can use this as a compaction scratchpad and preserve the rest of the memory - see the manual. You should use *COMPACT if the computer gives you either of the compaction messages above, or says 'Map full'. You will not necessarily get the maximum free space first time: type

*MAP, and you will get a revised list. For maximum efficiency continue to compact until there is only one line of numbers.

FORMAT WARNING

Any disc must be formatted before use with the computer, using formatters supplied on the DFS chip or on a utility disc. Most disc formatters will warn you if a disc which is already formatted is inserted, but a DFS formatter will *not* warn you if you put in an ADFS disc (or one from another computer system) to be formatted to DFS standard; the ADFS formatter is equally ignorant of DFS discs.

CHANGING FILING SYSTEMS

A filing system is usually selected by *DISC (or *DISK), *ADFS, *FADFS or *TAPE (there is also *NET for networks - if you have a network, consult your network manager for details). Holding the D, A, F, or Space key while pressing Break effectively perform *DISC, *ADFS, *FADFS or *TAPE respectively - so to boot an ADFS disc from any filing system you can simultaneously press Shift, A and Break. Remember that A-Break may try to read the catalogue from the drive, so make sure there's an ADFS disc there for it to read. In doubt, use F-Break. But remember that the computer can be made to boot on a normal Break and not on a Shift-Break so don't misinterpret this as an error.

BOOTING

You may have a program on disc - typed in from a magazine, copied from tape, or from another programmer - which you want to load by the Shift-Break method, but which has not been set up for such operation. This can easily be arranged on both DFS and ADFS discs. When you *CAT a disc, note the line at the top where it says 'Option' followed by a number. This number is set by the command *OPT 4,*n*. The most common values are *n*=0 (don't boot this disc) and

n=3 (read the words from a command file whose name is !BOOT).

Suppose you normally load the program using CHAIN 'MYProg'. Take the disc the program is on, put it in the drive, *MOUNT it if it's ADFS and then *CAT it. If there is a file on it called !BOOT, check to see what the Option is; there may be a boot file already which isn't used because the option is wrong. If there's no !BOOT, you can proceed as follows. Remove the write-protect tab if fitted (on ADFS, you needn't bother with DISMOUNT and MOUNT since it's the same disc you're removing and replacing) and type the following:

```
*BUILD !BOOT
*BASIC
CHAIN 'MYProg'
(or whatever your usual instruction is)
```

Press Escape to close the !BOOT file.

```
Type:
*OPT 4,3
```

The computer will supply line numbers which I haven't shown. The disc is now correctly configured. If you want to read an existing !BOOT file, type *LIST !BOOT.

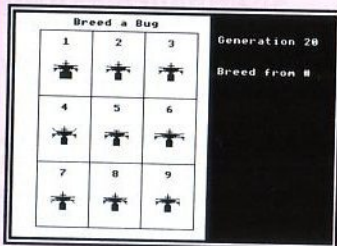
CASSETTE TO DISC

When a cassette program for a Beeb is transferred to disc, it may give the error 'No Room' when run. This is a sign of lack of memory: when a Beeb runs a disc system, memory which was free to hold programs for cassette is claimed by the DFS. In this case you will need a 'movedown' routine: see, for instance, BEEBUG Vol.8 No.8 p.59.

SIDEWAYS RAM

You can adapt a disc to install a file in sideways RAM automatically, in the following way. Take the disc with the file on it (let's suppose it's called RAMFILE) and *CAT it. If there is already a file called !BOOT on this disc, copy RAMFILE to a blank disc, and work on that. If there isn't,

Continued on page 30



- PERSONALISED ADDRESS BOOK** - on-screen address and phone book
- PAGE DESIGNER** - a page-making package for Epson compatible printers
- WORLD BY NIGHT AND DAY** - a display of the world showing night and day for any time and date of the year

Applications I Disc

- BUSINESS GRAPHICS** - for producing graphs, charts and diagrams
- VIDEO CATALOGUER** - catalogue and print labels for your video cassettes
- PHONE BOOK** - an on-screen telephone book which can be easily edited and updated
- PERSONALISED LETTER-HEADINGS** - design a stylish logo for your letter heads
- APPOINTMENTS DIARY** - a computerised appointments diary
- MAPPING THE BRITISH ISLES** - draw a map of the British Isles at any size
- SELECTIVE BREEDING** - a superb graphical display of selective breeding of insects
- THE EARTH FROM SPACE** - draw a picture of the Earth as seen from any point in space

File Handling for All

on the BBC Micro and Acorn Archimedes
by David Spencer and Mike Williams



Computers are often used for file handling applications yet this is a subject which computer users find difficult when it comes to developing their own programs. *File Handling for All* aims to change that by providing an extensive and comprehensive introduction to the writing of file handling programs with particular reference to Basic.

File Handling for All, written by highly experienced authors and programmers David Spencer and Mike Williams, offers 144 pages of text supported by many useful program listings. It is aimed at Basic programmers, beginners and advanced users, and anybody interested in File Handling and Databases on the Beeb and the Arc. However, all the file handling concepts discussed are relevant to most computer systems, making this a suitable introduction to file handling for all.

The book starts with an introduction to the basic principles of file handling, and in the following chapters develops an in-depth look at the handling of different types of files e.g. serial files, indexed files, direct access files, and searching and sorting. A separate chapter is devoted to hierarchical and relational database design, and the book concludes with a chapter of practical advice on how best to develop file handling programs.

The topics covered by the book include:

- Card Index Files, Serial Files, File Headers, Disc and Record Buffering, Using Pointers, Indexing Files, Searching Techniques, Hashing Functions, Sorting Methods, Testing and Debugging, Networking Conflicts, File System Calls

The associated disc contains complete working programs based on the routines described in the book and a copy of Filer, a full-feature Database program originally published in BEEBUG magazine.



ASTAAD

Enhanced ASTAAD CAD program for the Master, offering the following features:

- * full mouse and joystick control
- * built-in printer dump
- * speed improvement
- * STEAMS image manipulator
- * Keystrips for ASTAAD and STEAMS
- * Comprehensive user guide
- * Sample picture files

	Stock Code	Price
ASTAAD (80 track DFS)	1407a	£ 5.95
Applications II (80 track DFS)	1411a	£ 4.00
Applications I Disc (40/80T DFS)	1404a	£ 4.00
General Utilities Disc (40/80T DFS)	1405a	£ 4.00
Arcade Games (40/80 track DFS)	PAG1a	£ 5.95
Board Games (40/80 track DFS)	PBG1a	£ 5.95

	Stock Code	Price
ASTAAD (3.5" ADFS)	1408a	£ 5.95
Applications II (3.5" ADFS)	1412a	£ 4.00
Applications I Disc (3.5" ADFS)	1409a	£ 4.00
General Utilities Disc (3.5" ADFS)	1413a	£ 4.00
Arcade Games (3.5" ADFS)	PAG2a	£ 5.95
Board Games (3.5" ADFS)	PBG2a	£ 5.95

All prices include VAT where appropriate. For p&e see Membership page.

Board Games

SOLITAIRE - an elegant implementation of this ancient and fascinating one-player game, and a complete solution for those who are unable to find it for themselves.

ROLL OF HONOUR - Score as many points as possible by throwing the five dice in this on-screen version of 'Yahtzee'.

PATIENCE - a very addictive version of one of the oldest and most popular games of Patience.

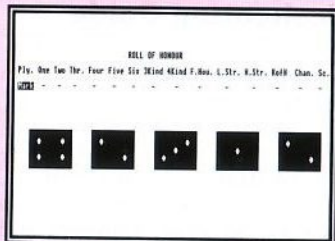
ELEVENSES - another popular version of Patience - lay down cards on the table in three by three grid and start turning them over until they add up to eleven.

CRIBBAGE - an authentic implementation of this very traditional card game for two, where the object is to score points for various combinations and sequences of cards.

TWIDDLE - a close relative of Sam Lloyd's sliding block puzzle and Rubik's cube, where you have to move numbers round a grid to match a pattern.

CHINESE CHEQUERS - a traditional board game for two players, where the object is to move your counters, following a pattern, and occupy the opponent's field.

ACES HIGH - another addictive game of Patience, where the object is to remove the cards from the table and finish with the aces at the head of each column.



Applications III Disc



SHARE INVESTOR - assists decision making when buying and selling shares

LABEL PROCESSOR - for designing and printing labels on Epson compatible printers

CROSSWORD EDITOR - for designing, editing and solving crosswords

MONTHLY DESK DIARY - a month-to-view calendar which can also be printed

3D LANDSCAPES - generates three dimensional landscapes

REAL TIME CLOCK - a real time digital alarm clock displayed on the screen

RUNNING FOUR TEMPERATURES - calibrates and plots up to four temperatures

JULIA SETS - fascinating extensions of the Mandelbrot set

FOREIGN LANGUAGE TESTER - foreign character definer and language tester

Arcade Games

GEORGE AND THE DRAGON - Rescue 'Hideous Hilda' from the flames of the dragon, but beware the flying arrows and the moving holes on the floor.

EBONY CASTLE - You, the leader of a secret band, have been captured and thrown in the dungeons of the infamous Ebony Castle. Can you escape back to the countryside, fighting off the deadly spiders on the way and collecting the keys necessary to unlock the coloured doors?

KNIGHT QUEST - You are a Knight on a quest to find the lost crown, hidden deep in the ruins of a weird castle inhabited by dangerous monsters and protected by a greedy guardian.

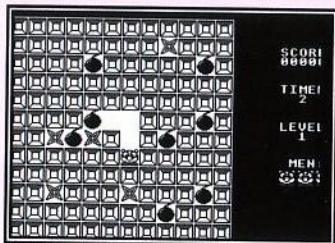
PITFALL PETE - Collect all the diamonds on the screen, but try not to trap yourself when you dislodge the many boulders on your way.

BUILDER BOB - Bob is trapped on the bottom of a building that's being demolished. Can you help him build his way out?

MINEFIELD - Find your way through this grid and try to defuse the mines before they explode, but beware the monsters which increasingly hinder your progress.

MANIC MECHANIC - Try to collect all the spanners and reach the broken-down generator, before the factory freezes up.

QUAD - You will have hours of entertainment trying to get all these different shapes to fit.



	Stock Code	Price
File Handling for All Book	BK02b	£ 9.95
File Handling for All Disc (40/80T DFS)	BK05a	£ 4.75
Joint Offer book and disc (40/80T DFS)	BK04b	£ 11.95
Magscan (40 DFS)	0005a	£ 9.95
Magscan (80T DFS)	0006a	£ 9.95
Magscan (3.5" ADFS)	1457a	£ 9.95

	Stock Code	Price
File Handling for All Disc (3.5" ADFS)	BK07a	£ 4.75
Joint Offer book and disc (3.5" ADFS)	BK06b	£ 11.95
Magscan Upgrade (40 DFS)	0011a	£ 4.75
Magscan Upgrade (80T DFS)	0010a	£ 4.75
Magscan Upgrade (3.5" ADFS)	1458a	£ 4.75

All prices include VAT where appropriate. For p&p see Membership page.

Tel. (0727) 840303

Fax. (0727) 860263

Best of BEEBUG

Troubleshooting Guide

but there's a write-protect tab, then you must temporarily remove it.

Take the disc without !BOOT but containing RAMFILE, and type *BUILD !BOOT. You will (as when creating the simple boot file) be given a prompt with a number 1, where you should type in the following:

```
*SRLOAD RAMFILE 8000 4 Q
*||
*|| Now do Ctrl-Break
*|| Then activate your ROM
```

Press Escape to close the !BOOT file.

