PRICE £1.30

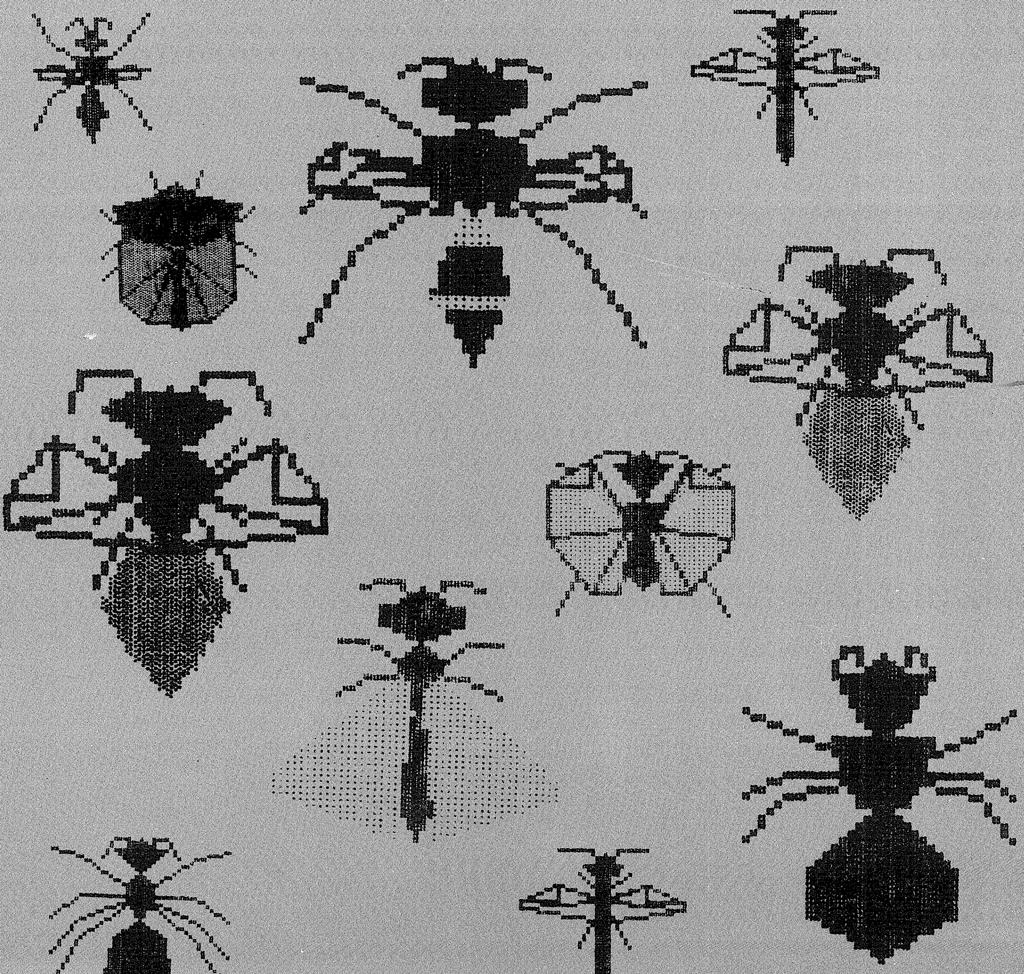
MAY 1987

BEEBUG

VOLUME 6 ISSUE 1

UNIVERSAL DISC FRONT END

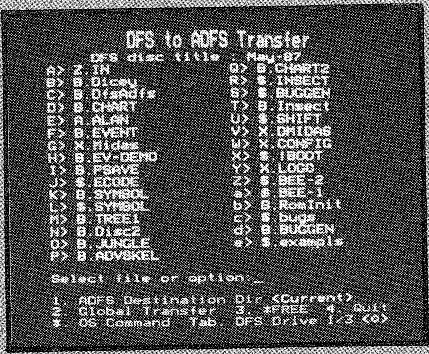
FOR THE BBC MICRO



SELECTIVE BREEDING

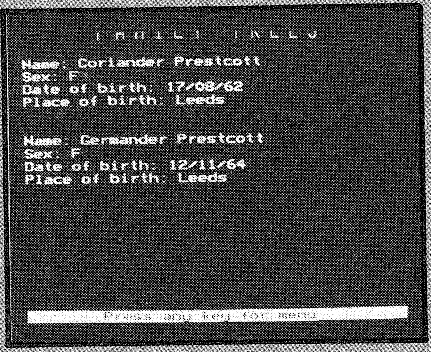
BEEBUG

Volume 6 Number 1
May 1987

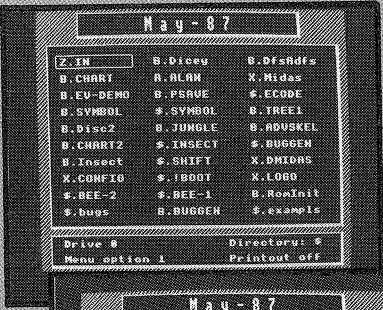


```
DFS to ADFS Transfer
   DFS disc title : May-87
A> Z.IN                 Q> B.CHART2
B> B.Dicey              R> $.INSECT
C> B.DfsAdfs            S> $.BUGGEN
D> B.CHART              T> B.Insect
E> A.ALAN               U> $.SHIFT
F> B.EVENT              V> X.DMIDAS
G> X.Midas              W> X.CONFIG
H> B.EV-DEMO            X> $.!BOOT
I> B.PSAVE              Y> X.LOGO
J> $.ECODE              Z> $.BEE-2
K> B.SYMBOL             a> $.BEE-1
L> $.SYMBOL             b> B.RomInit
M> B.TREE1              c> $.bugs
N> B.Disc2              d> B.BUGGEN
O> B.JUNGLE             e> $.exampls
P> B.ADVSKEL

Select file or option:_

1. ADFS Destination Dir <Current>
2. Global Transfer  3. *FREE  4. Quit
*. OS Command  Tab. DFS Drive 1/3 <0>
```

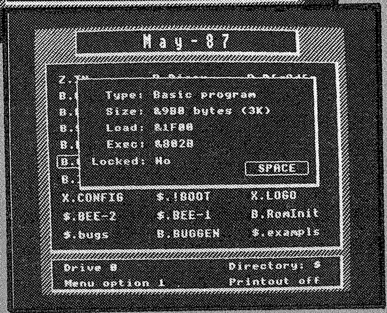
DFS to ADFS Transfers



```
         F A M I L Y   T R E E

Name: Coriander Prestcott
Sex: F
Date of birth: 17/08/62
Place of birth: Leeds

Name: Germander Prestcott
Sex: F
Date of birth: 12/11/64
Place of birth: Leeds


        Press any key for menu
```



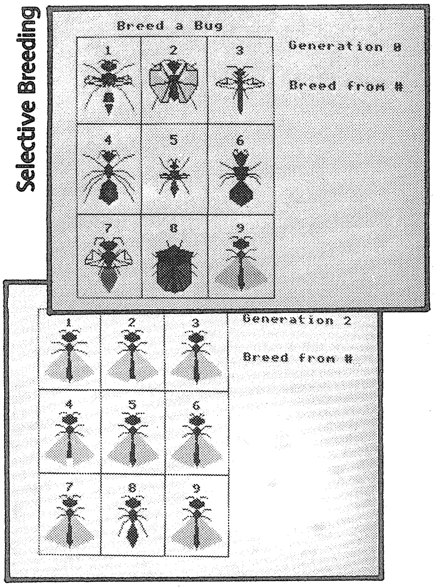
```
              May-87

Z.IN          B.Dicey       B.DfsAdfs
B.CHART       A.ALAN        X.Midas
B.EV-DEMO     B.PSAVE       $.ECODE
B.SYMBOL      $.SYMBOL      B.TREE1
B.Disc2       B.JUNGLE      B.ADVSKEL
B.CHART2      $.INSECT      $.BUGGEN
B.Insect      $.SHIFT       X.DMIDAS
X.CONFIG      $.!BOOT       X.LOGO
$.BEE-2       $.BEE-1       B.RomInit
$.bugs        B.BUGGEN      $.exampls

Drive 0                  Directory: $
Menu option 1            Printout off
```

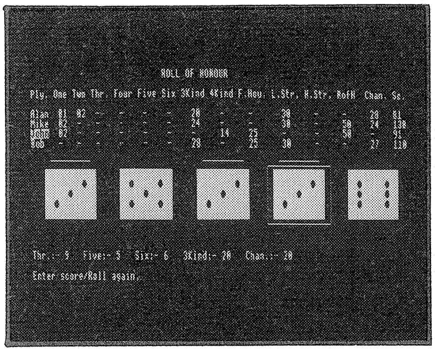


```
              May-87

Z.
B.   Type: Basic program
B.   Size: &9B8 bytes (3K)
B.   Load: &1F00
B.   Exec: &8028
B.  Locked: No              SPACE
B.
X.CONFIG      $.!BOOT       X.LOGO
$.BEE-2       $.BEE-1       B.RomInit
$.bugs        B.BUGGEN      $.exampls

Drive 0                  Directory: $
Menu option 1            Printout off
```

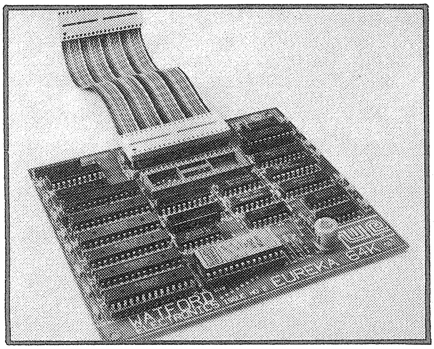
MIDAS Disc Front End

Selective Breeding




```
                         ROLL OF HONOUR
Ply. One Two Thr. Four Five Six 3Kind 4Kind F.Hou. L.Str. H.Str. RofH  Chan. Sc.
Alan  81  82  -    -    -    -   28    -     -     30     -      -     20    81
Mike  82  -   -    -    -    -   24    -     -     30     -      50    24   138
[illegible] 82 -   -    -    -   -     14    25    -      -      50    -     91
Bob   -   -   -    -    -    -   28    -     25    30     -      -     27   110

Thr.:- 9  Fives:- 5  Six:- 6  3Kind:- 28  Chan.:- 28
Enter score/Roll again.
```

Roll of Honour



Watford's Eureka Board

EDITORIAL JOTTINGS

FIVE YEARS OLD

With this issue of BEEBUG and the start of volume 6, we can look back on five complete years of publication. Many BEEBUG members have been with us from those early days, and we certainly appreciate the help and support that we have received over the years.

DARWINIAN EVOLUTION

One of the main features in this issue is an excellent program which illustrates the principles of evolution and selective breeding. We are aware that this idea has received recent coverage in another magazine, but we feel that the high quality of our program fully justifies publication. The results are well illustrated in the magazine, and we believe these speak for themselves.

BACK ISSUES

Please note that we are now beginning to run out of certain issues of volume 1 of BEEBUG. The latest position is shown in the back issue details inside the back cover, and on Micronet (*80090980#). We have no immediate plans to reprint any of the issues of volume 1 once our stocks are exhausted.

MEMBERSHIP NUMBERS

Please be prepared to quote your BEEBUG membership number whenever you need to contact us (orders, technical and magazine queries, subscription renewals etc). Where services are for the benefit of members only, we do want to make sure, in everyone's interest, that only members can take advantage. Membership cards, showing each person's number, are being sent out to all new members as they join, and to existing members on renewal. Some 50% of members have now received their card. It will also help us to respond to written enquiries promptly if you can indicate clearly the nature of your communication (technical query, order, subscription renewal, etc) on the front of the envelope.

We would also like to point out that membership of BEEBUG entitles you to a full 12 months in which to enjoy the advantages of membership. The expiry date on your membership card just indicates the last issue of the magazine which you will receive with your current subscription.

PROGRAM/REVIEW CLASSIFICATION

We hope that the new classification symbols for programs and reviews clarify matters with regard to the variety of Acorn systems. The complete set of icons is shown below. These show clearly the valid combinations of machine (version of Basic) and filing system for each item, and Tube compatibility. A single line through a symbol indicates partial working (normally just a few changes will be needed); a cross shows total incompatibility. Reviews do not distinguish between Basic I and II.

Computer System		**Filing System**	
Master (Basic IV)	M	ADFS	A
Compact (Basic IV)	C4	DFS	D
Compact (Basic VI)	C	Cassette	▭
Model B (Basic II)	II	**Tube Compatibility**	
Model B (Basic I)	I	Tube	⊖
Electron	E		

News News News News News News Nev

WYSIWYG for Wordwise

As the name implies, the new WYSIWYG PLUS ROM from Technomatic enables Wordwise Plus users to see on the screen exactly what will be printed out. WYSIWYG PLUS costs £22.85 inc VAT and p & p direct from Technomatic who are on 01-208 1177.

Specially for Kids

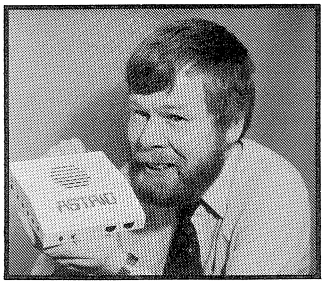
KIDS is a stand-alone videotex system for the BBC micro developed by Starter Software Ltd. Up to 2000 frames can be stored, including animated frames, to provide a comprehensive information system. This version, developed from one aimed at the tourist industry, is specifically aimed at schools and other educational establishments. The KIDS software on ROM with a disc of utilities costs £29.95, and other optional extras include an on-screen editor for £11.95 (VAT extra in both cases). For further information telephone 0874-730692.

Comprehensive School Admin.

School administration is certainly an area where computers can provide considerable help, and the School Administration System may be one answer. This comprehensive and integrated package covers everything a school is likely to need; teacher lists, pupil lists, set lists, form lists, options, exam entry and analysis, curriculum analysis, timetabling, staff absence cover, PTA meetings, sports leagues etc. The system is in two versions, one which will cope with up to 74 teachers and 240 pupils per year for use on a model B, and one handling up to 120 teachers and 400 pupils per year group on the Master Turbo. The package, with a very substantial manual, costs £120 from A.J.Dean, 6 Birkland Drive, Edwinstowe, Notts NG21 9LU.

ASTRID Receptive as Ever

ASTRID, the satellite receiver and decoder for the scientific satellites UoSAT1 and UoSAT2 is now being marketed directly by its designer, Steve Webb. ASTRID still costs £149 and further details are available on (0653) 697513.



Three Piece Suite

Minerva, producer of the System Delta database package (reviewed BEEBUG Vol.5 No.4) has announced three additional support programs. Mailshot is a versatile utility designed to link with word processors to produce mail-merged letters, and to provide a label printing facility. Reporter provides a comprehensive report generator claimed to be more powerful than dBASE III with Report Writer. Finally Inter/View Link will link Minerva's Card Index database with the Computer Concepts' Inter range, and Acorn's View family. The three new packages cost just £19.95 each and the original System Delta including Card Index costs £64.95. Minerva are on (0392) 37756.

New Acorn Products

Acorn has announced five 'new' products. Master OverView provides the rest of the View family for Master owners, who have View and ViewSheet bundled in with the Master, at a cost of £98.90 (ROM cartridge plus disc). For model B and B+ users, the Educational Compendium at £14.95 provides Spooky Manor, ABC and Workshop in a single pack. Both disc and cassette are included. More interestingly, Acorn has announced a 'Universal Second Processor'. Packaged in the standard Acorn add-on box (used for second processors, teletext adaptor etc) this allows model B and B+ users to fit the Master series Turbo and 512 co-processors to their machines. The Universal Second Processor is priced at £86.25 including VAT.

Acorn has also announced a Filestore unit for Econet systems and the availability of Comal, Lisp, Iso-Pascal and Micro-Prolog for the Master 128 and Compact. Full details from Acorn on (0223) 214411.

BEEBUG SOFT FORUM

This month's Forum is devoted entirely to Command, our very successful communications ROM. Due to its high degree of flexibility, Command can readily be programmed for many interesting applications. We would like to hear from people who have written programs using Command, or have any hints or tips that may be useful to other users.

Extended Teletext Editor

Command's teletext editor, VEDIT, does not have built-in options to load and save screens, although these functions are available at the press of a key from within the Viewdata Terminal mode. The short program below, using Command's facilities, allows teletext screens to be easily edited, loaded, saved, and dumped to your printer. It uses seven commands from the ROM, and can quite easily be extended to produce a very powerful editor to suit your own individual requirements.

```
   10 REM Teletext Editor
   20 MODE 7
   30 HIMEM=&7000
   40 *BCLEAR &7000
   50 REPEAT
   60 VDU23,1,0;0;0;0;
   70 REM Put screen in buf
fer
   80 *PSCREEN
   90 *PCLOSE
  100 REM Print menu
  110 MODE 7:PRINT"Teletext
 Editor"
  120 PRINT'"E Edit Screen"
  130 PRINT"L Load Screen"
  140 PRINT"S Save Screen"
  150 PRINT"D Dump Screen"
  160 PRINT"Q Quit Editor"'
  170 REPEAT
  180 K%=INSTR("LSEDQlsedq"
 GET$)
  190 UNTIL K%>0
  200 IF K%>5 K%=K%-5
  210 IF K%<3 INPUT'"Filena
me? :"F$
  220 VDU23,1,0;0;0;0;
  230 REM Load Screen
  240 IF K%=1 OSCLI("GSCREE
N "+F$)
  250 REM Get screen from b
uffer
  260 IF K%>1 THEN *GSCREEN
  270 *GCLOSE
  280 REM Edit screen
  290 IF K%=3 OR K%=1 VDU30
:*VEDIT
  300 REM Save Screen
  310 IF K%=2 OSCLI("PSCREE
N "+F$):*PCLOSE
  320 REM Screen Dump
  330 IF K%=4 THEN *SDUMP
  340 UNTIL K%=5:CLS:END
```

Hayes Compatible Modems

Although Command will not auto-dial as standard on Hayes type modems, it should be possible to dial by sending characters to the modem with *SAY. We would be very interested to hear from anyone who has successfully done this.

Accessing Telecom Gold

Some users are experiencing problems accessing Telecom Gold and similar text services. Usually, this is because the software standard has not been set to the correct value of 2. This should be done from within Text Terminal mode using *STANDARD2, or alternatively the value may be edited on the status screen. The default software standard in the Text Terminal is 5, which is suitable for most bulletin boards, and especially those which use colour.

Sending Sign-on Codes

A good tip when entering sign-on codes in the Telephone Directory, is to precede the string with a short delay. This prevents the string being sent before the host computer is ready to receive it. The symbol % in the string followed by a number will introduce a delay of the corresponding number of seconds. For example:

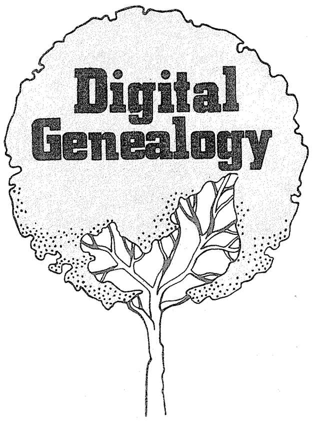
```
%5|4444444444
```

Notice that after %5 (indicating a 5 second delay), an extra bar symbol must be inserted before the sign-on string to prevent confusion between the two sets of numbers. Further details of this can be found in the manual under the command *SAY.

Auto-Answering

The auto-answer command *ANSWER works well on most telephone systems, but on some internal systems it may not appear to function correctly. This is because the time delays between the ringing tones can vary, and hence the call is not recognised by Command. This is quite easily solved by typing *RINGS 1 before using *ANSWER, forcing the modem to answer after the first ring.

Digital Genealogy



Product : Bel Gen
Supplier: Bel Tech
College House,
St Leonards Cl., Bridgenorth,
Shropshire WV16 4EW.
Tel: (07462) 5420
Price : £23.00 inc. VAT and p & p.

Product : Family History System
Supplier: Micro-Aid
25 Fore Street, Praze,
Camborne, Cornwall TR14 0JX.
Tel: (0209) 831274
Price : £23.00 inc. VAT and p & p.

There is a lot of interest these days in compiling a personal family tree. Paul Hendy, who has delved into his own family history, compares two packages for the BBC micro that might help you in your task.

One of the more unusual applications for which a computer is well-suited is the whole subject of family histories. In theory it needs just a dedicated database system, but in reality there are many problems to be overcome, not the least of which is to ensure that somebody doesn't apparently die before they were born, and manage to give birth to children before reaching the age of five!

Genealogy is a task which grows increasingly more complex as you delve further back into the past. The links between obscure members of the family grow tenuous, as records become increasingly difficult to trace. A computer is ideally suited to holding and comparing the various sets of data, enabling the entire family structure to be understood more easily.

Several traps have to be included in any program, like the obvious example above, to prevent links being made which are impossible, like brother and sister marrying, and software houses have a difficult task on their hands to try and provide all the facilities that are required.

BEL GEN

The best-known genealogy program is Bel Gen, produced by Bel Tech of Bridgnorth. It is a disc-based package, and has the ability to work with up to 720 records (assuming 80 track double-sided drives). This may seem a lot, but in reality the limit can soon be reached, mainly because families in the last century were so large. Also, Bel Gen requires a top limit to be set before any data is entered. This is absurd, and I would advise you always to select the maximum, because it appears impossible to increase it once set, and no genealogist ever knows in advance how many entries are going to be required.

There are 10 options in all, selected from a menu by function keys, (the last of which displays a secondary menu of report options). These deal with the creation, deletion, and editing of records, each consisting of 20 fields, covering surname, spouse, date-of-birth, occupation, etc, and a notes field of only 20 characters. One of the more fascinating aspects of genealogy is the discovery of interesting items of information about an ancestor, and to have only 20 characters in which to record these is hopelessly inadequate. You need a backup card index system - which defeats the whole object of the program!

An individual record can be displayed, with the option of a hardcopy printout which is extremely slow. It prints a character at a time, with the cursor travelling slowly across the screen. Curiously some middle names were omitted from some printouts. Other aspects of the program are also slow; for example, records are always written to and read from disc directly using random access, which results in a relatively slow response.

The secondary menu of reports options allows different formats for printouts, including an 'own report', whereby you can set up your own fields to a maximum length of one line (normally 8Ø characters for printers and 4Ø characters for screen). It is possible, for example, to list those who are alive during a specific period, or trace the male/female line. Genealogists will find this useful so that links between family members can be tested out to see if they are physically possible. The output can be EITHER hardcopy OR screen, not both, as is possible with 'find and display'.

When somebody imagines a family tree, the normal picture is of a complicated structure with interconnecting lines, and hopefully a few heraldic coats of arms thrown in. Don't buy Bel Gen if you want this. It will print out a line of male or female ascendancy - for example, it will print out the great-grandfather, grandfather, father, and son all linked by a line, but it won't produce the traditional family tree. Selecting the option to print the entire file will generate a series of individual records, not an integrated structure.

Bel Gen has some serious failings. There is negligible error-trapping, so you can die before you are born, and in the process marry your parent and give birth to yourself. As an index system it is good; for holding individual records, and tracing links between them it works well; but as a dedicated family tree system I did not find it satisfactory.

FAMILY HISTORY SYSTEM

This package from Micro-Aid contains a suite of programs for handling family tree data. It defaults to a record capacity less than Bel Gen, but the variables can be altered to store more records, provided the data entered is shorter. In operation, the records are held in the computer's memory. Thus operations are faster, but the database needs to be saved periodically to safeguard against possible corruption or loss.

The opening screen displays the various options available, the two main ones being Easy Tree and Print Tree. Entering Easy Tree reveals another screen of menu options. Data can be entered either as individual people, or as family groups. This works well, with the father, then spouse, followed by children, being entered together. As the data is entered, so the links between individuals are made automatically. Once one group has been entered, another may be started. If any person entered relates to somebody keyed in previously, their record number is inserted, and the link established. People can also be entered as individuals with no known links, and then modified later.

Editing data is a little cumbersome. For example, to append another child first requires you to work through the entire family group to get to the required place.

The date can be entered in various formats, but approximate dates (so often the case) can't be included. The manual states that 13, 14, and 15 can be used for January, February, March pre-1752. This is when the year was altered to run from January to December - nice to see that it has been catered for. Contradictory dates can't be entered (in contrast to Bel Gen), and the comments file will hold up to 24Ø characters, a much more realistic size. However there are less fields than with Bel Gen, thus limiting how the data can be sorted - indeed sorting is probably the weakest feature of this package.