Type:

```
*OPT 4,3
```

The *|| code tells the computer to ignore everything typed after it - this enables you to put comments into the file, in this case to remind you to perform CTRL-Break. While CTRL-Break must be pressed to initialise the RAM, it resets everything, so

the user must be reminded what to do him/herself. You can set up as many as four such sideways RAM packages at once by this method - after *SRLOAD RAMFILE 8000 4 Q, you should type *SRLOAD NEXT 8000 5 Q, for instance - and so on, for up to four files.

THANK YOU AND GOOD NIGHT

Over the five articles, I hope I have been able to shed some light on some of the problems you may have experienced, whether it is a tip on fixing a bent pin in a disc lead or merely the reassurance that a sideways ROM image is failing because your computer is set up with RAM deselected rather than because you are doing anything wrong. There are many faults which can occur with each piece of hardware or software and I can't possibly anticipate every possible problem, but perhaps I can save you an occasional call to your dealer. And if all else fails, don't forget that BEEBUG has a technical helpline: St Albans (0727) 840303. **B**

Robol - The Screen Designer (continued from page 10)

```
2220 LDA addr+1:ADC #0:STA addr+1
2230 DEX:BNE loop2
2240 :
2250 .store_addr
2260 LDA addr:STA scr_addr:LDA addr+1:S
TA scr_addr+1
2270 LDA index:STA spr_addr+1:LDA #&0C:
STA spr_addr+2
2280 :
2290 LDX #2:LDY #16:STX column:STY row
2300 LDX #0:LDY #0
2310 LDA scr_addr:STA temp:LDA scr_addr
+1:STA temp+1
2320 .display
2330 LDA row:STA temprow
2340 .spr_addr
2350 LDA &FFFF,X:STA (scr_addr),Y
2360 INX:LDA scr_addr:AND #7
2370 CMP #7:BEQ bottom
2380 INC scr_addr:BNE nextrow
2390 INC scr_addr+1:JMP nextrow
2400 .bottom
2410 LDA scr_addr:ADC #&38:STA scr_addr
```

```
2420 LDA scr_addr+1:ADC #1:STA scr_addr
+1
2430 .nextrow
2440 DEC temprow:BNE spr_addr
2450 LDA temp:ADC #8:STA scr_addr:STA t
emp
2460 LDA temp+1:ADC #0:STA scr_addr+1:S
TA temp+1
2470 DEC column:BNE display
2480 RTS
2490 ]
2500 NEXT
2510 ENDPROC
2520 :
2530 DEF FNis(N%):LOCAL F%
2540 R$="rob"+RIGHT$("00"+STRN%,2):F%=
OPENIN(R$)
2550 IF F%=0 VDU7:PRINT'CHR$129;"There
is no"CHR$130;R$;CHR$129;"file on the di
sc!'"CHR$130"Press any key ";:REPEATUNT
ILGET :=FALSE
2560 CLOSE#F%:=TRUE B
```



Sound (1)

This month Alan Wrigley starts a new series of 1st Course articles which take a look at sound on the BBC micro.

Sound tends to be a poor relation to graphics for those who dabble in programming. Often much effort is spent in producing pretty graphics, while sound is either ignored entirely or restricted to a few beeps here and there. Perhaps this is a result of our TV-oriented culture, which demands ever-more arresting images in order to grasp our attention, or perhaps it is because the production of exciting sound patterns on a micro is perceived to be more difficult than stunning graphics.

The fact of the matter is that the BBC micro's sound system, as with most of its other features, was designed to be as powerful and versatile as possible while still allowing full control of its facilities from the Basic language.

SOUND BASICS

Before we launch into a description of sound on the BBC micro, we need to explain some terminology. Any sound that we hear has a number of characteristics which enable us to distinguish it from any other sound, and which help us to simulate it electronically. The *amplitude* determines how loud the sound is (its volume), while the *pitch* describes how high or low the sound appears to be. We could also describe the pitch in terms of *frequency*; the higher the pitch, the higher the frequency of the sound wave. Many natural sounds are quite complex and contain components at different pitches - needless to say these can be difficult to simulate with simple electronic circuits.

Simple sounds also have a characteristic *waveform*, which is determined by the number and amplitude of the harmonics present (harmonics are components of the sound at a different frequency from the fundamental and at a reduced amplitude, and are a vital element of most sounds).

Most of you will have heard the flat, monotonous sound of a sine wave, which is the purest form of sound with no harmonics at all. A square wave on the other hand (which is the easiest to produce electronically) has large numbers of harmonics, some of them discordant, and thus has more of an edge to it. Simple electronic circuits rarely offer the facility to alter the harmonic parameters, and so the waveform is likely to be fixed by the designer of the circuit.

Many natural sounds contain an element of noise. This refers to sound which is made up of a large number of frequencies at the same time, with none being predominant. An example would be the hiss you hear between stations when tuning an FM radio. Noise which covers the whole frequency spectrum is referred to as *white noise*, and is usually created electronically by mixing random amounts of many frequencies. Sometimes this can be filtered to produce noise within a certain frequency band, and this is known as *pink noise*. Mixing noise with other sounds can often enhance their naturalness. The BBC sound chip provides both types of noise.

The other major parameter of a sound which is relevant to BBC users is its *envelope*. An amplitude envelope can be thought of as a graph of amplitude against time - in other words it describes how the sound rises and decays over a period of time. This can be very important indeed, since many quite different natural sounds can be difficult to distinguish from one another if their characteristic envelope is removed. Some sounds will also change their pitch over time; in such cases a pitch envelope can be specified as well as an amplitude envelope. Both types of envelope can be produced on a BBC micro, using a specially-provided keyword, as we shall see later in this series.

SOUND PRODUCTION

Sound can be produced on a micro in several different ways. The simplest method is to provide a straightforward oscillator circuit which beeps, at either a fixed or a programmable pitch. Such a system will have no means of specifying envelopes, and may not even have any kind of amplitude control. The most complex method of sound control is to allow the software to create the actual waveforms which are output by the sound system. This method is adopted by the Archimedes, for instance, and in theory allows the micro to reproduce faithfully *any* sound or combination of sounds, natural or artificial. Given enough memory, you can play a symphony on an Archimedes by sampling the stored data and sending it to the sound system at the required rate.

The BBC steers a course between these two extremes. Sound is produced by a dedicated sound chip - the kind of thing you find in musical doorbells but rather more versatile. This means that the waveform of the sounds you can produce is fixed within the design of the chip, but to compensate for this, several channels are provided to enable polyphonic sounds to be produced, including a channel specifically allocated to noise. Furthermore, BBC Basic has two commands which allow you to alter not only the pitch and amplitude, but also both types of envelope. We will look at the ENVELOPE command later, but the rest of this month's article will concentrate on simpler ways to produce a sound.

SIMPLE SOUNDS

The very simplest sound of all is the bell sound, produced by typing Ctrl-G or pressing the Copy key while at the Basic prompt, or by issuing a VDU7 command. You may have thought that the sound of the bell was fixed, but in fact you can alter the pitch, amplitude and duration, and even apply an envelope to it. This is done by using the calls *FX211 to *FX214. If you find the default beep annoying, for

example, you could lower its pitch by entering *FX213,35 and shorten it by entering *FX214,3.

The SOUND command forms the basis for the control of sound from Basic, and can be used either on its own or in conjunction with the ENVELOPE command which we will discuss later. On its own it can produce a range of sounds with a fixed envelope, but with the pitch, amplitude and duration all programmable. The simplest form is:

SOUND C,A,P,D

where C is the channel number (0-3); A is the amplitude (from 0 for no sound to -15 for full volume - thus it is always a negative number); P is the pitch (0-255); and D is the duration in twentieths of a second (0-254, or -1 for continuous). Because the sound is produced by a separate chip, the operation of the computer is not halted during the sound; once the command has been issued, the computer gets on with the next instruction even if the note is still playing.

Channel numbers 1-3 produce a square wave sound. For single or consecutive notes you can use any of the three, though it is usual to use channel 1 in such cases. Channels 2 and 3 come into their own when we want to indulge in polyphony, as we shall see later. Channel 0 is the noise channel, and for this the value of P in the SOUND command has a different meaning. We will look at noise next month.

The A and D parameters of the SOUND command are fairly self-explanatory, but P(itch) needs a little further description. Each value of P between 0 and 255 represents a quarter of a semi-tone; in other words, a value of 96 is a semi-tone higher than 92. Middle C is 53, so from that basis you can work out all the other notes that are encompassed by the 0-255 range (around 5 Ω octaves). Since there are 12 semi-tones in an octave, each octave spans 48 units. Putting all this detail together:

SOUND 1,-15,101,5

sounds a note on the C above middle C at maximum volume for a quarter of a second. Table 1 shows the pitch value for each possible note that you can produce.

Note	Octave						
	1	2	3	4	5	6	7
C		5	53	101	149	197	245
C#		9	57	105	153	201	249
D		13	61	109	157	205	253
D#		17	65	113	161	209	
E		21	69	117	165	213	
F		25	73	121	169	217	
F#		29	77	125	173	221	
G		33	81	129	177	225	
G#		37	85	133	181	229	
A		41	89	137	185	233	
A#		45	93	141	189	237	
B	1	49	97	145	193	241	

Table 1 Values of pitch for different notes

Using this information, it is very simple to turn your computer keyboard into a single-octave instrument. If we want "Z" to represent middle C, then "," will produce the C above it, and the keys in between will produce the remaining notes of the octave. Sharps can also be produced by using "S", "D", "G", "H" and "J". The following program will achieve this (it assumes you have the Caps Lock key off):

```

10 C%=53
20 REPEAT A$=GET$
30 Z%=INSTR("zszdcvghbnjm,",A$)
40 IF Z% SOUND 1,-15,C%+(Z%-1)*4,5
50 UNTIL FALSE
    
```

Because the relationship between the pitch of the notes is fixed, you can play the keyboard in any key just by changing the value assigned to C% in line 10, though the sharps will not then correspond with the positions of the black keys on a conventional keyboard.

You will see that this does not behave like a normal instrument keyboard in that the duration of each note is fixed irrespective of when you release the key. To mimic a proper instrument you would need to detect the release of the key, which is beyond the scope of this article.

I mentioned earlier that a value of -1 for D gives a continuous note. Such a note will continue until either Escape is pressed or another note is played on the same channel. If you modify the listing above as follows:

```

40 IF Z% SOUND 1,-15,C%+(Z%-1)*4,-1
    
```

each note will continue to sound after you release the key until you press another.

This simple use of the SOUND command enables you to play tunes in your programs very easily. All you need to do is to work out the values and durations of the notes, place them in one or more DATA statements, and then READ these values in a loop which contains the SOUND statement. Try the following example:

```

10 FOR I%=1 TO 30:READ P%,D%
20 IF P%=-1 A%=0 ELSE A%=-15
30 SOUND 1,A%,P%,D%
40 TIME=0:REPEAT UNTIL TIME>=D%*5+1
50 NEXT
60 DATA 81,8,69,4,73,4,81,4,69,4,53,12
70 DATA -1,4,33,4,53,4,53,4,49,4,49,4
80 DATA 53,4,61,4,69,4,73,16,-1,8,73,8
90 DATA 61,4,69,4,73,4,77,4,81,16,73,8
100 DATA 69,4,81,2,69,2,53,8,61,8,53,16
    
```

There are two comments to make about this little program. Firstly, line 40 ensures that the notes do not cut each other off or slur into each other, by waiting until 1 centisecond beyond the end of each note before playing the next. TIME on the BBC micro is incremented in 100ths of a second, while duration values are in 20ths. Thus D%*5 is equal to the duration of the note in 100ths. Secondly, in a couple of places the pitch value has been given as -1. This is a convenient way to produce a pause between notes, by picking up the -1 in line 20 and setting the amplitude to 0 (silence); thus no sound will be generated during the period specified.

Using Table 1 you should find it easy to program your own tunes in a similar way. You may also find the following function useful, which gives pitch from frequency:

```

DEF FNpitch(freq)
=INT(159.453*LOG(freq)-283.631)
    
```

Public Domain Software

This month's PD column looks at a new Basic program editor and some more golden oldies from BEEBUG volume 2.