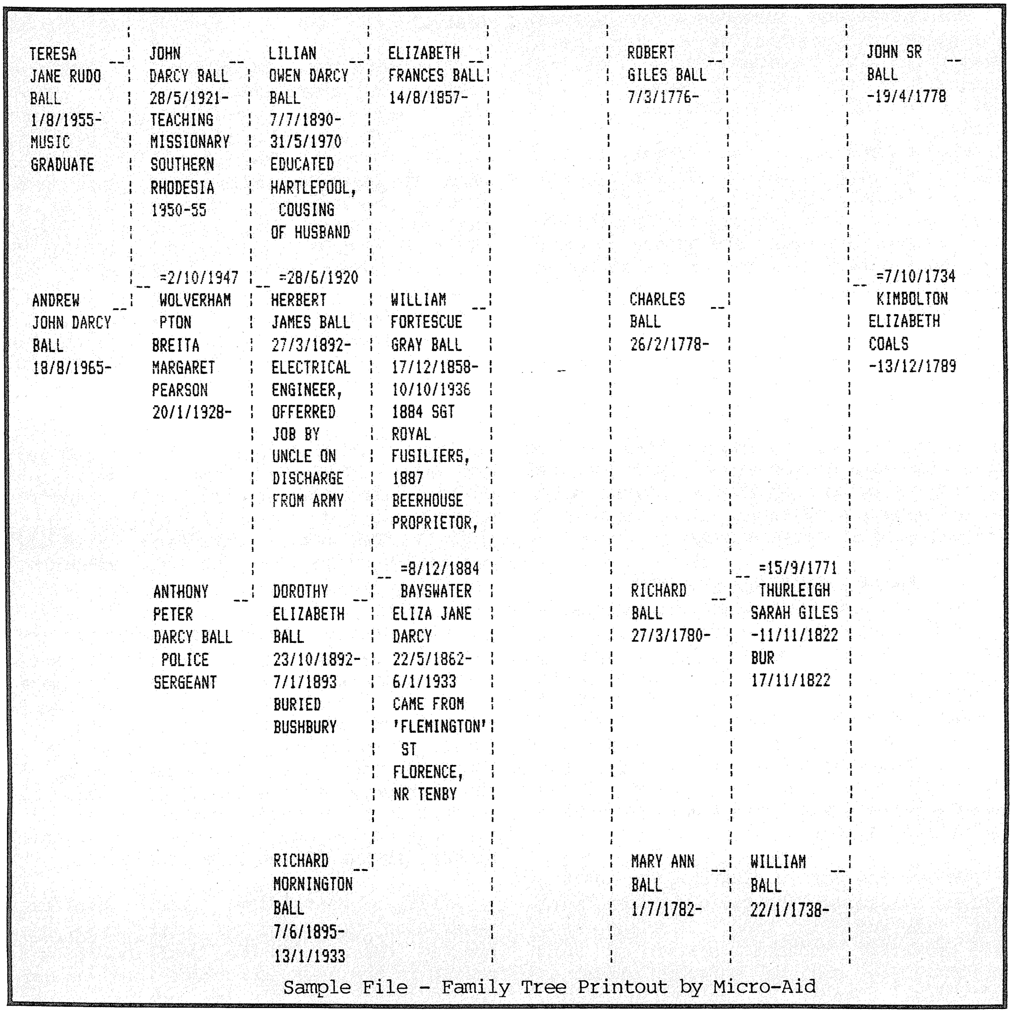
The Browse option is great - the record on display is treated as current and you can move to father, mother, spouse or child, with each record then becoming current, so the process can continue right through the entire structure. This was remarkably effective for just wandering round families.

There are options to show ancestors and descendants, on screen or printed out, with indents for each generation to make the result easier to follow. The latest version of the program now has the facility for setting printer codes from the menu, and this makes the control of any hardcopy much easier. Lists may be generated in record number order, or alphabetically, but the real joy of this package is the 'Print Tree' option.

After specifying the earliest generation you want to work from, the program calculates the position of each record in the tree structure and you can then print it out. You are advised to use condensed print and specify a width of 132. The final result is fantastic because the traditional tree picture is produced, a very rewarding sight after entering so much data. In the sample file, cousins were married in an attempt to fool the program, but it coped admirably, displaying the marriage correctly, and then showing the one cousin in his own right, with the comment 'already shown'.

Two other programs on the disc will convert the data into an international format known as the International Genealogical Index (IGI) devised by the Mormons at their massive database complex in Salt Lake City.

I enjoyed using this system, and felt well-rewarded when I achieved my printouts. The report facilities are not as comprehensive as Bel Gen, but it is more friendly and easier to understand. The present manual is poor and needs updating to match the software, but for anybody who wants to have a chart on the wall and be able to trace their family history lines, I can thoroughly recommend this program.



```
             |                           |             |             |             |             |
TERESA     __| JOHN        __ LILIAN    __| ELIZABETH __|             | ROBERT    __|             | JOHN SR    __
JANE RUDO    | DARCY BALL  | OWEN DARCY  | FRANCES BALL|             | GILES BALL  |             | BALL
BALL         | 28/5/1921-  | BALL        | 14/8/1857-  |             | 7/3/1776-   |             | -19/4/1778
1/8/1955-    | TEACHING    | 7/7/1890-   |             |             |             |             |
MUSIC        | MISSIONARY  | 31/5/1970   |             |             |             |             |
GRADUATE     | SOUTHERN    | EDUCATED    |             |             |             |             |
             | RHODESIA    | HARTLEPOOL, |             |             |             |             |
             | 1950-55     |  COUSING    |             |             |             |             |
             |             | OF HUSBAND  |             |             |             |             |
             |             |             |             |             |             |             |
             |__ =2/10/1947|__ =28/6/1920|             |             |             |             |__ =7/10/1734
ANDREW     __|  WOLVERHAM  | HERBERT     | WILLIAM   __|             | CHARLES   __|             |  KIMBOLTON
JOHN DARCY      PTON       | JAMES BALL  | FORTESCUE   |             | BALL        |             | ELIZABETH
BALL           BREITA      | 27/3/1892-  | GRAY BALL   |             | 26/2/1778-  |             | COALS
18/8/1965-     MARGARET    | ELECTRICAL  | 17/12/1858- |             |             |             | -13/12/1789
               PEARSON     | ENGINEER,   | 10/10/1936  |             |             |             |
               20/1/1928-  | OFFERRED    | 1884 SGT    |             |             |             |
                           | JOB BY      | ROYAL       |             |             |             |
                           | UNCLE ON    | FUSILIERS,  |             |             |             |
                           | DISCHARGE   | 1887        |             |             |             |
                           | FROM ARMY   | BEERHOUSE   |             |             |             |
                           |             | PROPRIETOR, |             |             |             |
                           |             |             |             |             |             |
                           |             |__ =8/12/1884|             |             |__ =15/9/1771|
               ANTHONY   __| DOROTHY   __|  BAYSWATER  |             | RICHARD   __|  THURLEIGH  |
               PETER         ELIZABETH   | ELIZA JANE  |             | BALL        | SARAH GILES |
               DARCY BALL    BALL        | DARCY       |             | 27/3/1780-  | -11/11/1822 |
                POLICE       23/10/1892- | 22/5/1862-  |             |             | BUR         |
               SERGEANT      7/1/1893    | 6/1/1933    |             |             | 17/11/1822  |
                             BURIED      | CAME FROM   |             |             |             |
                             BUSHBURY    | 'FLEMINGTON'|             |             |             |
                                         |  ST         |             |             |             |
                                         | FLORENCE,   |             |             |             |
                                         | NR TENBY    |             |             |             |
                                         |             |             |             |             |
                                         |             |             |             |             |
                             RICHARD   __|             |             | MARY ANN  __| WILLIAM   __|
                             MORNINGTON                |             | BALL        | BALL        |
                             BALL                      |             | 1/7/1782-   | 22/1/1738-  |
                             7/6/1895-                 |             |             |             |
                             13/1/1933                 |             |             |             |
```

Sample File - Family Tree Printout by Micro-Aid

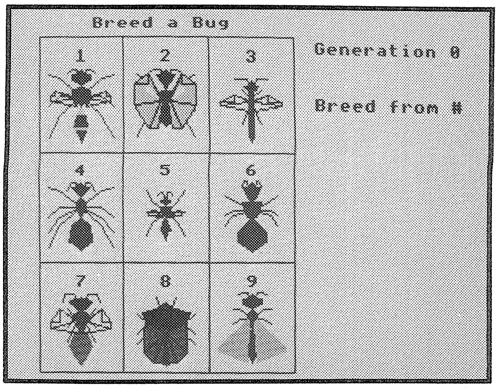
Machine Filing System Tube

Selective Breeding

This program, by P.F. Brampton, demonstrates in superb graphical format the principles of selective breeding applied to the insect world. Whether for serious or light-hearted use, the results are a source of infinite fascination.

The purpose of this program, based on an idea which was the subject of a recent Horizon programme on BBC TV, is to give the user a feeling for the principles involved in the selective breeding of a new variety of insect from an existing species. Also, if you regard yourself as a selection agent, you may attempt to choose those features that will help the insect to survive in a particular environment. By influencing the selection process in this way, the program gives some insight into the workings of natural selection.

Because of memory constraints, asexual reproduction only is simulated, with new mutational forms arising in each succeeding generation. Each generation (of 9 individuals) is displayed graphically on the screen (see illustrations), and one individual is then selected to breed the next generation. The selection process can continue as long as you wish, but there are facilities for saving and re-loading generations as you go along.

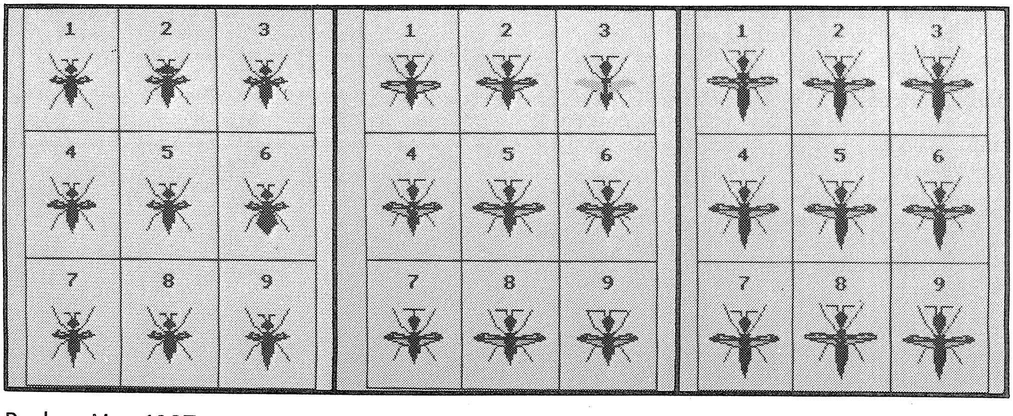


```
Breed a Bug                                  Generation 0

                                             Breed from #
```

USING THE PROGRAM

Type the program in and save it to disc or tape. If the program is then run, you will see nine numbered squares on the left hand side of the screen. In the centre of each square there will be a dot or short line. These dots and lines represent generation zero of the insect species. On the right hand side of the screen a prompt will appear, asking you to select a member of the current generation by typing the number of its square. This member is used to breed the next generation.

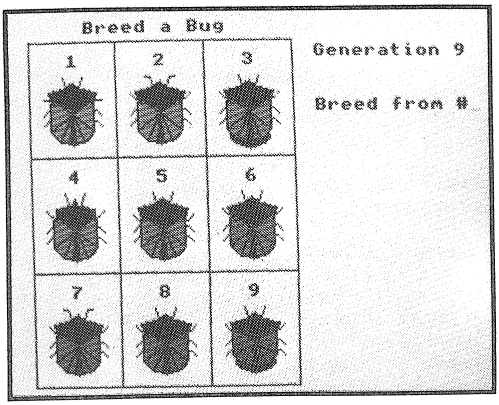
The particular form selected will be passed through to the next generation unchanged, but its eight companions will be mutant forms of the selected insect. The aim of the program is to breed the insect of your choice by selecting from each generation the form that most closely approximates to that which you are aiming for, thereby gradually arriving at the



desired species in a way which simulates certain aspects of Darwinian natural selection and selective breeding.

Two other responses are also allowed. Type Ctrl-S, and the program asks for a suitable filename. The current generation of insects is then saved to disc together with the current generation number.

If you press Ctrl-L, the program asks for a filename, and the insects are reloaded into the computer from the selected file. In this way you can save your insects, and continue later from where you left off. This is useful because two or three hundred generations may be needed to breed a particular variety of insect starting from scratch. It also allows alternative selections to be tried at any point, by saving the current generation for future use.



A file of nine example insects is supplied on the magazine cassette/disc as the file 'exampls'. Using Ctrl-L, these may be loaded and displayed. You may then select one insect from the examples, and perhaps use this as the starting point for the breeding of a new variety. The file "exampls" can also be created by typing in and running the short Basic program Bug-Gen.

HOW THE PROGRAM WORKS

In the program, the data describing the insects and the generation number are stored in RAM between addresses &900 and &AFF. This has the effect of releasing more space for Basic and its variables, and also allows data to be loaded and saved with simple *LOAD and *SAVE commands.