A SHORT FABLE

Some months ago now, I received a letter from Richard Taylor, who had written a Basic program editor in the form of a ROM image. I was pleased when he agreed to release the program as shareware, as it is a very professional piece of software.

There are various ways of editing programs written in BBC Basic, ranging from the simple cursor copying and editing facilities built into the micro, through various means of transferring the program into another format so that it may be edited in another program, like View, Wordwise or (in the case of the Master Series), the EDIT text editor. An alternative which became available some years ago was Acornsoft's 'Basic Editor', a ROM-based program which allowed programs to be edited in their native format, without conversion to plain ASCII text. This approach has a lot of advantages. It provided word processor-like facilities superior to the simplest cursor editing without the need for intermediate file saving and loading. Also, there was no need to hold two copies of the program in memory at the same time (one tokenised Basic, the other plain ASCII).

Richard's system, FABLE (an acronym for 'Fast Advanced Basic Line Editor'), takes the same approach as the Acornsoft ROM, but applies the pull-down menu style of a program like Computer Concepts' Interword, a very popular and usable system. Its feature list is impressive, including:

- easy link to Basic, with optional Escape toggling and return to edit mode when an error occurs during a run of the program;

- pop-up menu system, with quick control-key alternatives;
- preferences and the current state of the editor can be saved, allowing it to be customised;
- search & replace, with optional case sensitivity and wildcards;
- marked section copy, delete, renumber and print functions.

Some of the features noted briefly above are more impressive when you use them than when you read about them, for example the Basic link system. If the option 'Return on Errors' is ticked on the 'BASIC Link' pop-up menu, any Basic error which is generated whilst a program is running will cause FABLE to be entered with the cursor on the offending program line, ready to be edited. Using this and other options, a program under development can be edited and tested frequently and easily, and the effects of a program change can be seen almost instantly.

There are some limitations to the program, depending on the machine you use. For Electron and Model 'B' owners, the most important is likely to be that the program only works in Mode 0, so unless you have shadow RAM fitted, the maximum size of program which can be edited is limited. For Master users, the icon representations of program files in the load/save menus do not appear unless the files have an execution address of &8023, the default value for Basic II but not for Basic IV, and the extra keywords introduced in Basic IV are not recognised by the system (although I for one almost never use them). Basic

programs without the correct icon will still load and save correctly, however.

If you are feeling ambitious, Richard has been kind enough to allow the full source code for FABLE to be distributed, so it would be possible to add extra facilities or correct the 'Master icon' problem referred to above if you felt the desire and need. Five pages' worth of instructions are also included in the form of a spooled text file on the disc.

Richard actually marketed the program himself some time ago, but was a victim of timing when the market for commercial software for 8-bit machines was on the decline. However, it is a most professionally put together piece of work and deserves to be seen by a wider audience than has made use of it so far. The shareware payment requested is £5.00, which I think is very reasonable. I have already registered my copy.

GOLDEN OLDIES

Following my quick tour of the magazine discs from BEEBUG volumes 3, 4 and 5, I can now do the same for volume 2 thanks to the quick and helpful reactions of readers who saw a brief request for copies of those back issues which I was missing. Thanks to everyone who sent me some or all of the software from the original tapes for volume 2; if anyone has the software from volume 1 transferred from the original tape to disc, I would be grateful for a copy to complete the collection. It would be good to keep all that software from becoming unavailable, as I still regularly hear from new users who have acquired a secondhand BBC Micro and are having difficulty locating software.

As well as looking through the software, I dug out my copies of the early magazines to refresh my memory. How different our concerns were in those days! Looking through the December 1983 issue, I remembered exactly why the

magazine software was only available on tape. Cumana got a good review of their new disc drives, just introduced: but I wonder just how many people could afford to buy a (now standard) 80 track, double-sided disc drive at a price of £395.60 including VAT? if you were feeling really flush, you could splash out on a double drive unit, a snip at £734.85!

Enough of giving away my age; on to the software. Volume 2 saw the introduction of the CAD (Computer Aided Design) program everyone must know - ASTAAD. This program has changed quite a lot over the years, but the original was quite radical for its time - it refused to work with anything less than OS 1.2 which, I suppose, must then have been like demanding an Arc with 2Mb of RAM these days! Most software listings from that time guaranteed compatibility with OS 0.1 and OS 1.0, and if they didn't work with Basic I, there had to be a good reason why not.

Generally speaking, this was the time when programming exploration and creativity were reaching their peak on the BBC. A few techniques hadn't yet been invented or become as widely understood, but the mix of software and the quality were good. Volume 2 sported a greater proportion of programs which demonstrated the features of the computer than appear now, which is a source of nostalgia for us old hands and is something new if you come across it now for the first time. Most issues featured two games, usually of different styles, such as 'Killer Dice' (Poker) and 'Galactic Invasion', or 'Block Blitz' and 'RayBox'. Apart from the regular games, though, the mix is as wide as you would expect from BEEBUG - anything from Home Accounts to a Bach Cantata, from 3D Bar Charts to a Percussion Machine.

Note: BEEBUG programs earlier than volume 6 are only available as PD software through BBC PD, not from BEEBUG. **B**



512 Forum

Robin Burton takes a look at what's involved in adding a winchester (hard disc) to your Beeb, a topic relevant to all BBC micro owners.

The subject of this month's Forum has produced a steady trickle of queries over a long period of time.

Naturally it's of particular interest to 512 users, but it's useful information for any BBC micro user. Oddly, and despite that, this must be the least well covered topic ever for the BBC micro.

WANT A WINCHESTER?

When a BBC micro user acquires a 512 there's delight at the huge range of software which is available, and pleasure at the freedom offered by the vast amount of memory in the machine, at least compared with the 8-bit BBC micro.

Some BBC programs are quite remarkable given the restricted environment in which they run, but no matter how capable a program is, a major irritation that virtually everyone encounters is the limit which memory size imposes on data and file handling. You only have to sort a simple database of moderate size to become all too aware of this.

Of course strategies exist to maximise flexibility, such as multiple files for unstructured data (e.g. Inter-Word) or random access for keyed records in databases. These work, but they can be very tedious, especially when you're using floppy discs. A winchester (hard disc) certainly speeds file access and provides more space, but given cost and the limitations of the micro, this

has, not surprisingly, been viewed as an expensive luxury by most users.

Even an unexpanded 512 can load a program of 200K plus a file of 150K into memory simultaneously, so to the 512 user this freedom is initially wonderful. Equally the added speed and capacity of 512 800K discs is also welcome, but as time passes the limits of disc performance again impose themselves. Loading an application and a file of the sizes above is a long way short of being instantaneous if you're using floppies.

Worse, 512 users can find their ambitions restricted or sometimes totally frustrated. Quite a few PC programs simply won't install on floppies, while others will install, but can't be used effectively. 512 users can be in a very unpleasant position. Floppy disc limitations are more acute than in the BBC micro, but the cure is no easier.

For PCs, adding a winchester is relatively simple and cheap. Fit a controller card into the main board at a cost of about £20.00, plug in the hard disc, tell the BIOS about the drive and start to use it. Of course capacities and prices vary, but £200.00 easily buys a drive of 100Mb. these days.

For the BBC micro things are neither simple nor cheap. As was explained for display adaptors two issues ago, the design concepts of the BBC differ markedly from a PC, in consequence of

which adding or changing bits has always been more difficult even if it's possible.

HOW DO THEY WORK?

A BBC micro hard disc is probably the least well understood upgrade and, as I mentioned this has tended to remain so because BBC micro magazines have rarely if ever explained what's involved. It's obvious when you have a hard disc, but judging by letters I've received a fair number of users haven't but would like to know. We'll start from basics for any recent Beeb converts.

First, to use a winchester you need ADFS, no problem for 512 users who of course must have ADFS anyway. For BBC users with only DFS who are reading this and might want to add a hard disc, ADFS is the first requirement.

Note: If you have an early DFS which uses the 8271 controller, both the software and hardware must be upgraded. ADFS requires a WD1770/1772 controller, already fitted in the B+ and Master. In an 8271 based machine you need a 1770 upgrade therefore, but finding one might take persistence these days (still available through BEEBUG Ltd.). The 1770 chip can run DFS, so your existing discs aren't scrap, but remember that you must upgrade the DFS too. The way the two controllers are programmed is totally different.)

If you have ADFS, at its simplest adding a winchester means plugging a ribbon cable into the 1MHz bus, located underneath the machine along with the floppy disc port, the parallel connector, the user port and the Tube. Naturally this assumes that you have a suitable

hard disc ready, complete with all its bits - so what are they?

The 1MHz. bus isn't an industry standard, it's an 'Acorn special', so there's no such thing as a 1MHz bus hard disc. The Beeb hard disc interface was therefore custom designed for a non-standard connection. In truth, although it works well enough a BBC hard disc interface is frankly a lash-up, though perhaps it's kinder to call it an afterthought. It should be remembered that few micros, even PCs, had hard discs as standard when the BBC micro appeared.

Winchesters for the BBC micro are therefore 'standard' drives with extra circuitry to allow connection through the 1Mhz bus. To explain the various items and their role we'll follow the route from the hard disc back towards the micro so that you can see what's involved, with a bit of history for interest.

THE HARDWARE

The drive cartridge is a standard unit (of five to ten years ago remember) of any make, though Rodime is probably most common. The drive mechanism has its own control board (for motor and head stepping) physically attached to it and both together are regarded as one unit. This unit is the disc cartridge, which could be fitted to almost any micro of the time. Two types of data recording were common, MFM and RLL, though you don't need to understand these so long as the next link in the chain is the correct type.

Moving towards the micro, the next item is a circuit which translates logical disc commands issued by a micro into commands understood by the

cartridge's control circuits. In the early '80s the only suitable standard interface was SCSI (Small Computer System Interface, pronounced 'scuzzi') which was directly descended from SASI (Shugart Associates System Interface). SASI was conceived as a device independent interface which could handle virtually any peripheral, hard discs were simply its most frequent application. Although Shugart designed the interface, for general use a company name wasn't acceptable, hence the change.

Probably the most common SCSI card for discs was the ST506 and that's what BBC winchesters use. Of course in a PC the hard disc controller speaks directly to the SCSI board in its own language and that's all there is to it - not in the BBC. Since the BBC micro has no hard disc controller (and can't have one) something more is needed. This is the lash-up.

An extra circuit, variously called an Acorn adaptor or an Adaptec board (after the company that manufactured them for Acorn) is required to translate I-O commands issued by ADFS through the 1MHz bus into SCSI commands that the ST506 can understand. The Acorn adaptor therefore connects to the ST506 on the hard disc side and to the 1MHz bus on the micro side.

You might wonder why ADFS doesn't 'speak' SCSI itself. There are two reasons. First, the 1MHz bus was inherited from the model B and was never originally intended for a winchester, so its physical properties and signals are quite incompatible with SCSI. Second, even if the physical connections were correct there isn't

enough room in ADFS's 16K for the necessary code plus floppy disc control.

The Adaptec board therefore translates elementary requests from ADFS into more complex instructions for the SCSI board, as well as providing the correct physical connections. If you look at an Adaptec board you'll see that it contains a couple of EPROMS, a processor of its own and a large number of support chips - this is quite a complicated task. Of course that's not all it has to do, it also provides handshaking and data buffering in both directions, for which the micro also has no provision.

As a result of this considerable complexity and the fact that Acorn adaptors were a low volume item (compared with micros) they were fairly expensive. I can't remember prices now, but any extra complexity obviously adds to cost. Coupled to that, SCSI is expensive, in the context something of a sledge-hammer to crack a walnut as it were, for which reason PC hard discs inevitably soon adopted a different approach.

Hard drives themselves have improved in performance over the years of course, but the costs of SCSI are now avoided. These days the standard for PC hard discs is IDE (Integrated Drive Electronics). All control circuitry is built into the single board mounted on the disc cartridge and, since hard disc control is a simple and highly defined task, much of the complexity and cost of a general purpose interface like SCSI is avoided.

The best current IDE drives operate at data rates of up to about 1.5Mb/second, but this isn't enough for some jobs. For this reason SCSI is still employed in PCs,

but generally only when large drives (typically half a gigabyte upwards) and very high (circa 6Mb/sec) data throughput is required, a common need for main storage for network file-servers. Naturally large drives are still expensive and, since they form only a small section of the market, SCSI interfaces are still relatively costly too.

Unfortunately (that word again) the fact that a BBC winchester uses SCSI doesn't mean it will be fast by today's standards. Apart from the age, hence performance of the drive, the BBC micro's 1MHz bus has a maximum instantaneous throughput of only 125Kb/sec, which in practice probably means an effective rate of half this figure. Even so this is equivalent to reading an entire 640K disc in about ten seconds, so it's a huge improvement on floppies and is still a worthwhile step.

STUMBLING BLOCKS

OK, so Beeb hard discs use a bit of special hardware, but why aren't they easier to get hold of? The answer is paradoxical, the cause is historical. (that's this month's intellectual bit over with!).

When model B micros were current you couldn't add a winchester. That had to wait for ADFS, which although offered as an upgrade when the B+ appeared, was primarily produced for the launch of the Master.

The addition of a winchester was therefore a late option in the BBC micro's history. Unfortunately (CANCEL) - It was just bad luck that when it did become possible the interface used in PCs was changing (to ESDI, now itself superseded by IDE) so the drive type and interface was no

longer in the mainstream, with attendant cost penalties.

Using an SCSI interface drives had always been pretty expensive, but by the time the BBC micro employed it, SCSI was no longer the normal approach, so prices didn't fall as they normally do after a few years for most electronics. You could pay over £400.00 for even a 10Mb BBC drive six or seven years ago, so obviously it was never the most popular add-on even though it was quite probably the most desired one.

SUPPLIERS

These days, the 8-bit BBC micro is of little interest to most Acorn suppliers and, to the best of my knowledge, none can now supply BBC winchesters. Those who specialise in hard discs (e.g. Oak, Morley, etc) might have a few spares left, but I wouldn't bank on it.

This leaves prospective purchasers with second-hand sales. Naturally, users who purchased a hard disc years ago were real enthusiasts, so it's likely a good many of them still use their Beeb with no intention of parting with it, or the hard disc, for some time. Other hard disc users, such as businesses or schools particularly, can be expected to keep machines until they fall to pieces, so they're not much help either.

A quick scan of Beebug personal ads. over a few issues will show just how rarely hard discs are offered, in fact I wouldn't be surprised if 'Wants' outnumbered 'Sales'. Apart from a good supply of patience and a lucky find in personal ads (still expect to pay about £200.00 for a working 20 or 30Mb drive) what are the other options for getting hold of a drive?

Well, you might find one for sale that doesn't work, so it will be cheaper, but then what about spares and repairs? Of course that depends on the fault, but be aware that there is no 'standard' or likely fault, the problem might be anywhere. For BBC use the components of a hard drive are a case, a PSU, an Acorn adaptor, an ST506, a drive cartridge plus connectors and cables. Depending on which parts you have and which you need, sources for spares vary from scarce to virtually non-existent.

There are a few advertisers of PC hardware surplus in PC magazines and some offer new 'old stock' 10, 20 or 30Mb drives, with prices of well under £100.00 for 20Mb. If you buy one of these and add the other bits you might be in business. Alternatively you could cannibalise an old, expired XT with a working hard drive and start from there, which might be even cheaper.

For the remaining parts, cables, plugs and sockets are no problem of course, but cost about £30.00 for a complete set. PSUs aren't totally impossible and failed units can be repaired if you know the right people. Acorn dealers no longer supply Acorn adaptors and cases have always been a problem, but it might be worth asking your local dealer about these, you never know.

HELP? - MAYBE!

All in all, acquiring a hard disc for a BBC micro isn't easy. It was never cheap and, even for used components it still isn't. Consider! A hard disc probably cost its original buyer as much if not more than the micro. Because of that there aren't many of

them, so no-one needs to sell one at a bargain price. The bottom line, however you go about it, is that you'll be very lucky to get a working hard disc for a BBC for less than £200.00 all in, assuming you find one, or the appropriate parts.

I know of a limited source of Adaptec and ST506 boards and the people concerned (who do not wish to be contacted directly) occasionally come across cartridges, PSUs and cases. They can supply cables and will even build a drive for you if you have (most of) the bits. They can sometimes repair faulty drives or components, but take note that a faulty drive mechanism, if that's the diagnosis, is forever a lost cause. Even the original manufacturers either can't or won't repair them!

If anyone has a go at building a hard drive but gets stuck or has a faulty drive you can write me via Essential Software and I'll help if I can. Here are the rules.

Be very sure to explain precisely what parts you have and what parts or help you need. If you have a faulty drive supply as much info. as possible. Do include an SAE, but do not ask about a complete hard disc supply service - there isn't one!

Finally, anything you can do for yourself will be cheaper. You already know what the total cost of a drive is likely to be but, sorry to be brutal, while the help is free, phone calls, postage and the time of other people isn't, so include £2.00, payable to me, to cover initial costs.

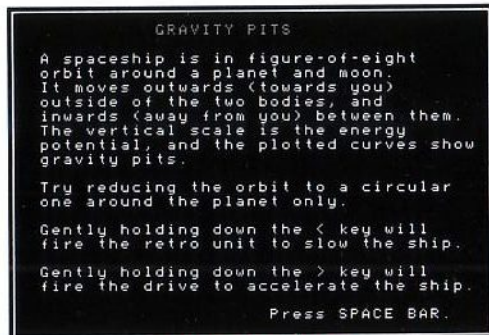
Address: Robin Burton, P.O.Box 5, Groby, Leicester LE6 0ZB.

B

Gravity and Orbits (Part 3)

This month Cliff Blake takes us into the depths of Gravity Pits.

A rocket ship climbing away from a planet can be compared to a jet-propelled and brakeless car climbing a hill. Near the planet the hill is very steep, but the incline becomes less as the distance covered becomes greater.



Initial instructions

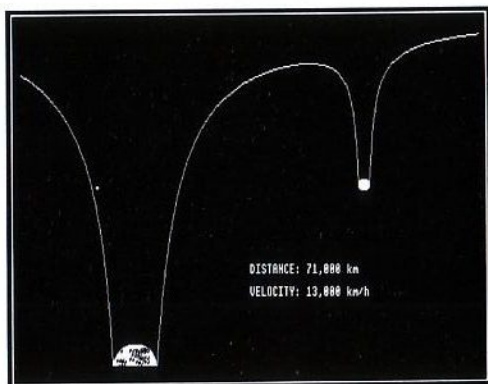
Assume the car is given only a small initial thrust. It will coast up the hill, slowing down, before running back. Increasing the initial speed will take it further. Eventually a speed will be reached such that although the car is slowing down, the slope is becoming much less steep, and it never quite comes to a stop to run back. This is escape velocity.

Of course, the hypothetical hill goes away from the planet in all directions, but by considering 2 dimensions only, the 3rd can be used to plot the energy potential to illustrate the hill itself. Imagine space to be a flat sheet with funnel pits going down to the planets and other bodies. A spaceship going around the wall of a pit at constant height will be in circular orbit around the body. If it is changing height it is in elliptical orbit.

THE PROGRAM

This simulation is the same as the previous earth-moon figure-of-eight, except that the orbits are viewed edge on, and the vertical scale shows the energy potential. Assume the spaceship is moving towards you when it is outside the two bodies, and away from you when between them. This time the track is not marked, but velocity and distance from the centre of the earth are displayed.

Watch at least one complete orbit of the moon, then when the spaceship returns to the left side of the earth, press the '<' key to retard it. With luck you should find that it stays around the planet. Slowing with the '<' key when in the lowest part of the orbit, and accelerating with the '>' key when in the highest, can produce almost a circular orbit which will appear as a horizontal line. This is a fairly easy exercise, but if you crash, press R to repeat.



Gravity pits: plotting energy potential

You may like to try the difficult tasks of returning to the figure-of-eight, or

Gravity and Orbits

orbiting the moon alone, but I think you'll have to practice on the previous conventional program first. Next time we'll survey an asteroid.

```
10 REM Program PITS
20 REM Version B2.0
30 REM Author Cliff Blake
40 REM BEEBUG May 1993
50 REM Program subject to copyright
60 :
100 ENVELOPE1,8,1,-1,1,1,1,1,121,-10,-
5,-2,120,120
110 MODE7:*FX11
120 PROCinfo:g%=GET
130 MODE0:VDU5
140 REPEAT
150 quit%=FALSE
160 CLS
170 GCOL0,1:PROCcurve:PROCplanet:PROCm
oon
180 GCOL3,1:PROCspaceship
190 REPEAT
200 rerun%=FALSE
210 PROCdistance:PROCpotential:PROCgra
vity
220 PROCthrust:PROCmove:PROCflags
230 UNTIL rerun%
240 UNTIL quit%
250 VDU4:*FX12
260 CLS:*FX21
270 END
280 :
1000 DEF PROCdistance
1010 Xdp=320-Xs:Zdp=512-Zs
1020 Xdm=960-Xs:Zdm=512-Zs
1030 Rdp=Xdp*Xdp+Zdp*Zdp:Rdp=SQR(Rdp)
1040 Rdm=Xdm*Xdm+Zdm*Zdm:Rdm=SQR(Rdm)
1050 dis%=INT(.6*Rdp)
1060 IF Rdp<60 OR Rdm<15 SOUND0,1,0,5:P
ROCCrash
1070 VDU4:PRINT TAB(40,23)"DISTANCE: ";
dis%," ,000 km";SPC3:VDU5
1080 ENDPROC
1090 :
```

```
1100 DEF PROCpotential
1110 Ep=61440/Rdp:Em=6144/Rdm
1120 Ysp=Ys
1130 Ys=1024-Ep-Em
1140 ENDPROC
1150 :
1160 DEF PROCgravity
1170 Xgp=20000*Xdp/Rdp/Rdps:Zgp=20000*X
dp/Rdp/Rdps
1180 Xgm=2000*Xdm/Rdm/Rdms:Zgm=2000*Zdm
/Rdm/Rdms
1190 Xv=Xv+Xgp+Xgm:Zv=Zv+Zgp+Zgm
1200 Rvs=Xv*Xv+Zv*Zv:Rv=SQR(Rvs)
1210 vel%=INT(.85*Rv)
1220 VDU4:PRINT TAB(40,25)"VELOCITY: ";
vel%," ,000 km/h";SPC3:VDU5
1230 ENDPROC
1240 :
1250 DEF PROCthrust
1260 Xt=0.15*Xv/Rv:Zt=0.15*Zv/Rv
1270 IF INKEY(-103) THEN Xv=Xv-Xt:Zv=Zv
-Zt:SOUND0,-7,5,5
1280 IF INKEY(-104) THEN Xv=Xv+Xt:Zv=Zv
+Zt:SOUND0,-7,5,5
1290 ENDPROC
1300 :
1310 DEF PROCmove
1320 MOVE Xs-6,Ysp+2:VDU64
1330 Xs=Xs+Xv:Zs=Zs+Zv
1340 MOVE Xs-6,Ys+2:VDU64
1350 ENDPROC
1360 :
1370 DEF PROCflags
1380 IF INKEY(-52) THEN rerun%=TRUE
1390 IF INKEY(-17) THEN rerun%=TRUE:qui
t%=TRUE
1400 ENDPROC
1410 :
1420 DEF PROCcurve
1430 Xs=0:PROCplot:MOVE Xs,Ys
1440 FOR Xs=0 TO 260 STEP 5:PROCplot:DR
AW Xs,Ys:NEXT Xs
1450 Xs=380:PROCplot:MOVE Xs,Ys
1460 FOR Xs=380 TO 945 STEP 5:PROCplot:
DRAW Xs,Ys:NEXT Xs
```

```

1470 Xs=975:PROCplot:MOVE Xs,Ys
1480 FOR Xs=975 TO 1280 STEP 5:PROCplot
:DRAW Xs,Ys:NEXT Xs
1490 ENDPROC
1500 :
1510 DEF PROCplot
1520 Xdp=ABS(320-Xs):Xdm=ABS(960-Xs)
1530 Ep=61440/Xdp:Em=6144/Xdm
1540 Ys=1016-Ep-Em
1550 ENDPROC
1560 :
1570 DEF PROCplanet
1580 Ca=COS(PI/40):Sa=SIN(PI/40)
1590 CA=1:SA=0:MOVE 320+60,0
1600 FOR A=1 TO 40
1610 Cp=CA:Sp=SA
1620 CA=Cp*Ca-Sp*Sa:SA=Sp*Ca+Cp*Sa
1630 x=60*CA+320:y=60*SA
1640 MOVE 320,0:PLOT 85,x,y
1650 NEXT A
1660 GCOL0,0
1670 FOR n%=1 TO 25
1680 MOVE 260+RND(90),RND(50):PRINT"~"
1690 NEXT n%
1700 GCOL0,1
1710 ENDPROC
1720 :
1730 DEF PROCmoon
1740 CA=1:SA=0:MOVE 960+15,512
1750 FOR A=1 TO 80
1760 Cp=CA:Sp=SA
1770 CA=Cp*Ca-Sp*Sa:SA=Sp*Ca+Cp*Sa
1780 x=15*CA+960:y=15*SA+512
1790 MOVE 960,512:PLOT 85,x,y
1800 NEXT A
1810 ENDPROC
1820 :
1830 DEF PROCspaceship
1840 VDU23,64,240,240,0,0,0,0,0
1850 Xs=302:Zs=329
1860 Xdp=320-Xs:Xdm=960-Xs
1870 Ep=61440/Xdp:Em=6144/Xdm
1880 Ys=1024-Ep-Em
1890 Xv=10.1:Zv=-7.2
1900 ENDPROC