The data describing each insect is stored in the 'dna' strings $(dna+46*n), where n is an integer between 0 and 8. Each string is 43 characters long, each character being upper case alphabetic. Each character can be thought of as a gene controlling some aspect of the insect's size, shape and colour. In use each gene is converted to its ASCII code, and 65 is subtracted to give a number used in calculating dimensions and other features. Numbering the 43 genes 0 to 42 they control the following features.

Genes	Feature
0	Colour of abdomen
1-7	Dimensions of abdomen
8	Not used
9-13	Dimensions of thorax
14	Fourth pair of legs
15-21	Dimensions of head
22-26	Size of antenna
27-33	Size of legs
34	Colour of wings
35-42	Dimensions of wings

The string $tm is also stored in memory between &900 and &AFF. The characters in this string are all 0 or 1, and each character determines the form of mutation suffered by the corresponding character in the $dna strings. If the character in $tm is 0, then the corresponding gene in $dna will be incremented or decremented by the number in N%, when it suffers mutation. This type of mutation is used for genes determining body dimensions. If the character in $tm is 1, then the corresponding gene in $dna will be replaced by a completely new random value. This form of mutation is used for genes that determine colour or absence or presence of some feature.

Other important variables used in the program are as follows:

gxo,gyo	Graphics origins
xi,yi,gx,gy	Scaling factors
xn(5),yn(5)	Array of plotting points
M%	Number of mutations per individual per generation
N%	Size of incremental mutations
tw,tl	Width and length of thorax
hw,hl	Width and length of head

The various functions and procedure perform the following actions:

PROCbug	Draws a complete insect
PROCgrid	Draws grid in which insects are displayed
PROCdisgen	Displays a whole generation of insects
FNmut	Executes M% mutations in a $dna string
PROCclear	Clears an insect from the display area
PROCnewgen	Breeds a new generation of insects
PROCabdomen	Draws an insect abdomen
FNgene	Extracts a selected gene from a $dna string
PROCthorax	Draws an insect thorax
PROCprethorax	Calculates thorax dimensions
PROChead	Draws an insect head
PROCprehead	Calculates dimensions of the head
PROCantenna	Draws the antennae
PROClegs	Draws the legs
PROCwings	Draws the wings

Program INSECT

```
   1Ø REM Program INSECT
   2Ø REM Version BØ.2
   3Ø REM Author  P.Bampton
   4Ø REM BEEBUG  May 1987
   5Ø REM Program subject to copyright
   6Ø :
  1ØØ ON ERROR GOTO 29ØØ
  11Ø MODE1:DIMxn(5),yn(5),oscom 4Ø
  12Ø PROCmain
  13Ø END
  14Ø :
 1ØØØ DEFPROCmain
 1Ø1Ø xi=12:yi=12:gxo=1ØØ:gyo=1ØØ
 1Ø2Ø gx=18*xi:gy=17*yi+64
 1Ø3Ø A%=49:M%=1:N%=2
 1Ø4Ø tm=&9ØØ:dna=tm+46:gen=dna+9*46:!ge
n=Ø
 1Ø5Ø $tm="1ØØØØØØØØØØØØØ1ØØØØØØØØØØØ1ØØ
ØØØØ1ØØØØØØØØØ"
 1Ø6Ø $dna=STRING$(43,"A")
 1Ø7Ø FORi=1TO8:$(dna+i*46)=$dna:NEXT
 1Ø8Ø VDU19,1,6,Ø,Ø,Ø
 1Ø9Ø VDU19,Ø,7,Ø,Ø,Ø,19,3,Ø,Ø,Ø,Ø
 11ØØ PROCgrid
 111Ø VDU28,25,31,39,Ø:COLOUR131:COLOURØ
:VDU12,28,26,31,39,Ø
 112Ø PROCnewgen(A%-49)
 113Ø CLS:PRINTTAB(Ø,4)"Generation ";!ge
n
 114Ø VDU7:IFA%<>19 PROCdisgen
 115Ø PRINTTAB(Ø,8)"Breed from #";
 116Ø REPEAT:A%=GET
 117Ø UNTIL((A%>48)AND(A%<58))OR(A%=12)O
R(A%=19)OR(A%=5):IFA%>48 VDUA%:!gen=!gen+1
 118Ø IFA%>48GOTO121Ø
 119Ø IFA%=12 THEN PROCload
 12ØØ IFA%=19 THEN PROCsave
 121Ø FORi=ØTO8:PROCclear(i):NEXT
 122Ø VDU24,Ø;Ø;1279;1Ø23;
 123Ø IFA%<48 GOTO113Ø ELSE GOTO112Ø
 124Ø ENDPROC
 125Ø :
 126Ø DEFPROCbug(k,xp,yp,xj,yj):LOCALi,j
 127Ø cxo=gxo+k MOD3*gx+xi:cyo=gyo+gy*2-
k DIV3*gy+yi
 128Ø xj=xj DIV2:yj=yj DIV2
 129Ø PROCabdomen
 13ØØ PROCthorax
 131Ø PROChead
 132Ø PROClegs
 133Ø PROCwings
 134Ø ENDPROC
 135Ø :
 136Ø DEFPROCgrid:LOCALyt,xt,i
 137Ø FORi=ØTO 3*gy STEPgy
 138Ø MOVEgxo,gyo+i:DRAWgxo+gx*3,gyo+i
 139Ø NEXT
 14ØØ FORi=ØTO 3*gx STEPgx
 141Ø MOVEgxo+i,gyo:DRAWgxo+i,gyo+gy*3
 142Ø NEXT
 143Ø VDU5:FORi=ØTO8
 144Ø yt=gyo+3*gy-32-i DIV3*gy
 145Ø xt=gxo+gx DIV2-16+i MOD3*gx
 146Ø MOVExt,yt:VDUi+49
 147Ø NEXT
 148Ø MOVEgxo+6*xi,gyo+3*gy+48
 149Ø PRINT"  Breed a Bug"
 15ØØ VDU4
 151Ø ENDPROC
 152Ø :
 153Ø DEFPROCdisgen:LOCALyt,xt,i
 154Ø FORi=ØTO8
 155Ø yt=gyo+3*gy-gy+yi-i DIV3*gy
 156Ø xt=gxo+xi+i MOD3*gx
 157Ø PROCbug(i,xt,yt,xi,yi)
 158Ø NEXT
 159Ø ENDPROC
 16ØØ :
 161Ø DEFFNmut(a$):LOCALp,r,d
 162Ø p=INT(RND(1)*LEN(a$))+1
 163Ø IFMID$($tm,p,1)="1" r=INT(RND(1)*1
Ø) MOD9:GOTO168Ø
 164Ø IFRND(1)>Ø.5 d=N% ELSE d=-N%
 165Ø r=ASC(MID$(a$,p,1))-65+d
 166Ø IFr>16 r=16
 167Ø IFr<Ø r=Ø
 168Ø =LEFT$(a$,p-1)+CHR$(r+65)+RIGHT$(a
$,LEN(a$)-p)
 169Ø DEFPROCclear(n)
 17ØØ VDU24,gxo+n MOD3*gx+xi;gyo+gy*3-n
DIV3*gy-gy+yi;gxo+n MOD3*gx+gx-xi;gyo+gy
*3-n DIV3*gy-64;
```

```
1710 CLG:ENDPROC
1720 :
1730 DEFPROCnewgen(n):LOCALi,j
1740 FORi=0TO8
1750 IFi=n GOTO1810
1760 $(dna+i*46)=FNmut($(dna+n*46))
1770 IFM%<2 GOTO1810
1780 FORj=1TOM%-1
1790 $(dna+i*46)=FNmut($(dna+i*46))
1800 NEXTj
1810 NEXTi
1820 ENDPROC
1830 :
1840 DEFPROCabdomen:LOCALi,j,kk
1850 xn(0)=FNgene(1) MOD10:yn(0)=FNgene
(2) MOD4
1860 xn(1)=FNgene(3) MOD(16-xn(0))+xn(0
)
1870 yn(1)=FNgene(4) MOD(12-yn(0))+yn(0
)
1880 IFxn(1)>0 xn(2)=xn(1)-FNgene(5) MO
Dxn(1) ELSE xn(2)=0
1890 yn(2)=FNgene(6) MOD(15-yn(1))+yn(1
)
1900 yn(3)=FNgene(7) MOD(16-yn(2))+yn(2
)
1910 cx=16*xj+cxo:cy=16*yj+cyo
1920 kk=1:FORi=1TO2
1930 gc=3:g=FNgene(0) MOD4:IFg=2 gc=2
1940 IFg=3 gc=1
1950 GCOL0,gc
1960 MOVEcx,cy:DRAWcx,cy-yn(0)*yj
1970 FORj=0TO2
1980 PLOT85,cx+kk*xn(j)*xj,cy-yn(j)*yj
1990 IFg=2 gc=5-gc
2000 GCOL0,gc
2010 PLOT85,cx,cy-yn(j+1)*yj
2020 NEXTj:kk=-1
2030 NEXTi:ENDPROC
2040 :
2050 DEFFNgene(n):=ASC(MID$($(dna+k*46)
,n+1,1))-65
2060 DEFPROCthorax:LOCALi,j
2070 PROCprethorax:kk=1
2080 FORi=0TO1
2090 GCOL0,3
2100 MOVEcx,cy:MOVEcx,cy+yn(0)*yj
2110 FORj=0TO1
2120 PLOT85,cx+kk*xn(j)*xj,cy+yn(j)*yj
2130 PLOT85,cx,cy+yn(j+1)*yj
2140 NEXTj:kk=-1
2150 NEXTi:ENDPROC
2160 :
2170 DEFPROCprethorax
2180 xn(0)=FNgene(9) MOD10
2190 yn(0)=FNgene(10) MOD8
2200 xn(1)=FNgene(11) MOD(12-xn(0))+xn(
0)
2210 yn(1)=FNgene(12) MOD(10-yn(0))+yn(
0)
2220 yn(2)=FNgene(13) MOD(12-yn(1))+yn(
1)
2230 tw=xn(1):tl=yn(2)
2240 ENDPROC
2250 :
2260 DEFPROChead:LOCALi,j
2270 PROCprehead:kk=1
2280 FORi=0TO1
2290 MOVEcx,cy+tl*yj
2300 MOVEcx,cy+yn(0)*yj
2310 FORj=0TO2
2320 PLOT85,cx+kk*xn(j)*xj,cy+yn(j)*yj
2330 PLOT85,cx,cy+yn(j+1)*yj
2340 NEXTj:kk=-1
2350 NEXTi:PROCantenna
2360 ENDPROC
2370 :
2380 DEFPROCprehead
2390 xn(0)=FNgene(15) MOD10
2400 yn(0)=FNgene(16) MOD(13-tl)+tl
2410 xn(1)=FNgene(17) MOD(12-xn(0))+xn(
0)
2420 yn(1)=FNgene(18) MOD(14-yn(0))+yn(
0)
2430 IFxn(1)=0 xn(2)=0 ELSE xn(2)=xn(1)
-FNgene(19) MODxn(1)
2440 yn(2)=FNgene(20) MOD(15-yn(1))+yn(
1)
2450 yn(3)=FNgene(21) MOD(16-yn(2))+yn(
2)
2460 hw=xn(1):hl=yn(2)
2470 ENDPROC
2480 :
2490 DEFPROCantenna
2500 IFhw=0 xn(0)=0 ELSE xn(0)=FNgene(2
2) MODhw
2510 yn(0)=FNgene(23) MOD(16-hl)+hl
2520 xn(1)=FNgene(24) MOD(12-xn(0))+xn(
0)
2530 yn(1)=yn(0)
2540 xn(2)=FNgene(25) MOD(15-xn(1))+xn(
1)
2550 yn(2)=yn(1)-FNgene(26) MOD15
2560 kk=1:FORi=0TO1
2570 MOVEcx,cy+tl*yj
2580 FORj=0TO2
2590 DRAWcx+kk*xn(j)*xj,cy+yn(j)*yj
2600 NEXTj:kk=-1
2610 NEXTi:ENDPROC
2620 :
2630 DEFPROClegs:LOCALi
2640 htl=tl DIV2:g=FNgene(14) MOD3
2650 xn(0)=FNgene(27) MOD(13-tw)+tw
2660 yn(0)=FNgene(28) MOD(13-htl)+htl
2670 yn(1)=htl-FNgene(29) MOD10
2680 xn(1)=FNgene(30) MOD(15-xn(0))+xn(
0)
2690 yn(2)=FNgene(31) MOD(15-yn(0))+yn(
0)
2700 yn(3)=htl-FNgene(32) MOD5
```

```
 2710 yn(4)=yn(1)-FNgene(33) MOD(15+htl-
yn(1))
 2720 kk=1
 2730 FORi=ØTO1
 2740 MOVEcx,cy+htl*yj
 2750 DRAWcx+kk*xn(Ø)*xj,cy+yn(Ø)*yj
 2760 DRAWcx+kk*xn(1)*xj,cy+yn(2)*yj
 2770 MOVEcx,cy+htl*yj
 2780 DRAWcx+kk*xn(Ø)*xj,cy+htl*yj
 2790 DRAWcx+kk*xn(1)*xj,cy+yn(3)*yj
 2800 MOVEcx,cy+htl*yj
 2810 DRAWcx+kk*xn(Ø)*xj,cy+yn(1)*yj
 2820 DRAWcx+kk*xn(1)*xj,cy+yn(4)*yj
 2830 IFg<>Ø GOTO287Ø
 2840 MOVEcx,cy+htl*yj
 2850 DRAWcx+kk*xn(Ø)*xj,cy+(htl+yn(1))D
IV2*yj
 2860 DRAWcx+kk*xn(1)*xj,cy+(yn(3)+yn(4)
)DIV2*yj
 2870 kk=-1:NEXT
 2880 ENDPROC
 2890 :
 2900 ONERROROFF
 2910 MODE7:IFERR=17 END
 2920 PRINT:REPORT:PRINT" at line ";ERL
 2930 END
 2940 :
 2950 DEFPROCwings:LOCALi,j
 2960 xn(Ø)=FNgene(35) MOD(13-tw)+tw
 2970 yn(Ø)=FNgene(36) MOD(15-htl)+htl
 2980 xn(1)=FNgene(37) MOD(15-xn(Ø))+xn(
Ø)
 2990 yn(1)=yn(Ø)-FNgene(38) MOD(12+yn(Ø
))
 3000 xn(2)=xn(1)
 3010 yn(2)=yn(1)-FNgene(39) MOD(14+yn(1
))
 3020 IFxn(2)=Ø xn(3)=Ø ELSE xn(3)=FNgen
e(40) MODxn(2)
 3030 yn(3)=yn(2)-FNgene(41) MOD(15+yn(2
))
 3040 IFxn(3)=Ø xn(4)=Ø ELSE xn(4)=xn(3)
-FNgene(42) MODxn(3)
 3050 yn(4)=yn(3)
 3060 xn(5)=tw-1:yn(5)=htl-1:IFtw=Ø xn(5
)=Ø
 3070 g=FNgene(34) MOD9:IFg=Ø ENDPROC
 3080 gc=2:kp=85:IFg<3 gc=3
 3090 IFg>5 gc=1
 3100 IF(g=2)OR(g=4)OR(g=7) kp=5
 3110 GCOLØ,gc:PROCpostwings
 3120 GCOLØ,3
 3130 IF(g=5)OR(g=8) kp=5:PROCpostwings
 3140 ENDPROC
 3150 :
 3160 DEFPROCpostwings:LOCALi,j
 3170 kk=1
 3180 FORi=ØTO1
 3190 MOVEcx+kk*tw*xj,cy+htl*yj:MOVEcx+k
k*xn(5)*xj,cy+yn(5)*yj
 3200 PLOTkp,cx+kk*xn(Ø)*xj,cy+yn(Ø)*yj
 3210 PLOTkp,cx+kk*xn(Ø)*xj,cy+yn(1)*yj
 3220 PLOTkp,cx+kk*xn(1)*xj,cy+yn(1)*yj
 3230 PLOTkp,cx+kk*xn(2)*xj,cy+yn(2)*yj
 3240 PLOTkp,cx+kk*tw*xj,cy+yn(5)*yj
 3250 PLOTkp,cx+kk*xn(3)*xj,cy+yn(3)*yj
 3260 PLOTkp,cx+kk*xn(4)*xj,cy+yn(4)*yj
 3270 PLOTkp,cx+xn(5)*xj*kk,cy+yn(5)*yj
 3280 IFkp<>85 MOVEcx+kk*tw*xj,cy+htl*yj
:FORj=ØTO5:DRAWcx+xn(j)*kk*xj,cy+yn(j)*y
j:NEXTj
 3290 kk=-1:NEXTi
 3300 GCOLØ,3
 3310 ENDPROC
 3320 :
 3330 DEFPROCload
 3340 CLS:PRINT'''"Load file"''"Filename
"
 3350 INPUT":"F$:F$="L."+F$+" 9ØØ"
 3360 PROCos(F$):ENDPROC
 3370 :
 3380 DEFPROCsave
 3390 CLS:PRINT'''"Save file"''"Filename
"
 3400 INPUT":"F$:F$="SA."+F$+" 9ØØ +1FF"
 3410 PROCos(F$):ENDPROC
 3420 :
 3430 DEFPROCos(F$)
 3440 $oscom=F$:X%=oscom MOD 256
 3450 Y%=oscom DIV 256:CALL&FFF7
 3460 ENDPROC
```

Program BUGGEN

```
   10 tm=&9ØØ:dna=tm+46:gen=dna+9*46:!ge
n=Ø
   20 $tm="1ØØØØØØØØØØØØØØ1ØØØØØØØØØØØØ1ØØ
ØØØØØ1ØØØØØØØØØØ"
10000 $dna="CCDBFBBFXEDADCXECACBACCGCCBF
HHFJJJCHCCCCCCC"
10010 $(dna+46)="BCEDEBADQCDADBXEFAABACC
GDCBFHFFMJJFFKGEIHJE"
10020 $(dna+46*2)="ABAAHAECXBAAGAXCBBBCB
BXCEBAABDDDGGCGCODBBCC"
10030 $(dna+46*3)="EEDBHBADXCDADCDEEACBA
CCGCCDHHHFJJJAHCCCCCCC"
10040 $(dna+46*4)="ABCBDBBDXCCACBXCBABBA
BBDBBBDEEDFFFCEBBBBBBB"
10050 $(dna+46*5)="AAAGGAEECCACGACCCCECA
CCHBBCEECKCCEAAACACAAA"
10060 $(dna+46*6)="DDCCFCEEQBACGCXCADDCB
BXCEBDABDDDIGCGEQFDACC"
10070 $(dna+46*7)="AGBAICECXAAAAAXIACHHC
CEEAAAKEECCCCIICBBLFEE"
10080 $(dna+46*8)="ACDBFBBFXCDADCXECACBA
CCDCCBDEEDHHHDAAMLCCFG"
10230 *SAVE exampls 9ØØ +1FF
10240 END
```

EUREKA FOR WATFORD

It's been a long wait but Watford Electronics' Eureka board is now available, putting a massive 58K of RAM at your disposal. Dave Somers takes a look at this novel device.

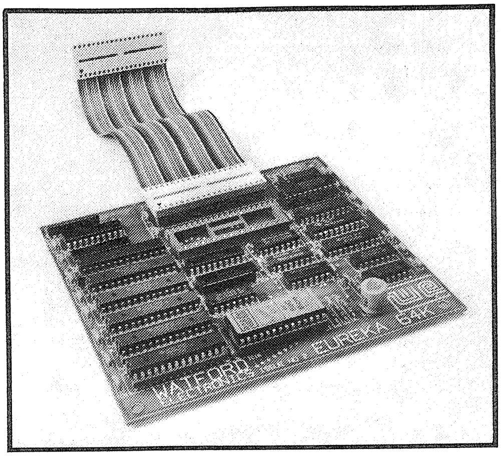
Product : Eureka Board
**Supplier: Watford Electronics,
Jessa House, 250 High Street,
Watford, Herts WD2 2AN.
Tel: (0923) 37774**
Price : £105.80 inc. VAT and p & p.

The BBC model B has often been criticised for its lack of memory. Disc filing systems and other ROMs soon claim vast amounts of RAM, leaving the poor user with hardly any left, and needing to use low-resolution screen modes in order to save memory. Shadow RAM boards have become popular, providing all the storage required for screen displays, thus giving the user more memory for programs and data. However, even with shadow RAM, etc, there is still very little memory available for some applications. To address this need, Watford Electronics has produced its "Eureka" board which allows the model B user to have a staggering 58K of memory at his/her disposal when used with certain popular application 'languages', namely Basic, View, and Wordwise Plus.

The complete package comprises a small circuit board measuring approximately 5 inches square, a sideways ROM, and a 22 page A5 instruction manual.

INSTALLATION

Installation of the Eureka board proved straightforward, but a little fiddly. The 6502 CPU chip should be gently removed and placed into the socket provided on the Eureka board, and the Eureka board should then be stuck to the underside of the lid of the computer using the adhesive pads supplied. The Eureka ROM, which is used to control the RAM board, is fitted into any vacant sideways socket. Finally a flexible lead from the Eureka board must be plugged into the 6502 CPU socket on the main circuit board. Should you have any other expansion boards which use the CPU socket, e.g. Watford Electronics RAM/ROM board, etc., then the Eureka board should be connected to the CPU socket on the expansion board.



WHAT IS THE EUREKA BOARD?

The Eureka board has 96K of memory. This is organised as two 16K banks and one 64K bank. To use the extra memory offered on Eureka, the language to be used, e.g. Basic, View, etc., is copied into one of the 16K banks where it is modified to work with Eureka. The 64K RAM is used by that language as if it were main memory - all totally transparent to the user. The third 16K bank is used by the Eureka board itself to control various functions.

The Eureka ROM supplied contains all the necessary code to carry out the above operation, as well as some useful utilities for manipulating the Eureka's memory.

USING THE EUREKA BOARD

The Eureka board can be used with Basic, View, and Wordwise Plus only. Due to the fact that the language will have to be modified to work on the Eureka board (see above), only certain versions of software can be successfully used, namely:

Basic version II
View version A2.1
Wordwise Plus version 1.4E

To use one of these languages on the Eureka board a star command has to be issued, e.g. *BAS64 for Basic. There is then a short pause while the ROM image is transferred into the Eureka board and then modified. The language then works just as normal, except that it has 58K of memory at its disposal!

USING EUREKA WITH BASIC II

PAGE will remain at its previous value (typically &1900 with DFS fitted). However, HIMEM (the top of memory) is at a staggering &FF00! This means that the user is able to use memory from &1900 to &FF00 for programs, data storage etc., that's just under 58K of memory!

There are, however, some limitations imposed by Eureka on Basic. The most important one is the use of star commands from within a program. These are not allowed - any star commands have to be turned into their OSCLI equivalent, e.g.:

```
    10 *FX 4,2
```

would have to become:

```
    10 OSCLI("FX 4,2")
```

Secondly, any occurrences of *LOAD or *SAVE will have to be converted to their *NLOAD or *NSAVE counterparts (see below).

Thirdly, the use of indirection operators (i.e. ?, !, $) to poke/peek screen, RAM or port values, etc. is not allowed - e.g. ?&7C05=33 to write to the screen will not work. Instead, the official O.S. routines will have to be used (e.g. OSBYTE, OSWRCH, etc). Anyway, this shouldn't be too much of a problem as everyone always writes legal code, don't they?

Fourthly, when using Basic's in-built assembler, any machine code produced will not be placed into Eureka's memory, but will be in the main memory. Should any programs use machine code, they will then have to be modified to work with Eureka.

USING EUREKA WITH VIEW AND WORDWISE PLUS

Eureka imposes no limitations on the operation of View which acts as normal. Wordwise Plus will behave exactly as normal except that the Wordwise Plus programming language is not available!

USING MACHINE CODE ON EUREKA

NO machine code programs will work on the Eureka card - they are always executed in the main memory. However, the Eureka's RAM can be used as a large data store for overlays, data, etc. The Eureka ROM provides an OSWORD call which is used to transfer data between main and Eureka RAM, and vice versa.

THE EUREKA ROM

The Eureka ROM, apart from copying and modifying languages to work on the Eureka board, also contains some useful routines:

*NEDIT is used to invoke the Eureka memory editor. This can be used to examine the memory contents of the Eureka board, and modify them should you wish.

*NLOAD and *NSAVE are used in the same manner as *LOAD and *SAVE. However, instead of main memory being loaded or saved from disc, Eureka memory is loaded or saved. This is particularly useful when loading or saving data into Eureka's RAM.

THE TIME FACTOR

Since it is not possible to access the Eureka RAM directly, all loading and saving operations to disc are slowed down.

Filing System:	DFS	ADFS
LOAD 1K into MAIN memory:	0.39	0.19
LOAD 1K into EUREKA memory:	1.19	0.79
% time increase:	305%	415%
SAVE 1K into MAIN memory:	0.79	1.20
SAVE 1K into EUREKA memory:	4.21	6.21
% time increase:	533%	517%

The above table shows various times for loading and saving data to/from Eureka memory and main memory whilst using the DFS and the ADFS filing systems. Roughly speaking, save operations take five times longer than usual, and load operations take three to four times longer than usual. This is obviously a considerable loss of speed.

Whilst times are being mentioned, it should be pointed out that programs running on the Eureka card execute at the same speed as if they were in main memory.