```

```

1910 :
1920 DEF PROCinfo
1930 y$=CHR$131:c$=CHR$134:w$=CHR$135
1940 PRINT TAB(10,2)y$+"GRAVITY PITS"
1950 PRINT c$+"A spaceship is in figure
-of-eight"
1960 PRINT c$+"orbit around a planet an
d moon."
1970 PRINT w$+"It moves outwards (towar
ds you)"
1980 PRINT w$+"outside of the two bodie
s, and"
1990 PRINT w$+"inwards (away from you)
between them."
2000 PRINT c$+"The vertical scale is th
e energy"
2010 PRINT c$+"potential, and the plott
ed curves show"
2020 PRINT c$+"gravity pits."
2030 PRINT c$+"Try reducing the orbit t
o a circular"
2040 PRINT c$+"one around the planet on
ly."
2050 PRINT w$+"Gently holding down the
< key will"
2060 PRINT w$+"fire the retro unit to s
low the ship."
2070 PRINT w$+"Gently holding down the
> key will"
2080 PRINT w$+"fire the drive to accele
rate the ship."
2090 PRINT TAB(20)w$+"Press SPACE BAR."
2100 REPEAT:g%=GET:UNTIL g%=32
2110 CLS
2120 PRINT w$+"Press any key to start."
2130 PRINT w$+"Press R to clear screen
and Repeat."
2140 PRINT w$+"Press Q to Quit."
2150 ENDPROC
2160 :
2170 DEF PROCcrash
2180 REPEAT:UNTIL INKEY(-52):rerun%=TRU
E
2190 ENDPROC

```

B

Sorting (Part 2)

by David Fell

Last month we looked at ways of sorting data and ended up with a couple of useful PROCedures. As I hinted, though, there are problems when it comes to sorting strings.

The snag is the profligate way that BBC Basic allows space for each variable to hold the longest string it has ever held, regardless of its present value. This approach is appallingly wasteful compared to that of other computers. Most keep string space under control and, through a process known as "garbage collection", weed out unwanted space. But on the Beeb, we can have trouble if we try to sort a string array directly, particularly if the strings are of varying size. The strings could easily rampage through memory, ending up with the dreaded "No room" error message.

To avoid the problem, sort a set of pointers to the strings, rather than the strings themselves. Use a second array, which eventually holds, in order, the indices to the strings. For example,

suppose that the 38th string should be first; the first element in the pointer array would hold the value "38". Here is a modified Shell sort to put a string array into alphabetical order.

POINTER SORT

The routine assumes that you have already DIMmed the array *ptr%()* to have as many elements as *array\$()*. Line 10020 puts the pointers into numerical order and sorting starts, using *ptr%()* to access the strings. Note how PROCswap only exchanges pointers and does not directly affect the strings.

The pointer approach is also useful when you sort groups of related data. For instance, a list of names and addresses can be put into order without manipulating names AND addresses.

POINTER SORT

```

10000 DEF PROCstrshell(st%,fin%)
10010 LOCAL D%,F%,I%,S%,T%
10020 FOR I%=1 TO (fin%-st%)+1:ptr%(I%)
      =I%:NEXT
10030 S%=2^INT(LOG(fin%-st%)/LOG(2))
10040 REPEAT
10050   T%=fin%-S%
10060   REPEAT
10070     F%=FALSE
10080     FOR I%=st% TO T%
10090       IF array$(ptr%(I%))>array$(
          ptr%(I%+S%)) THEN PROCswap(
          F%=TRUE
10100       NEXT
10110       T%=T%-1
10120       UNTIL NOT F%
10130       S%=S% DIV 2
10140       UNTIL S%=0
10150 ENDPROC
10990 :
```

```

11000 DEF PROCswap
11010 D%=ptr%(I%)
11020 ptr%(I%)=ptr%(I%+S%)
11030 ptr%(I%+S%)=D%
11040 ENDPROC

```

So far, though, we have assumed that all the data is in memory. What if we need to sort a too-big-to-fit disc file? The answer is remarkably simple and obvious (when you know...).

Split the large file into smaller ones which WILL fit. Sort each small one and save it back to disc. Then, and this only works on disc-based systems, read the data from the small files in parallel. Select the largest (or smallest, depending) of the values at the start of each small file, and write it to the large file. Continue like this, taking the wanted value from whichever small file holds it, until they are all empty; the original large file is then sorted.

It's like splitting a pack of cards into 4 hands, sorting each hand, and then taking cards from each hand, in order, to end up with a sorted pack. It's usually also faster than trying to sort a single file.

GIANT FILE SORT

```

10000 DEF PROCfilsort(srtfile$,nitems%)
10010 LOCAL F1
10020 F1=OPENIN srtfile$
10030 PROCsort(nitems% DIV 2,"D.TEMP1")
10040 PROCsort(nitems%-(nitems% DIV 2),
"D.TEMP2")
10050 CLOSE #F1
10060 PROCmerge("D.TEMP1","D.TEMP2",srt
file$)
10080 *DELETE D.TEMP1
10090 *DELETE D.TEMP2
10100 ENDPROC
10190 :
10200 DEF PROCsort(n%,outfil$)

```

```

10210 LOCAL i%,f2
10220 FOR i%=1 TO n%:INPUT #F1,array
(i%):NEXT
10230 PROCshell(1,n%)
10240 f2=OPENOUT outfil$
10250 FOR i%=1 TO n%:PRINT #f2,array
(i%):NEXT
10260 CLOSE #f2
10270 ENDPROC
10390 :
10400 DEF PROCmerge(in1$,in2$,op$)
10410 LOCAL d1,d2,f1,f2,f3
10420 f1=OPENIN in1$:f2=OPENIN in2$:
f3=OPENOUT op$
10440 INPUT #f1,d1:INPUT #f2,d2
10450 REPEAT
10460 IF EOF #f1 THEN PROCwrapup(d1,d
2,f2):GOTO 10500
10470 IF EOF #f2 THEN PROCwrapup(d2,d
1,f1):GOTO 10500
10480 IF d1<d2 THEN PRINT #f3,d1:
INPUT #f1,d1 ELSE PRINT
#f3,d2:INPUT #f2,d2
10490 UNTIL EOF #f1 AND EOF #f2
10500 CLOSE #f1:CLOSE #f2:CLOSE #f3
10510 ENDPROC
10590 :
10600 DEF PROCwrapup(d1,d2,filno)
10610 LOCAL d1val,d2val
10620 d1val=TRUE:d2val=TRUE
10630 REPEAT
10640 IF NOT d1val AND NOT EOF #filno
THEN REPEAT:PRINT #f3,d2:INPUT
#filno,d2:UNTIL EOF#filno:PRINT
#f3,d2:d2val=FALSE:GOTO 10670
10650 IF NOT d2val THEN PRINT #f3,d1:
d1val=FALSE ELSE IF d1<=d2 AND
d1val THEN PRINT #f3,d1:
d1val=FALSE ELSE PRINT #f3,d2:
IF NOT EOF #filno THEN INPUT
#filno,d2 ELSE d2val=FALSE
10660 UNTIL NOT d1val AND NOT d2val
10670 ENDPROC

```

PROCfilsort starts with the name of the file to be sorted and the number of items

Continued on page 48

Mr Toad's Machine Code Corner

This month Mr T introduces us to FRED, JIM and SHEILA.

Greetings, fellow eggheads. The rest of you may grovel. There are thousands of readers who didn't write in to express their enthusiasm about a feature on FRED, JIM, SHEILA, HAZEL, LYNNE and ANDY, but we'll do one anyway, just to annoy you and because I've mentioned SHEILA recently.

First we'll look at the three which exist in all Beebs. FRED is the official Acorn name for the page of memory from &FC00 to &FCFF; JIM is page &FD next door. They are both normally in RAM and are reserved in the old Beeb for devices which might be plugged into the 1 MHz bus, especially extra RAM which can be paged in over JIM. On the Master they can also serve extra RAM in cartridges; again the RAM appears as pages overlaid on JIM and selected by the last address in FRED, the *paging register*. JIM itself (himself?) is reserved for short routines used by the hardware in FRED. There are a whole flock of OSBYTES to serve these pages: OSBYTES &92 to &95 inclusive, and &6B. If you're not writing code for hardware to go on the 1MHz bus or for cartridge RAM expansion, leave FRED and JIM well alone. In particular, don't ever try to use them as a handy place to stick a small assembler routine.

SHEILA is page &FE; an absolutely vital area of the Beeb. Physically she consists of various parts of all sorts of chips, page &FE being the sequence of addresses to which these hardware devices are mapped: on the old Beeb it's stuff like the 6845 video controller, the video ULA,

the system VIA, the floppy disc controller and the 74LS161 chip known as ROMSEL. So, for example if the programmer codes a write instruction to &FE20, the byte sent out on the data bus by the 6502 goes to VIDPROC, the video ULA. We made the point the other month, in connection with ROMSEL, that the 6502 neither knows nor cares whether any address it writes to is a RAM chip, a control chip, a ROM chip or an empty socket, even though in the last two cases the write has no effect - that's the programmer's problem.

On the Master, one or two more things have been added to SHEILA, such as an extra disc-interface register and ACCON (more about ACCON later.) On the old machine, each device appears across a block of several addresses - for example, ROMSEL is theoretically at &FE30, but in fact any address between &FE30 and &FE3F will work. This is because only eleven devices are connected to the address bus lines of a whole page of addresses. On the Master this is still true, but since there are more devices, there are not so many redundant 'copies'. The most important example of this for the average hobbyist-programmer like you and me is that on the Master, &FE34 is ACCON, whereas before it was a 'copy' of ROMSEL. So long as programmers for the older machines didn't try to be clever and stuck to &FE30, there are no compatibility problems. Incidentally, for tube compatibility you should write to ROMSEL via OSBYTE &97, 'write to I/O area SHEILA'.

The various books and manuals are inconsistent in their use of the names JIM, SHEILA and so on. For example, section F.6 of the Master's manual refers to FRED and JIM but avoids the name SHEILA, speaking only of 'page &FE', yet elsewhere the Acorn manuals speak of SHEILA, as do most of the 'unofficial' books. Do not, therefore, be confused by talk of 'page &FE offsets', 'SHEILA offsets' or 'SHEILA addresses' - all simply mean 'addresses within page &FE'.

As to ROMSEL, it's been amply explained in recent issues, but for newcomers it's address &FE30 - SHEILA offset &30 - to which you write in order to select one of the 16 ROM slots (including 4 RAM sections, 4 cartridge sockets and the spare socket 8, IC27). Just write the slot number to ROMSEL, e.g.:

```
LDA #7:STA &FE30:STA &F4
```

selects RAM slot 7. Do remember that ROMSEL is supposed to be a write-only device, and that every time you write to it you also immediately write to the RAM copy of it at &F4, as in the snippet above. Provided we all stick to this, we can always read the value from &F4, say Acorn, though in practice Mr T has always been able to read directly from &FE30 itself on his Master. What we've not seen before is that on the Master, setting the top bit of ROMSEL pages in ANDY, of which more next month.

A while back, we saw that a number of things like timers mapped to SHEILA can be read to provide 'random numbers' - the range from &FE40 to &FE6F is

fruitful, but I stick to my refusal to tell you the best locations, else you lose all the fun of trying different ones out.

If you can remember back to April 1992, some rogue by the name of David Holton wrote an article with two listings. The second printed 'APRIL FOOL!' but the first was genuine and showed that on the Master 128, there are three pages of ROM full of the names of people who worked on the Beeb. These pages are normally mapped out, but if you set bit 6 of ACCON and try to read from FRED, JIM or SHEILA you get the names instead. This brilliant bloke Holton made the point that you can write to ACCON to reinstate FRED and pals - even though you had just paged ACCON out, as it's in SHEILA! This is a nice example of how one area of memory can be mapped in for write access whilst a completely different chip gets the 'read' instructions. To see the names, write a program which sets up registers, etc. to copy &300 bytes from &FC00 to, say, &6000. Then put:

```
SEI:LDA &FE34:PHA:ORA #&40:STA &FE34
```

now copy out &FC00 - &FEFF to main memory. You must then immediately re-enable interrupts and restore ACCON:

```
CLI:PLA:STA &FE34
```

then print the names from main memory, use a memory-editor or whatever. Better still, get hold of BEEBÚG Vol.10 No.10.

And now, a mystery. Mr T genuinely doesn't know the answer and would dearly like an explanation from a hardware expert. It's like this: the TOAD ROM 90 which lives in slot 8 of Mr T's

Master Turbo contains a home-brewed memory-editor of the kind which continually re-reads and re-displays the half-page of addresses on screen, in order to show changes in RAM as they occur. When this software is reading any or all of locations &FE10 - &FE17 in SHEILA, there is a curious, loud, irregular ticking noise from the computer. It has to be the cassette relay going in and out. SHEILA &10 - &17 are mapped to the serial ULA, and Mr T suspects that that's the clue, but we're only reading from the device (or are we?) so what's the exact reason for the actuation of the cassette relay? Problem: don't bother to try this from Basic - it doesn't do it! That's this month's competition; there are still some badges to be awarded, so if you know, do write in.

Next time we'll look at the other bits of ACCON, also at HAZEL, LYNNE and ANDY in the new 32K RAM chip which is in the B+ and Master. There should also be space to review the new waterproof Compact with the snorkel on the chimney which permits Mr T to write all this stuff in his preferred environment.

Finally, someone asked about last month's reference to 'The Nine Billion Names Of God'. Well, it's a wonderful short story by Arthur C. Clarke. The old Sinclair Spectrum manual, page 165, states that 'In general, computers are pretty godless creatures.' Clarke's story demonstrates why this is not so! If you haven't read it, try to get hold of it; it's in several Sci-Fi collections. **B**

BEEBUG Workshop - Sorting (continued from page 45)

in it. It calls PROCsort twice, halving and sorting the file into D.TEMP1 and D.TEMP2 (ADFS users will have to create a sub-directory "D" first). These 2 files are merged, overwriting the original file, and then deleted.

PROCsort reads half the main file into *array()*, which it Shell sorts - see last month's Workshop for the code. The sorted data is then written to the temporary file. NOTE: You will have to create *array()* with: *DIM array(nitems% DIV 2 + 1)* in the main program. The '+1' allows for *nitems%* being odd.

The 2 sub-files are merged by repeatedly taking the smaller value from their tops (we're sorting into ascending order) and writing it to the original file. Eventually, we get to the end of one of the temporary files while still having data in the other.

At that point, PROCwrapup simply moves the remaining data from the non-

empty file to the main file. In doing so, it slots the last item from the empty sub-file into its correct place. Line 10650 does the job; it is a horrible compound IF statement, of which I am not particularly proud. It works, however, and saves a lot of space. The 2 variables *d1val* and *d2val* are flags which show when data from each file is used up.

Although I have only used 2 sub-files, the DFS allows 5 files to be open at any time. You could, therefore, use up to 4 sub-files. This would make PROCwrapup even more complex, however. What would you do if the original file needed more than 4 sub-files?

Next month I will take a look at how to use sorted files. In particular I'll describe a couple of ways of finding the data you want quickly.

Demonstrations of both sort procedures are included on this month's magazine disc. **B**

Dynamic Free Memory Display

by Alan Webster

It is often useful to see how much free memory is still left in your machine when typing in, developing or modifying a Basic program.

This short utility displays the amount of free memory left at any time and updates the information continually, particularly useful when developing programs that are at risk of running out of memory. The program displays the words 'FM=&' at the top left hand corner of the screen, and then displays the actual free memory in hex. It then displays a number of spaces after the amount of free memory at the top of the screen to clear a small "window" around the information. The cursor will now alternate between the current position in your program and the message at the top of the screen. On odd occasions, the routine may interfere with some VDU routines such as 'clear screen', but this is a minor problem which can be easily rectified by typing CLS or VDU12.

Type in the program and save it. Then run it and, if no errors occur, press Return. The free memory should be displayed at the top of the screen. If not then check the program carefully against the printed listing.

As soon as the program is functioning correctly, you can save the machine code by typing:

```
*SAVE FREM 900 +110 900
```

and re-run the utility at any time by typing *RUN FREM.

TECHNICAL NOTE

The code is saved at &900, which is usually unused: the area is used to store envelopes 5-16 and the cassette and RS423 buffers. This utility should not

therefore be used in conjunction with any of the above unless an alternative location, at least &110 in length, is found. Those affected are referred to the User Guide for suitable areas of memory that are likely to be unused on their machines.

PROGRAM NOTES

Most of the important lines in the program are followed by comments, but here is a brief description of each part.

Lines 1040 to 1090: Re-program the event vector to point to the start of our routine. Save the old vector for 'linking' event driven routines and set our routine to respond to the 'vertical sync' event (event number 4).

Lines 1280 to 1310: Output the free memory value in 4 digit hex.

Lines 1430 to 1480: Print out the text following the JSR. It gets the program counter and prints the text from there until it meets a NOP instruction (&EA in line 1520). It then jumps back to the NOP instruction. This is a quick way to output a piece of text.

This program was first published in BEEBUG Vol.4 No.1

```
10 REM Program FREMEM
20 REM Version B0.38
30 REM Author Alan Webster
40 REM BEEBUG May 1993
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :
100 V=&FFEE:F=&FFF4
110 D%=10:*FX13 4
120 base=&900
130 PROCassemble
140 CALLbase
150 CLS:PRINT
```

Dynamic Free Memory Display

```
160 END
170 :
1000 DEFPROCassemble
1010 FOR A=0 TO 3 STEP 3
1020 P%=base
1030 [OPT A
1040 LDA&220:STAbuf
1050 LDA&221:STAbuf+1 ; Get old Event vector
ector
1060 LDA#start MOD 256:STA &220
1070 LDA#start DIV 256:STA &221 ; Our new Event vector
1080 LDA#14:LDX#4:JSRV ; Event 4 - Vertical Sync.
1090 LDA#D%:STAbuf+7
1100 .start
1110 PHA:TYA:PHA:TXA:PHA
1120 LDA#&DA:LDX#0:LDY#255:JSRV ; VDU queue empty?
1130 TXA:BEQsplit:JMPendit ; If no then end routine
1140 .split:LDA#117:JSRV:TXA:AND#&40 ; Are cursor and edit cursor split?
1150 BEQdec:JMPendit:.dec:DECbuf+7
1160 BEQgo:JMPendit:.go:LDA#D%:STAbuf+7
1170 .cryon
1180 LDA&318:STAbuf+2
1190 LDA&319:STAbuf+3 ; Get Cursor position
1200 SEC:LDA&4:SBC&2:STAbuf+4
1210 LDA&5:SBC&3:STAbuf+5 ; Calculate free memory
1220 LDA#31:JSRV:LDA#0:JSRV:JSRV ; Put cursor at 0,0
1230 JSRtext ; Routine to print text between JSR and NOP instructions
1240 ]
1250 A$="[FM=&":$P%=A$
1260 P%=P%+LEN(A$)
1270 [OPT A:NOP
1280 LDAbuf+5:JSRshift
1290 LDAbuf+5:AND #&F:JSR disp
1300 LDAbuf+4:JSRshift
1310 LDAbuf+4:AND #&F:JSR disp ; Output free memory in Hex
1320 LDA#93:JSRV:LDA#32:JSRV:JSRV
1330 JSRV:JSRV
1340 LDA#31:JSRV:LDAbuf+2:JSRV ; Restore cursor
```

```
1350 LDAbuf+3:JSRV
1360 .endit:PLA:TAX:PLA:TAY:PLA:JMP (buf) ; Return from routine
1370 RTS
1380 .disp
1390 CLC:CMPI#10:BCC num:CLC
1400 ADC#55:JSRV:RTS ; Value is A-F
1410 .num:ADC#48:JSRV ; Value is 0-9
1420 RTS
1430 .text:PLA:STA&72 ; Prints text from PC until NOP
1440 PLA:STA&73:LDY#0
1450 .text2:INC&72:BNEtext3
1460 INC&73:.text3:LDA(&72),Y
1470 CMP#&EA:BEQtext4:JSRV
1480 JMPtext2:.text4:JMP (&72)
1490 .shift:AND#&F0:LSR A:LSR A ; Get high byte
1500 LSR A:LSR A:JSRdisp:RTS
1510 .buf:JMP0:JMP0:JMP0 ; Quick way of reserving 9 bytes!
1520 ]
1530 NEXT
1540 ENDPROC
```

B

The best Serial Link BBC micro to Archimedes

- o fully multitasking
- o works with all new Acorn machines
- o simply drag files to / from BBC

Software only £25
or with 1.5m lead £34



Prices include Postage and VAT



also Multitasking
Teletext Display £25

Ivoryash Limited,
14 Perwell Close, Bredon
Glos, GL20 7LJ 0684 73173

HINTS HINTS HINTS HINTS HINTS

and tips and tips and tips and tips and tips

Please keep sending in any tips for all BBC and Master computers. Remember, if your hint gets published, there's money in it!