CONCLUSIONS

The Eureka board is well manufactured and works extremely well. The instruction manual is well laid out and has clear operating instructions. For the techno-

→ 55

Reading *FX calls

David Graham describes a compact function-key routine which reads back data from FX and OSBYTE calls.

Many of the BBC micro's OSBYTE calls may be performed from Basic as FX calls. But FX calls have no way of returning values, so those OSBYTE calls whose function is to READ data from the machine rather than WRITE to it must be performed from machine code. This is sometimes quite inconvenient when you just want to read out the default value of some machine parameter such as the state of the function keys, the keyboard repeat rate or the machine's start-up options.

The routine supplied, of course, solves the problem. It is presented as an EXEC file which may be EXECed in from disc at any time that you need to perform such an operation. The purpose of the EXEC file is to set up function keys fØ and f1 to perform slightly different versions of the same routine. You may if you wish extract the two definitions, and insert them into a Basic program listing, or load the key definitions with *LOAD, having first saved them using

```
    *SAVE keys BØØ BFF
```

(though not on the Master series).

With function key fØ, executed either by EXECing in the file, or by pressing fØ, the user is simply asked for the number of the OSBYTE call, and the result printed out as the contents of two registers X and Y. The major parameter of most OSBYTE calls is read out in the X register, though on occasion, Y also contains valid data, and that is why it has been included here.

For example if you press fØ, and enter 226 as the number of the call, you will probably see the following displayed:

```
    X=  128
    Y=  144
```

The principal result (128) is returned in X, indicating that the shifted function keys will return ASCII values whose base number is 128. The value returned in Y is in fact the base number of the function keys when used with Ctrl.

Some OSBYTE calls require extra parameters to be supplied by the user, and this is why a second version of the FXREAD routine has been included. If f1 is pressed, the user is prompted for the OSBYTE number, as before, but now two further parameters (X and Y) are requested. If you return zero for the first, and 255 for the second, the result will be exactly as with fØ. But the way is open to enter other parameters as appropriate.

The ability to define the Y register when using f1 gives the user a further dimension of control, since setting Y to any value other than 255 will, with many OSBYTE calls, cause a WRITE operation to be performed immediately after the READ, writing the value entered by the user as the X parameter. Thus if you key f1, then enter the following:

```
    226
    2Ø
    Ø
```

you will WRITE the value 2Ø as the new shifted function key state (the equivalent of performing *FX226,2Ø), as you can see if you repeat the call.

For full details of the effect of the full range of FX and OSBYTE calls, the reader is referred to the Advanced User Guide to the BBC Micro, or the Master Reference Manual Part One.

```
MO.4
***************************
*|      OSBYTE READ       |*
*|                        |*
*|  Use fØ/f1 to repeat   |*
*|      version Ø.4       |*
***************************
CO.Ø
*KEYØCO.Ø|MCO.1:REP.:I.'"FX call no "A%:X%
=Ø:Y%=255:!&7Ø=USR(&FFF4):P."X=",?&71'"Y="
,?&72:U.Ø|M

*KEY1CO.Ø|MCO.1:REP.:I.'"FX call no "A%:I.
"X "X%:I."Y "Y%:!&7Ø=USR(&FFF4):P."X=",?&7
1'"Y=",?&72:U.Ø|M

*FX138,Ø,128
```

MIDAS

A Universal disc front end

Install this feature packed utility on all your discs and experience the sophistication and user-friendliness that we have come to expect from software author Jonathan Temple.

MIDAS is an advanced 'front end' for your disc filing system, providing an automatic menu, easy file selection and many disc housekeeping functions for single density DFS format discs. The main display shows all the files on the disc. Any particular file may be referenced by moving a 'selection box' around the screen using the cursor keys. All the facilities available then apply to the currently selected file.

To assist in using MIDAS, general help pages can be displayed by simply pressing the '?' key. These describe all the options available to you, though as usual, experimenting with the use of the facilities will probably be the quickest way to learn just what MIDAS can do.

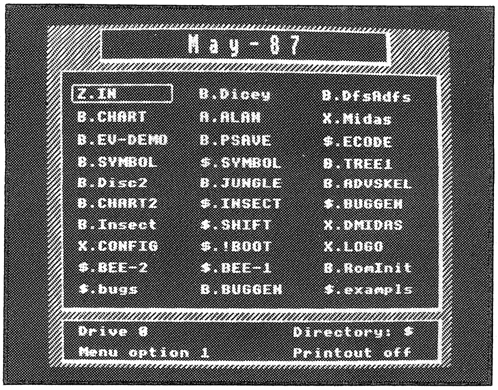
USING MIDAS

Type the program in and save it to disc under the name 'X.MIDAS'. Then run the program, and MIDAS should produce a catalogue of your disc, and await your further response.

The functions available to the user (simply by pressing the specified key) are shown in the table. Where necessary, the program prompts for the additional information required.

The menu option 'M' provides a further six options, which allow you to catalogue all or just selected files. The six options are:

1. All files on disc.
2. All files in current directory.
3. Basic programs only.
4. Machine code files only.
5. Memory dumps only.
6. View/Wordwise files only.



```
                  May-87

Z.IN          B.Dicey       B.DfsAdfs
B.CHART       A.ALAN        X.Midas
B.EV-DEMO     B.PSAVE       $.ECODE
B.SYMBOL      $.SYMBOL      B.TREE1
B.Disc2       B.JUNGLE      B.ADVSKEL
B.CHART2      $.INSECT      $.BUGGEN
B.Insect      $.SHIFT       X.DMIDAS
X.CONFIG      $.!BOOT       X.LOGO
$.BEE-2       $.BEE-1       B.RomInit
$.bugs        B.BUGGEN      $.exampls

Drive 0                 Directory: $
Menu option 1           Printout off
```

A further set of keys is used to select the remaining MIDAS operations. These are as follows:

Ctrl-C.	Compact the disc.
Ctrl-I.	Provide information about disc (free space and whether compaction is possible).
Ctrl-L.	Lock all files.
Ctrl-U.	Unlock all files.
Ctrl-P.	Print catalogue (hardcopy).
Ctrl-Q.	Quit MIDAS.

In addition, the keys numbered Ø to 5 are used to select the disc drive that you wish to work on, and the 'Return' key is used to load the currently selected file from disc, including automatic selection of View or Wordwise as appropriate (using the files' load and execution address). View is entered in mode 7, but you may change this by amending line 137Ø.

MIDAS FUNCTIONS

A.	Alter load and execution addresses of file.
C.	Convert directory: Change directory for all files in a given directory.
D.	Delete file.
E.	Examine file (list or ASCII dump).
H.	Produce hex dump of file.
I.	Provide info on file (file type, size, load and execution addresses and whether locked or unlocked).
L.	Lock file.
M.	Menu files (see later).
N.	Select new directory for Menu files.
P.	Enable/disable printout of file being examined.
Q.	Toggle sound on/off.
R.	Rename file.
T.	Change disc title.
U.	Unlock file.

The program as published is set up to work with drives 0 and 2, without View, Wordwise or a printer. To alter these configurations, list lines 160 to 190 and change the relevant settings. Set each variable to either '0' or '1' with '1' meaning available and '0' meaning not available. Thus, if you have a printer and Wordwise, but not View, you would change line 160 to read:

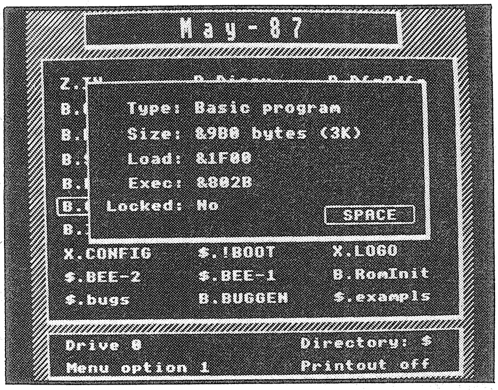
```
160 view=0:wise=1:printer=1
```

The version of the program supplied on the monthly magazine disc/tape differs very slightly from the one published here, in that a configuration program is supplied to allow you to set up the drives, printer and choice of word processor directly. This information is then written directly to the MIDAS program and re-saved as X.MIDAS.

Regardless of which version you are using, always keep a copy of the MIDAS program on the disc under the name 'X.MIDAS', as actions such as compacting the disc require MIDAS to return to Basic, compact the disc, and CHAIN back to 'X.MIDAS'. You should also note that 'X.MIDAS' is the only file on the disc which is not listed in the menu.

```
   10 REM Program MIDAS
   20 REM Version B0.2
   30 REM Author  Jonathan Temple
   40 REM Beebug  May 1987
   50 REM Program subject to copyright
   60 :
  100 found=0
  110 ON ERROR GOTO 3000
  120 MODE 4
  130 HIMEM=&5500
  140 fc=7:bc=4
  150 PROCin
  160 view=0:wise=0:printer=0
  170 drive(0)=1:drive(1)=0
  180 drive(2)=1:drive(3)=0
  190 drive(4)=0:drive(5)=0
  200 PROCsc
  210 REPEAT
  220 PROCg:found=0
  230 COLOUR 0:*INFO X.MIDAS
  240 COLOUR 1:found=1
  250 PROCf:J%=1:K%=1
  260 PROCgt:PROCsh:PROCm
  270 UNTIL G=17
  280 VDU26,12,10:*FX 4
  290 *FX 11,32
  300 END
  310 :
 1000 DEFPROCm
 1010 *FX 4,1
 1020 *FX 11,12
 1030 *FX 21
 1040 J%=1:K%=1:REPEAT Z=-1
 1050 X=J%:Y=K%:GCOL 3,1:G=GET
 1060 IF G=136 IFJ%>1 J%=J%-12
 1070 IF G=137 IFJ%<25 J%=J%+12
 1080 IF G=139 IFK%>1 K%=K%-2
 1090 IF G=138 IFK%<19 K%=K%+2
 1100 N%=J% DIV12+(K% DIV2)*3
 1110 IF N%>=F% J%=X:K%=Y:GOTO1100
 1120 IF J%<>X OR K%<>Y PROCb(X,Y):PROCb
(J%,K%)
 1130 F$=CHR$(D%(N%))+"."+F$(N%):T=T%(N%
)
 1140 IF G=13 PROCld
 1150 IF G=42 PROCos ELSE IF G=47 OR G=6
3 PROCh
 1160 IF G>32 Z=G AND&CF:IF Z>=0 AND Z<6
 GOTO1250
 1170 G=G AND&DF
 1180 IF G=65 PROCab ELSE IF G=67 PROCcd
 1190 IF G=68 PROCde ELSE IF G=69 PROCex
 ELSE IF G=72 PROChd ELSE IF G=73 PROCfi
 1200 IF G=76 OR G=12 PROCac(" L") ELSE
IF G=85 OR G=21 PROCac("")
 1210 IF G=77 PROCms ELSE IF G=78 PROCnd
 ELSE IF G=82 PROCrn ELSE IF G=84 PROCnt
 1220 IF G=80 AND printer P=P EOR1:PROCv
 ELSE IF G=81 THEN *FX210,1
 1230 IF G=9 PROCdi ELSE IF G=16 IF prin
ter PROCpc
 1240 IF G=3 PROCcm
 1250 UNTIL (Z>=0 AND Z<6) OR G=17
 1260 IF G<>17 IF drive(Z)=1 PROCc("DRIV
E "+STR$(Z))
 1270 ENDPROC
 1280 :
 1290 DEFPROCb(J%,K%):MOVE J%*32+48,875-
K%*32:PLOT 1,0,-50:PLOT 1,316,0:PLOT1,0,
50:PLOT1,-316,0:ENDPROC
 1300 :
 1310 DEFPROCld:A$=CHR$34+F$+CHR$34:IF T
=2 OR T=3 PROCc("*L. "+F$):ENDPROC
 1320 IF T=0 PROCk("*L. "+F$)
 1330 IF T=4 PROCbw:PRINT'"Mode number (
0-7) : ";:REPEAT K=GET AND&CF:UNTIL K>=0
 AND K<8:PROCk("MODE "+STR$K+"|M*L. "+F$
)
 1340 IF T=6 OR T=7 PROCk("*"+F$)
 1350 IF T=8 PROCk("CH."+A$)
 1360 IF T=9 PROCk("*WORD.|M:N.|M2"+F$)
 1370 IF T=10 PROCk("*WORD|MMODE 7|ML "+
F$)
 1380 PROCer("This file cannot be loaded
"):PROCv:ENDPROC
 1390 :
```

```
 1400 DEFPROCos:PROCbw:PRINT'" *";:C$=FN
i(0,30,32,126):VDU 28,2,25,37,5,12:PROCc
(C$):PROCs:PROCd:ENDPROC
 1410 :
 1420 DEFPROCab:PROCbw:PRINT'"Alter Load
/Execute address (L/E) ? ";:REPEAT K=GET
 AND&DF:UNTIL K=69 OR K=76:IF K=76 N=2 E
LSE N=6
 1430 PRINT'"Old address: &";:IF N=2 PRI
NT;~L%(N%) ELSE PRINT;~E%(N%)
 1440 PRINT'"New address: &";:A$=FNi(0,4
,33,126):PROCc("ACCESS "+F$):A%=2-(N=6):
X%=0:Y%=&55:!B%=B%+20:B%!N=EVAL("&"+A$):
$(B%+20)=F$:CALL &FFDD:PROCr:PROCf:PROCd
:PROCv:ENDPROC
 1450 :
 1460 DEFPROCcd:PROCbw:PRINT"Old directo
ry: ";:B$=FNid:PRINT'''"New directory: ";
:N$=FNid:FOR N%=0 TO F%-1:F$=B$+"."+F$(N
%):IF D%(N%)=ASC B$ PROCc("ACCESS "+F$):
PROCc("RENAME "+F$+" "+N$+"."+F$(N%)):PR
OCr(N$+"."+F$(N%))
 1470 NEXT:PROCf:PROCd:PROCv:ENDPROC
 1480 :
 1490 DEFPROCde:PROCbw:PRINT'"Delete ";F
$;" (Y/N) ? ";:REPEAT K=GET AND&DF:UNTIL
 K=78 OR K=89:VDU K:IF K=89 PROCc("ACCES
S "+F$):PROCc("DELETE "+F$):S=1:PROCf:PR
OCd
 1500 PROCv:ENDPROC
 1510 :
 1520 DEFPROCex:PROCw(2,2,12,2):PRINT F$
;:PROCw(2,30,37,4):VDU17-P,P:IF T=5 PROC
c("LIST "+F$) ELSE IF T=8 PROCsb ELSE Y%
=OPENUP(F$):CALL&900:CLOSE#Y%
 1530 VDU3,15,10,10:PROCs:PROCsc:PROCsh:
ENDPROC
 1540 :
 1550 DEFPROChd:PROCw(1,30,37,1):VDU17-P
,P:PROCc("DUMP "+F$):VDU3,15,10,10:PROCs
:PROCsc:PROCsh:ENDPROC
 1560 :
 1570 DEFPROCfi:PROCw(6,18,34,7)
 1580 VDU28,9,18,34,8:PRINT"Type: ";T$(T
%(N%));'''"Size: &";~S%(N%);" bytes (";IN
T(S%(N%)/&400)+1;"K)"'''"Load: &";~L%(N%)
;'''"Exec: &";~E%(N%)
 1590 VDU28,7,18,34,16:PRINT"Locked: ";:
IF A%(N%) PRINT"Yes" ELSE PRINT"No"
 1600 PROCs:PROCd:ENDPROC
 1610 :
 1620 DEFPROCac(L$):IF G<32 F$="*.*"
 1630 PROCc("ACCESS "+F$+L$):PROCc("ACCE
SS X.MIDAS L"):PROCrs:PROCf:ENDPROC
 1640 :
 1650 DEFPROCms:PROCw(6,23,34,7)
 1660 VDU 28,9,23,34,8:PRINT"List: "'''"1
 - All files"'''"2 - Directory ";D$'''"3 -
 Basic programs"'''"4 - M/C programs"'''"5
 - Memory dumps"'''"6 - Wordwise/View"
```