DISCRETE LOCKS

by David Polak

It is possible to lock or unlock groups of files if their names all begin with the same special prefix. For example, if a group of files are saved with the prefix 'SP' ('SPtext', 'SPprog', 'SPfile' etc.), the files can be unlocked as a group by typing:

```
*ACCESS SP*
```

and locked by:

```
*ACCESS SP* L
```

It can be useful to program keys F8 and F9 to perform these actions:

```
*KEY8 *ACC. SP*|M  
*KEY9 *ACC. SP* L|M
```

Note also that having locked files with this special prefix, the rest of the files in the directory can be deleted using *Wipe or *Destroy, and that the prefixed files can be copied using *Copy. The '?' wildcard matches a single character only, as opposed to '*', used above, which matches any group of characters, or nothing. 'SP???' matches files that begin with 'SP' and are followed by three other characters.

When choosing special prefixes, try to choose rare prefixes ('!!', 'ZZ', 'QQ' etc.) that don't often naturally occur in a file name. 'SP' is a part of many file names, e.g. Sprite, Spool, Spell, Spot, Spin. It's therefore probably better to use one of the prefixes just mentioned.

BIT FIELDS IN BASIC

revisited by Nick Mellor

For those of you who didn't spot my deliberate mistake in the last issue, the line that read:

```
bits%=stop_bit%-start_bit%+1
```

can be deleted, as it duplicates part of the previous line. *FNstore_field* wasn't incorrect, but it was executing this line pointlessly.

MASTER SIDEWAYS RAM BANKS SELECTION

by Pete Booth

Gareth Leyshon in the helpful Trouble Shooting Guide (Vol.11 No.8 page 15) mentions that sideways RAM blocks in the Master can be de-selected. This is necessary in order to use the internal IC ROM sockets, for example if you want to use InterWord and SpellMaster.

Gareth recommends getting a dealer to reselect them, but it's only a simple matter of changing a link. The ones to look for are links 18 and 19, located just behind the cartridge sockets. They are not quite adjacent - link 19 is 4cm back from link 18.

Link 18 selects bank 4 and 5 as either sideways RAM or socket IC41. Link 19 selects banks 6 and 7 to be either sideways RAM or socket IC37.

The *SRDATA and *SRROM commands can be used to see the results of experimenting. Type *SRDATA n, with n = 4, 5, 6 or 7, then *ROMS to inspect.

DATE CHECKING

by David Abbot

The following function will check a date where D, M and Y hold the day, month and year. Zero is returned for an invalid date. It won't manage the non-leap years in 1900, 2100, 2200 etc.):

```
DEFNvalidDate(D,M,Y)=D*M*(M<13)*(D<31+  
(M+(M>7))AND1)+(M=2)*(2+((Y AND3)=0))
```

Shortlisted for the 'Most Unfathomable Line of Basic' award. Awarded to unconscious talent only - don't send in your own efforts!

Personal Ads

BEEBUG members may advertise unwanted computer hardware and software through personal ads (including 'wants') in BEEBUG. These are completely free of charge but please keep your ad as short as possible. Although we will try to include all ads received, we reserve the right to edit or reject any if necessary. Any ads which cannot be accommodated in one issue will be held over to the next, so please advise us if you do not wish us to do this. We will accept adverts for software, but prospective purchasers should ensure that they always receive original copies including documentation to avoid any abuse of this facility.

We also accept members' Business Ads at the rate of 40p per word (inclusive of VAT) and these will be featured separately. Please send all ads (personal and business) to MEMBERS' ADS, BEEBUG, 117 Hatfield Road, St. Albans, Herts AL1 4JS.

Master 512 in excellent working order, included are two 5.25 floppies, AMX mouse, joysticks and the following EPROMs in addition to the resident View and Viewsheet: Epson printer driver, Pascal, Screendump, Graphics, full original manuals and discs for the Master 512 plus various extra manuals, years of BBC dedicated magazines, software for both BBC & PC mode including games. Only £275 o.n.o. Will deliver north of England. Tel. (0535) 662157.

HELP WANTED: Does anyone know how I can contact P Treveethick, author of Dumpout 3 ROM? The suppliers (Watford Electronics) cannot help - if you can, please Tel. (0294) 52250 eves.

Electron Plus 1.AP4 disc interface, DFS 2.20, View cartridge, T2P4 tape/disc, Cumana disc drive DS 40/80T switch, user guide, Electron User magazines 1987-1989 inc. 1990 (Jan-July), 22 Electron User tapes £220 the lot. Tel. (0235) 834544.

CP/M 2nd processor with discs and manuals for Fileplan, Graphplan, Memoplan, Accountant, Nucleus, Basic and Cobal 30. Tel. 081-877 3398.

BBC B issue 4 with DFS £60, Taxan Kaga 9 pin printer £50, twin drive (1x40, 1x40/80) £55, floppywise ROM £8, ADT toolkit ROM £15, telexit adaptor only £25, postage extra. Tel. (0226) 762450.

BBC B issue 7, DFS, Akhter dual drives 40/80T in bridge unit, JVC colour monitor, AMX mouse, ROM and Art package, Wordwise Plus, various ROMs, manuals, discs, magazines, tapes etc. send for list - everything must go - upgraded to RISC. Tel. (0266) 880997.

BBC & Master 128 users: Free to collector, or for cost of carriage, 50 issues of A&B Computing magazine spanning September '83 to June '88. Tel. (0245) 225671 after 7pm.

Panasonic printer KX-P1180 v.g.c. complete with manual and two spare ribbons £55 o.n.o. Tel. (0366) 728917.

WANTED: Epson compatible printer also Wordwise Plus, have Sharp MZ80K as swap. Tel. 081-550 8294.

20Mb hard disc suits any Master or BBC B with ADPFS, Interbase, Masterfile II (ADPFS), Interspell, PMS Genie, Advanced User Guide. Tel. (0203) 410047.

BBC B + DFS + sideways RAM and 40T double sided disc drive, housed in teak veneered cabinet, Oak BBC B computer comprising BBC B, DFS, sideways RAM and two switched 40/80T double sided disc drives in one case and a separate keyboard, 2 Microvitec high res. monitors, one in metal case and one in plastic case, Epson FX80 dot matrix printer. Offers please. Tel. (0642) 342453 (778282 eves).

A5000 4Mb RAM, Acorn multisync, Learning Curve with PC Emulator v.18, various software, discs, magazines, books (including RISC OS Programmer's Reference manuals) etc. £1200 o.n.o. (all boxed) Tel. 081-698 3772.

LC 24 200 colour printer, Arc 440, internal hard drive, 3.5" floppy, external 5.25 floppy, monitor and software £800. Tel. (0344) 481815 after 6pm.

Z88, original packaging, manual and 2x128K ROM packs, Spellmaster and power supply, Arc link and software £150. Tel. (0727) 841126 eves.

Master 512, PC with Copro Turbo SP, Cumana combination 5.25/3.5 floppy drives, tape, full View/Overview package, full Interword/Sheet/Base/Spell package, DTP, Art, Graphics, PC software, Canon PW1080 printer, all for £450. Morley telexit adaptor £60, Pace modems £40, library of manuals/books/magazines, call for list. Tel. (0932) 865221.

WANTED: Vigen PC style case for Master 128 computer, joysticks (dual) for BBC model B computer. Tel. (0634) 341310.

BBC model B with Zif socket, View ROM, Watford 40/80 switchable disc drive, Epson LX400 printer, centronics cable, discs Welcome and Utilities, Lords of Time, Aviator, also some cassettes, user manual, Advance User Manual, Assembly Language Programming for BBC, all v.g.c £175 buyer collects please. Tel. (0733) 892985.

WANTED: BBC B+ 128k sideways RAM application, note pages and PCB settings for sideways RAM also DFS, manual, disc and ADPFS manual and disc both Acorn. Write to: Mr C Game, 24 Grosvenor Close, Tiptree, Colchester, Essex CO5 0JN.

Switched mode PSU model ext90/12 £20, Spellcheck III + ROM £13, Toolkit ROM £7, Spellcheck discs £10, Overview ROMs £50, Master Operating System (orig) £18,

BEEBUG discs £3 each, 2764 new EPROMs £2 each, 2732 new EPROMs £3 each, 6264 LP RAM £3 each, 19" rack with guides £20 6809 CPU computer complete with OS ring for details, software supplied in original packaging, EPROMs in anti-static strips. Tel. (0254) 701573.

BBC B issue 4 with DFS and single D/5 40/80T drive, Star DP-150 dot matrix printer all in perfect condition including software, games, word processors etc. £145. Tel. (0772) 865087 4.30pm onwards.

Archimedes 310, IFEL 2Mb memory board, 12Mb static RAM, PC Emulator, one extra disc drive, RISC OS upgrade, IO podule, CC ROM podule, battery, dust cover, keyboard extension lead, word processor, 1st Word Plus (rel.2), Philips green screen monitor all for £500, Epson LQ500 printer complete with single and continuous sheet feed, plus 1,000 sheets of continuous feed paper £165, four vols. Archive with binders £30, Arc Shareware discs (1-17) with backup discs £2 pair, Chocks Away compendium £18, E Type £9, Interdictor 1 £9, 40 blank discs in lockable box £18, Arc First Steps £4, Basic V manual (Acorn) £9, RISC OS PRM £35, joystick with mouse adaptor £20, dozens of computer magazines free to purchasers of equipment. Buyer collects Hastings area - Wheelchair invalid. Tel. (0424) 813794.

M512 with expanded memory, twin 5.25 double disc drive, green screen monitor, mouse, resident View and ViewSheet also EPROMs Wordwise Plus and Word Store with manuals etc. £350 o.n.o. Tel. 071-254 6072.

BBC B, 32K Acorn DFS, twin disc drive, Amex mouse, View ROM, Graphic Extension Clip, Citizen 120D printer, TV monitor plus games, software complete with joystick, manuals, only £375 o.n.o. Tel. 061-721 4328.

Archimedes A3000, RISC OS 3, 11 months old, 4Mb RAM, 20Mb hard drive, Learning Curve, DTP, Word Processing, Home Accounts, Hard Disc Companion, DeskEdit, ClipArt, Games, magazines and discs. Total cost new £2000, will sell for £600. Tel. (0795) 472939.

Various ROMs and books including, Quest Paint Plus Mouse £20, BEEBUG C £20, Viewsheet/store, Dabhand Guide £5 plus lots more. Tel. (0782) 415963 after 7pm.

WANTED: Electron games (cassette). Tel. (0703) 644709.



WEATHER STATION METRICATION

With reference to my program BAROM (BEEBUG Vol.11 No.2) and the subsequent update (Vol.11 No.6 and Vol.11 No.9), some people may have a barometer which is calibrated in millibars rather than in inches of mercury, so I enclose a fix for this purpose.

In the original program BAROM, add lines:

```
1545 Y$=STR$(Y(1)):IF MID$(Y$,3,1)=". "
THEN M=1 ELSE M=33.86
```

Add '/M' in three places, viz:

```
1550 ...MOVE...80+(Y(1)/M-28.4)...
1610 ...PLOT...80+((Y(N)/M-28.4)...
1710 ...totalpress=totalpress+(Y(N)/M)
:...
```

In the later version of the program, BARTEM2 (Vol.11 No.9), amend line 1450:

```
1450 ...READ X,Y,R:Y$=STR$(Y(1)):IF
MID$(Y$,3,1)=". " THEN M=1 ELSE M=33.86
```

Add line 1455:

```
1455 totalpress=totalpress+Y/M:totalrai
n= totalrain+R
```

Amend line 1480:

```
1480 ...READ X,Y,R:Y$=STR$(Y(1)):IF
MID$(Y$,3,1)=". " THEN M=1 ELSE M=33.86
```

And add:

```
1485 MOVE160,80+((Y/M-28.4)*(5*hght%)):
GCOL0,1:FOR N=2 TO end:READ X,Y,R:IF
MID$(Y$,3,1)=". " THEN M=1 ELSE M=33.86
```

And finally amend line 1490:

```
1490 PLOT5,160+((X-1)*wdth%),80+((Y/M-
28.4)*(5*hght%)):NEXT:VDU4
```

However, note that the amended program is now too large to fit into mode 1 on a model B, unless the value of PAGE is reduced from &1900 to &1200 before loading and running the program.