May-87

Type: Basic program
Size: &900 bytes (3K)
Load: &1F00
Exec: &802B
Locked: No

SPACE

Z.
B.
B.
B.
B.
B.
B.
X.CONFIG $.!BOOT X.LOGO
$.BEE-2 $.BEE-1 B.RomInit
$.bugs B.BUGGEN $.exampls

Drive 0 Directory: $
Menu option 1 Printout off

```
 1670 PRINT'"Option (1-6) : ";:REPEAT M=
GET AND&CF:UNTIL M>0 AND M<7:VDU M+48:PR
OCf:PROCv:J%=1:K%=1:PROCd:ENDPROC
 1680 :
 1690 DEFPROCnd:PROCbw:PRINT'"New direct
ory: ";:D$=FNid:PROCc("DIR "+D$):IF M=2
S=1:PROCf:J%=1:K%=1:PROCd
 1700 PROCv:ENDPROC
 1710 :
 1720 DEFPROCnt:PROCbw:PRINT'"New title:
 ";:PROCc("TITLE """+FNi(1,12,32,126)+""
""):PROCrs:PROCgt:PROCsh:ENDPROC
 1730 :
 1740 DEFPROCrn:PROCbw:PRINT"Old filenam
e: ";F$'''"New filename: ";:*FX 4
 1750 N$=FNi(3,10,36,126):*FX 4,1
 1760 PROCc("ACCESS "+F$):PROCc("RENAME
"+F$+" "+N$):PROCr(N$):PROCf:PROCsh:ENDP
ROC
 1770 :
 1780 DEFPROCr(N$):IF A%(N%) PROCc("ACCE
SS "+N$+" L")
 1790 ENDPROC
 1800 :
 1810 DEFPROCdi:PROCw(6,16,34,9):PRINT'"
 &";~F;" bytes (";INT(F/&400);"K) free"'
:IF C=0 PRINT" Disc fully compacted" ELS
E PRINT" Compaction possible"
 1820 PROCs:PROCd:ENDPROC
 1830 :
 1840 DEFPROCpc:VDU12,2:PRINT" Drive ";D
;": "T$':FOR Y%=0 TO 9:FOR X%=0 TO 2:N%=
Y%*3+X%:IF N%>=F% X%=2:Y%=9:GOTO1860
 1850 PRINTTAB(X%*12+1);CHR$(D%(N%));"."
F$(N%);
 1860 NEXT:PRINT':NEXT:VDU3:PROCs:PROCd:
ENDPROC
 1870 :
 1880 DEFPROCcm:VDU26,12:PROCk("*COMPACT
|MCHAIN""X.MIDAS""|M")
 1890 :
```

```
 1900 DEFPROCsb:S%=FNBasic:@%=5:C%=OPENU
P(F$):REPEAT B=BGET#C%:IF B<>13 GOTO 194
Ø
 191Ø N%=BGET#C%:IF N%=255 GOTO 194Ø
 192Ø PRINT N%*256+BGET#C%;:FOR M%=1 TO
(BGET#C%)-4:A%=BGET#C%:IF A%=&8D PROCpn
ELSE CALL S%
 193Ø NEXT:PRINT
 194Ø UNTIL N%=255 OR B<>13 OR EOF#C%:CL
OSE #Ø:@%=&9ØA:ENDPROC
 195Ø :
 196Ø DEFPROCpn:A%=BGET#C%:PRINT;((A%*4
AND&CØ EOR BGET#C%)AND&FF)+256*((A%*16 E
OR BGET#C%)AND&FF);:M%=M%+3:ENDPROC
 197Ø :
 198Ø DEFPROCd:VDU 28,2,25,37,5,12:GCOL
3,1:PROCb(J%,K%):FOR Y%=Ø TO 9:FOR X%=Ø
TO 2:N%=Y%*3+X%:IF N%>=F% X%=2:Y%=9:GOTO
2ØØØ
 199Ø VDU 31,X%*12+1,Y%*2+1,D%(N%),46:PR
INT F$(N%)
 2ØØØ NEXT,:ENDPROC
 2Ø1Ø :
 2Ø2Ø DEFPROCsc:VDU26,12:?&DØ=2:FOR N%=1
 TO 1Ø:PRINTSTRING$(128,CHR$224);:NEXT:?
&DØ=Ø:PROCw(6,2,32,1):PROCw(2,3Ø,37,28):
PROCw(2,25,37,5):ENDPROC
 2Ø3Ø :
 2Ø4Ø DEFPROCsh:VDU 28,6,2,32,1,12,31,14
-LEN T$,Ø:A%=1Ø:X%=&7Ø:Y%=Ø:FOR L%=1 TO
LEN T$:?X%=ASC MID$(T$,L%):CALL&FFF1
 2Ø5Ø VDU 23,254,X%?1,X%!1;X%!2;X%!3;X%?
4,23,255,X%?5,X%!5;X%!6;X%!7;X%?8;254,1Ø
,8,255,11,9:NEXT
 2Ø6Ø PROCv:PROCd:ENDPROC
 2Ø7Ø :
 2Ø8Ø DEFPROCv:PROCbw:PRINT"Drive ";D;TA
B(21,Ø)"Directory: "D$'''Menu option ";M
;TAB(21,2)"Printout ";:IF P=2 PRINT"on";
 ELSE PRINT"off";
 2Ø9Ø VDU 28,2,25,37,5:ENDPROC
 21ØØ :
 211Ø DEFPROCer(E$):PROCbw:PRINT E$:PROC
s:ENDPROC
 212Ø :
 213Ø DEFPROCbw:VDU28,3,3Ø,37,28,12:ENDP
ROC
 214Ø :
 215Ø DEFFNid=FNi(1,1,36,126)
 216Ø :
 217Ø DEFFNi(J%,K%,X%,Y%):*FX21
 218Ø L$="":REPEAT L%=LEN(L$):A%=GET:IF
A%=127 IF L%>Ø L$=LEFT$(L$,L%-1):GOTO221
Ø
 219Ø IF A%=13 IF L%>=J% GOTO221Ø
 22ØØ IF A%>=X% AND A%<=Y% AND L%<K% L$=
L$+CHR$(A%) ELSE A%=7
 221Ø VDU A%:UNTIL A%=13:=L$
 222Ø :
 223Ø DEFPROCs:X=?&3ØA-6:Y=?&3Ø9-1:VDU26
:PRINTTAB(X,Y)"SPACE";:GCOL3,1:MOVE X*32
-52,1Ø35-Y*32:PLOT 1,Ø,-5Ø:PLOT 1,26Ø,Ø:
PLOT 1,Ø,5Ø:PLOT 1,-26Ø,Ø:*FX 21
 224Ø REPEAT UNTIL GET=32:ENDPROC
 225Ø :
 226Ø DEFPROCw(A%,B%,C%,D%):a%=A%*32-16:
b%=987-B%*32:c%=C%*32+48:d%=1Ø39-D%*32
 227Ø GCOL Ø,129:VDU 26,24,a%-4;b%-12;c%
+12;d%+4;16:GCOL Ø,128:VDU 24,a%;b%;c%;d
%;16,26,28,A%,B%,C%,D%
 228Ø ENDPROC
 229Ø :
 23ØØ DEFPROCg:?B%=Ø:B%!1=R%:B%?5=1:B%!6
=Ø:B%!1Ø=Ø:B%!14=Ø:A%=6:X%=Ø:Y%=&55:C%=Ø
:CALL &FFD1
 231Ø D=R%?1-48:?(R%+R%?2+3)=13:D$=$(R%+
3):ENDPROC
 232Ø :
 233Ø DEFPROCrs
 234Ø ?B%=D:B%!1=V%:B%!5=&53Ø3:B%!9=34:A
%=&7F:X%=Ø:Y%=&55:CALL &FFF1
 235Ø IF B%?1Ø>Ø ?R%=Ø:R%?1=&BØ:$(R%+2)=
"Disc error "+STR$(B%?1Ø)+CHR$Ø:CALL R%
 236Ø N%=(W%?5)DIV8
 237Ø IF N%<2 PROCer("No menu files in d
rive "+STR$D):D=L:PROCc("DRIVE "+STR$D):
GOTO234Ø
 238Ø L=D:T=W%?13+W%?15+(W%?12=Ø)+16*(W%
?14 AND48)+256*(W%?14 AND3)+1:F=256*(W%?
7+256*(W%?6 AND3)-T)
 239Ø X=2:M%=W%+W%?5:REPEAT X=X-(M%?4<>Ø
)+M%?5+16*(M%?6 AND48):M%=M%-8:UNTIL M%=
W%:C=((T-X)>Ø)
 24ØØ ENDPROC
 241Ø :
 242Ø DEFPROCgt:T$=FNs(V%,V%+7)+FNs(W%,W
%+3):IF ASC T$=Ø T$="No Title"
 243Ø IF RIGHT$(T$,1)=CHR$Ø T$=LEFT$(T$,
LEN(T$)-1):GOTO243Ø
 244Ø ENDPROC
 245Ø :
 246Ø DEFPROCf:PROCrs
 247Ø F%=Ø:M%=V%:G%=Ø:REPEAT
 248Ø G%=G%+1:M%=M%+8:F$=CHR$(?M% AND&7F
)+FNs(M%+1,M%+6):D%(F%)=M%?7 AND 127:IF
D%(F%)=Ø D%(F%)=36
 249Ø IF LEFT$(F$,5)="MIDAS" GOTO265Ø
 25ØØ IF M=2 IF D%(F%)<>ASC(D$) GOTO265Ø
 251Ø A%(F%)=M%?7 AND128:$R%=CHR$(D%(F%)
)+"."+F$:!B%=R%:A%=5:X%=Ø:Y%=&55:A%=USR(
&FFDD) AND &FF
 252Ø L%=B%!12 AND&FFFF:E%=B%!6 AND&FFFF:
S%=B%!1Ø AND&FFFF:L%(F%)=L%:E%(F%)=E%:S%
(F%)=S%:T=Ø
 253Ø IF E%<>L% T=7 ELSE IF L%=Ø T=1
 254Ø IF FNa(&BØØ) T=2 ELSE IF FNa(&CØØ)
 T=3
 255Ø IF L%>&2FFF IF FNa(&3ØØØ) OR FNa(&
4ØØØ) OR FNa(&58ØØ) OR FNa(&6ØØØ) OR FNa
(&7CØØ) T=4
 256Ø IF F$="!BOOT" T=5 ELSE IF A%=255 T
=6
 257Ø IF wise IF (B%?5=&FF OR B%?5=Ø) T=
9
 258Ø IF E%=&8Ø19 OR E%=&8Ø23 OR E%=&8Ø2
B T=8
 259Ø IF view IF D%(F%)=86 OR D%(F%)=118
 T=1Ø
 26ØØ IF M=3 IF T<>8 GOTO265Ø
 261Ø IF M=4 IF T<>6 IF T<>7 GOTO265Ø
 262Ø IF M=5 IF T>4 OR T=1 GOTO265Ø
 263Ø IF M=6 IF T<>9 IF T<>1Ø GOTO265Ø
 264Ø T%(F%)=T:F$(F%)=F$:F%=F%+1
 265Ø UNTIL G%=N%
 266Ø IF F%=Ø PROCer("No files for optio
n "+STR$(M)):M=S:GOTO 247Ø
 267Ø S=M:ENDPROC
 268Ø :
 269Ø DEFFNa(X%):IF X%>&CØØ Y%=&7FFF ELS
E Y%=X%+&FF
 27ØØ =(X%=L% AND (L%+S%=Y% OR L%+S%=Y%+
1))
 271Ø :
 272Ø DEFFNs(J%,K%):Z%=K%?1:K%?1=13:R$=$
J%:K%?1=Z%:=R$
 273Ø :
 274Ø DEFPROCk(K$):PROCc("KEY9 "+K$+"|M"
):*FX 138,Ø,137
 275Ø END
 276Ø :
 277Ø DEFPROCc($B%):X%=Ø:Y%=&55:CALL &FF
F7:ENDPROC
 278Ø :
 279Ø DEFPROCh:RESTORE282Ø:CLS:FOR X%=Ø
TO 1:FOR Y%=Ø TO 6:READ K$,A$:PRINTTAB(X
%*19+2,Y%*2+2)"<"K$">"A$:NEXT,:PROCs
 28ØØ VDU 28,2,25,37,5,12:FOR N%=Ø TO 5:
READ K$,A$:PRINTTAB(4,N%*2+2)"<Ctrl-"K$"
>  "A$:NEXT:PRINT'TAB(4)"<Ø to 5>  Selec
t drive"'''TAB(4)"<Return>  Load file":P
ROCs:PROCd:ENDPROC
 281Ø :
 282Ø DATA A,lter addr,C,onvert dir,D,el
ete,E,xamine,H,ex dump,I,nfo,L,ock,M,enu
 files,N,ew dir,P,rintout,Q,uiet,R,ename
,T,itle,U,nlock
 283Ø DATA C,Compact disc,I,Info on disc
,L,Lock all files,U,Unlock all files,P,P
rint disc catalogue,Q,Quit Midas
 284Ø :
 285Ø DEFPROCin
 286Ø ?&DØ=Ø:CLOSE#Ø:VDU 3,4,17,1,26,12
 287Ø DIM L%(3Ø),S%(3Ø),E%(3Ø),D%(3Ø),A%
(3Ø),T%(3Ø),F$(3Ø),T$(1Ø),drive(5)
 288Ø RESTORE 298Ø
 289Ø FOR N%=Ø TO 1Ø:READ T$(N%):NEXT
 29ØØ VDU 23,224,136,17,34,68,136,17,34,
68,19,Ø,bc;Ø;19,1,fc;Ø;23;1Ø,32;Ø;Ø;Ø;Ø;
 291Ø FOR pass=Ø TO 2 STEP 2
 292Ø P%=&9ØØ:[OPT pass
 293Ø .lp LDA &FF:BMI rt:JSR &FFD7:BCS r
t:CMP #13:BEQ pt:CMP #32:BCC np:CMP #127
:BCS np:.pt JSR &FFE3:.np CLC:BCC lp:.rt
 RTS:]
 294Ø NEXT
 295Ø B%=&55ØØ:R%=&558Ø:V%=&56ØØ:W%=&57Ø
Ø:L=Ø:M=1:S=1:J%=1:K%=1:P=3
 296Ø ENDPROC
 297Ø :
 298Ø DATA Saved memory,Data file,Saved
function keys,Saved characters,Saved scr
een,Auto-boot file,*RUN only M/C,M/C pro
gram,Basic program,Wordwise file,View fi
le
 299Ø :
 3ØØØ COLOUR 1:IF ERR=214 IF found=Ø PRO
Cer("No X.MIDAS in drive "+STR$D):D=L:PR
OCc("DRIVE "+STR$D):VDU28,2,24,37,23:GOT
O21Ø
 3Ø1Ø IF ERR>128 PROCbw:PRINT"Filing err
or:":REPORT:PROCs:CLS:GOTO21Ø
 3Ø2Ø ?&DØ=Ø:CLOSE #Ø
 3Ø3Ø VDU 3,4,17,1,26,12
 3Ø4Ø IF ERR=17 THEN 2ØØ
 3Ø5Ø ON ERROR OFF
 3Ø6Ø *FX 4
 3Ø7Ø *FX 11,32
 3Ø8Ø *FX 12,6
 3Ø9Ø REPORT:PRINT" at line ";ERL'
 31ØØ PROCk("L."+STR$(ERL)+"|K")
 311Ø END
 312Ø :
 313Ø DEFFNBasic
 314Ø BA%=(?&8Ø15)-48
 315Ø IFBA%=1 =&B53A
 316Ø IFBA%=2 =&B5ØE
 317Ø =&BD37
```