Nick Case

We are including a copy of the BARTEM2 program updated as described above on this month's magazine disc.

JOBLOG UPDATE

BEEBUG reader David Polak wrote to us with a problem he was having with JobLog (BEEBUG Vol.11 Nos.7&8). JobLog author, Jeff Gorman, writes:

Thank you for sending me the copy of Mr. Polak's letter. The error to which he refers occurs after sorting or deleting from an extended file. The Principal Index is overwritten with the contents of the current file instead of the altered file being rewritten.

The remedy is to change line 5200 in JobLogTwo (from part 2) to the following:

```
5200 NEXT:rcdNo%=max%:PROCWriteData(fnm
e$(level%)):PROCmenu:ENDPROC
```

thereby effectively changing *fn\$* to *fname\$ - fn\$* actually is "Index", so causing the difficulty. I have removed its declaration from the initialisation procedure (line 1270).

I notice, too, that there is a REM left behind on line 4610. It does no harm, of course, but may be worth mentioning.

With head hung in shame, apologies for the error. I thought I had checked all possibilities!

Jeff Gorman

BEEBUG MEMBERSHIP

Send applications for membership renewals, membership queries and orders for back issues to the address below. All membership fees, including overseas, should be in pounds sterling drawn (or cheques) on a UK bank. Members may also subscribe to RISC User at a special reduced rate.

Renewal rates for subscriptions to BEEBUG magazine and magazine disc shown on these pages are for 12 months commencing with May 1993 only. For renewals after this date please refer to the rates shown on page 17 (see this month's editorial).

BACK ISSUE PRICES

All overseas items are sent airmail. We will accept official UK orders for subscriptions and back issues, but please note that there will be a £1 handling charge for orders under £10 which require an invoice. There is no VAT on magazines.

BEEBUG SUBSCRIPTION RATES

£18.40
£27.50
£33.50
£36.50
£39.50

1 year UK, BFPO, Ch.1
Rest of Europe & Eire
Middle East
Americas & Africa
Elsewhere

BEEBUG & RISC USER

£28.90
£42.90
£53.10
£58.40
£62.50

Volume	Magazine	5" Disc	3.5" Disc
6	£1.00	£3.00	£3.00
7	£1.10	£3.50	£3.50
8	£1.30	£4.00	£4.00
9	£1.60	£4.00	£4.00
10	£1.60	£4.75	£4.75
11	£1.90	£4.75	£4.75

POST AND PACKING

Please add the cost of p&p when ordering. When ordering several items use the highest price code, plus half the price of each subsequent code.

Stock Code	UK, BFPO Ch.1	Europe, Eire	Americas, Africa, Mid East	Elsewhere
a	£1.00	£1.60	£2.40	£2.60
b	£2.00	£3.00	£5.00	£5.50

BEEBUG

117 Hatfield Road, St. Albans, Herts AL1 4JS
Tel. St. Albans (0727) 840303, FAX: (0727) 860263
Manned Mon-Fri 9am-5pm (for orders only 9am-6pm and 9am-5pm Saturdays)
(24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by RISC Developments Ltd.

Editor: Mike Williams
Assistant Editor: Kristina Lucas
Technical Assistant: Nick Mellor
Editorial Assistance: Marshal Anderson
Production Assistant: Sheila Stoneman
Advertising: Sarah Shrive
Subscriptions: Helen O'Sullivan
Managing Editor: Sheridan Williams

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, RISC Developments Limited.

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £50 per page, but please give us warning of anything substantial that you intend to write. A leaflet 'Notes to Contributors' is available on receipt of an A5 (or larger) SAE.

Please submit your contributions on disc or cassette in machine readable form, using "View", "Wordwise" or other means, but please ensure an adequate written description is also included. If you use cassette, please include a backup copy at 300 baud.

In all communication, please quote your membership number.

RISC Developments Ltd (c) 1993

Printed by Arion Printers (0923) 288328 ISSN - 0263 - 7561

Magazine Disc

May 1993

SLIDE CATALOGUER - the first part of a new program to assist in cataloguing a slide collection (or similar) by computer. This application is designed to take advantage of the ADFS file structure, but could be adapted for DFS without much difficulty.

ROBOL - this month's disc contains an additional program for designing new Robol screens, or editing existing ones. We have also included on this disc the complete 50 screen version of *Robol: The Game*.

3D TEXT - this program is a demonstration of the techniques described in the magazine for displaying text in 3D format on the screen for titles etc. Use these ideas to brighten up your title screens.

GRAVITY & ORBITS - another fascinating program to challenge your skill in mastering the complexities of satellites in orbit. This month you can plumb the depths of *Gravity Pits* - will you manage to escape?

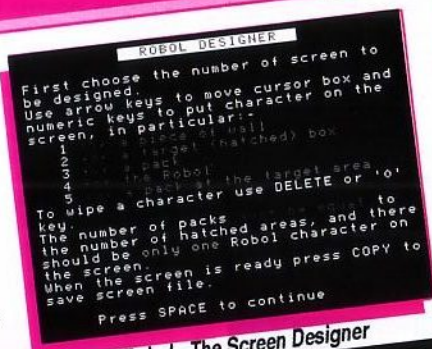
BEEBUG WORKSHOP - two separate programs, one for sorting strings by using pointers for efficiency, and one for sorting files on disc, as demonstrations of more sorting techniques.

FREE MEMORY DISPLAY - a useful utility for programmers which shows a continually updated display on screen of the amount of free memory as a program is developed - ideal for programs which use a lot of precious memory.

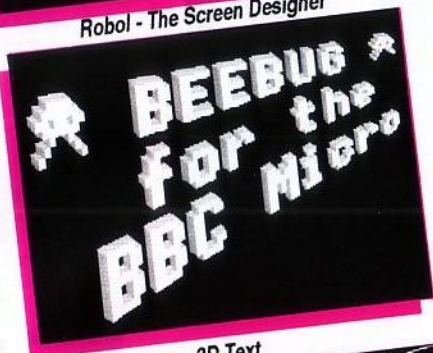
MAGSCAN DATA - Bibliography for this issue of BEEBUG (Vol.12 No.1).

BONUS ITEM

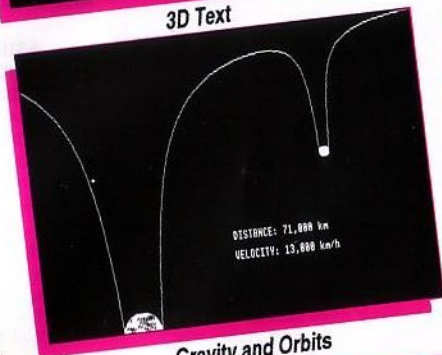
WEATHER STATION - an updated version of the program BARTEM2 from Vol.11 No.9 which allows pressure readings to be entered in millibars rather than in inches of mercury.



Robol - The Screen Designer



3D Text



Gravity and Orbits

ALL THIS FOR £4.75 (5.25" & 3.5" DISC) + £1 P&P (50p FOR EACH ADDITIONAL ITEM)
Back issues (5.25" and 3.5" discs from Vol.6 No.1) available at the same prices.

DISC (5.25" or 3.5") SUBSCRIPTION RATES

6 months (5 issues)
12 months (10 issues)

UK ONLY
£25.50
£50.00

OVERSEAS
£30.00
£56.00

Prices are inclusive of VAT and postage as applicable. Sterling only please.
See page 17 for renewal rates applicable after May 1993.

RISC Developments, 117 Hatfield Road, St. Albans, Herts AL1 4JS

Upgrading to an Archimedes

We know that many BEEBUG readers have already upgraded to an Archimedes, and no doubt many more will choose to follow a similar route. For their benefit we offer our advice to help them make a sensible decision on whether to upgrade and if so, what path to take.

Any prices quoted relate to our associated company Beebug Ltd., but note that all prices, particularly those on trade-ins and secondhand items, are likely to change without notice. You should always telephone or write for the latest information.



Archimedes A5000

What System to Choose

All new Archimedes systems are now supplied with the RISC OS 3.10 operating system. Any secondhand system should be upgraded to this. Based on the experience of existing users, we would strongly recommend a minimum of 2Mb of RAM. Most users find a hard disc adds significantly to the convenience of using an Archimedes, but you can always add a low-cost hard drive later, and more memory, but check on the likely price of future expansions - it is not necessarily the same for all machines. If you might be interested in more specialised add-ons (scanners, digitisers, etc.) then check the expansion capability of your preferred system.

Compatibility and Transferability

You will need to decide to what extent you wish to continue using existing discs and disc drives on an Archimedes. An Archimedes and a BBC micro can be directly connected for transfer of files. You can also connect a 5.25" drive to an Archimedes via an additional interface to continue to access 5.25" discs (ADFS format).

Our DFS Reader will also allow files to be transferred to the Arc from DFS format discs. However, none of this is possible with the latest A3010/A3020/A4000 systems.

Much BBC micro software will run directly on an Archimedes, or via the 6502 emulator. However, consider this carefully; in our experience, despite prior misgivings, most Archimedes users find that they rapidly adjust to the Desktop environment of the Archimedes, and quickly abandon the software and data of their old system after an initial period.

Software for the Archimedes

The Archimedes is supplied complete with a range of basic applications software. Before embarking on any further purchases it may be better to familiarise yourself with the new machine. Most users look for a word processor (or DTP package), maybe a spreadsheet, or a database, plus other more specialist software. We cannot give detailed guidance here, but back issues of RISC User contain a wealth of useful information - we can advise on suitable issues.

the outset. Note: the price on some systems includes a monitor; in other cases a choice of monitor is available at an additional cost. The details given in the table are minimum specifications of the different Archimedes models.



The A3010

It may also be possible to trade in an existing monitor and/or disc drive, but check if your existing monitor is suitable for use with an Archimedes first. You may find it better to advertise your BBC system in BEEBUG and sell privately - this applies particularly to any software and hardware add-ons which cannot be

Archimedes Systems - Typical or Current Prices

	Secondhand	New
+ A310 1Mb RAM	£350	
+ A410/1 1Mb RAM	£565	
+ A420/1 2Mb RAM, 20Mb hard drive	£650	
+ A440/1 4Mb RAM, 40Mb hard drive	£725	
+* A3000 1Mb RAM	£350	
* A3010 1Mb RAM, Family Solution		£ 499.00
* A3020 2Mb RAM, 60Mb hard drive		£1056.33
* A4000 2Mb RAM, 80Mb hard drive		£1115.08
* A5000 2Mb RAM, 80Mb hard drive		£1643.83
+* Acorn standard colour monitor	£145	£ 258.50

All systems above include a single floppy disc drive.
New (*) and secondhand (+) - all prices inc. VAT.
The A5000 price includes a multiscan colour monitor,
A3020/A4000 price includes standard colour monitor.

BBC Micros - Typical Trade-in Prices

Model B (Issue 7)	£ 35
Model B (issue 7) + DFS	£ 75
Master 128	£125
Master Compact	£ 50

General Advice

It is advisable to discuss your requirements with the BEEBUG technical team before making a final decision on what you want. Try to anticipate future expansion needs at

accepted for a trade-in. In future, all personal ads for Archimedes systems in RISC User will also be included in BEEBUG. You may also defer a trade-in until a later date provided you make this clear at the time of purchase.

For further information on all Archimedes systems contact:

BEEBUG Ltd. 117 Hatfield Road, St Albans, Herts AL1 4JS. Tel. 0727 840303 Fax 0727 860263