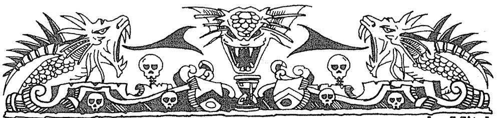
POINTS ARISING POINTS ARISING POINTS ARISING POINTS A

PERSONALISED LETTER HEADINGS (BEEBUG Vol.5 No.8)

With some versions of Wordwise (e.g. 1.4A, 1.4C, 1.4E) the Convert program appears not to work correctly. If you experience a problem, add these two lines to the program:

```
  1264 FOR X%=W% TO P%:IF ?X%=&7F ?X%=&FF
  1266 NEXT
```

These additions should be made ONLY to the Convert program, and NOT the Logo Designer.



by Mitch

TURE GAMES ADVENTURE GAMES ADVE

There has been a dearth of new adventures lately especially in the "Hack and Slash" genre. It's all very well having clever puzzles but the Dragon and I are tired of playing Mr. Nice Guy, we want to kill somebody! Luckily along has come "The KET Trilogy".

Supplier: Incentive Software Ltd.
2 Minerva House,
Calleva Park, Aldermaston,
Berkshire RG7 4QW.
Tel. (07356) 77288
Price : £9.95 (cassette only)

This is a compendium of three adventures in one packet suitable for the BBC and Electron. This is a traditional mix of wizards, orcs and goblins who are all spoiling for a fight. Unjustly condemned for a murder you did not commit, you have been offered the chance of escaping the hangman's noose by undertaking a perilous quest. Naturally you accept! Your task is to travel beyond the mountains and then underground via Vran's Temple to the ultimate confrontation with Vran himself, penetrating his inner sanctum, beyond the Guardians of the Gates....

One reason why these games have such a traditional feel to them is because they are old Spectrum games which first saw the light of day some years ago. The split mode 6 screen, limited text description and two word entry commands is old hat. And a big surprise - the programs are written in Basic - a fact I found out the hard way.

Try typing MOVE PANELS in the panelled corridor and you will receive the merry reply, "No such variable at line 472Ø" and the program will crash.
Dungeonmaster 1 : Incentive Ø.

I was prepared to abandon the games at this point as the limited vocabulary and descriptions seemed so old fashioned but I must confess I found the more I played the more engrossed I became. Dungeonmaster 1 : Incentive 1.

This trilogy was extremely popular when first launched and as I recall was the subject of a large competition. The reason is that the games do contain some very neat puzzles which draw you onwards and hold your interest. There is some unconscious humour in that the game prompts with "What shall WE do now" but when things get tough, its a case of YOU are dead!

Your PROWESS, LUCK and ENERGY totals are displayed to show you when you should choose discretion on approaching some bloodthirsty goblin. There are the inevitable bottles of Strength Potion, suits of Armour and a wealth of weapons for dealing out death and destruction. A small amount of help is built in but this as usual is as confusing as the puzzle you are trying to solve. Each of the games may be played completely separately which means you have three chances to become hopelessly stuck instead of the usual one!

Within the confines of Basic these games are as good as they can be and when you remember you are getting three for the price of one they are probably a fair buy. Adventures on the BBC have moved on since KET was created but it still has the power to grab the imagination and drag you wide-eyed into its dark places.
Dungeonmaster 1 : Incentive 2.

Whilst on the subject of Incentive Software I should mention that they are now offering a Graphic Adventure Creator handbook. For £1.25 they will send you a small book containing additional hints and tips to enable you to get the best out of your G.A.C.

GOOD BYE XEROX

The BBC Adventurer's Handbook from H & D Services has now decided to take the plunge and become a truly professional monthly magazine. Abandoning their former tatty xerox image, the new look mag has now become the glossy "WHAT NOW". The new title has obviously been taken from the prompt used in the old Scott Adam adventures. The magazine will encompass all home micro adventures as opposed to only the BBC, but as most of the larger games are produced for all micros the BBC should be well catered for. Keep a lookout at your newsagents.

FONTAID

Fontaid is another ROM which provides a choice of stylish fonts on your printer. Geoff Bains finds good value in this offering from CJE Micros.

Product : Fontaid
Supplier: CJE Micros
78 Brighton Road, Worthing,
West Sussex BN11 2EN.
Tel: (0903) 213361
Prices : ROM and Disc (13 fonts) - £31
Alternative Font Discs (three, with 11, 13 and 15 fonts) - £16 each
Prices inc. VAT and p & p.

Few owners of the Canon PW-1080 or Taxan KP-810 and KP-910 printers probably realise that these printers are not only capable of NLQ printing in the usual typewriter style but also an alternative NLQ character set which is user definable.

This alternative NLQ requires a 6264 RAM chip to be fitted in the printer, but is otherwise a standard feature of the machines. The trouble is that the codes to define the characters are both tricky to create and tedious to download. This is where Fontaid comes in. Fontaid is both a character definer and a download utility.

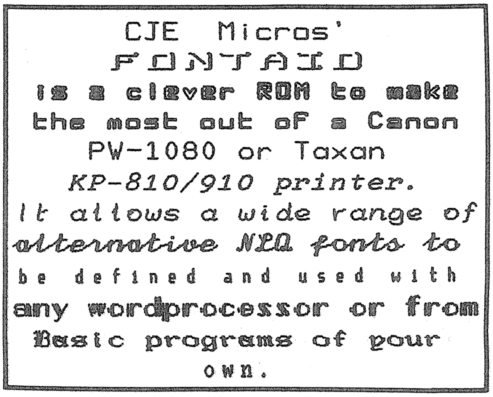
The definer program looks much like any Beeb character definer, except that its character grid comprises 23 by 16 dots. The cursor keys are used to move around the grid, and the space bar toggles a dot on and off in any position. The entire font is displayed next to the character grid, and individual characters can be selected using either their ASCII code or their position on the keyboard, and expanded onto the grid.

Other commands alter the proportional width of a character, shift the grid a line or column at a time in any direction, and print out the font for inspection. Dots cannot be positioned horizontally adjacent as the printers do not allow this. Once the font has been defined it can be saved to disc ready for use with the rest of the Fontaid programs.

Several download utilities are provided, duplicated on disc and ROM. These are variations on a theme. Two (using different workspace areas) enable the printer, download the font, then select it. Two other downloaders perform the same task only if the printer is enabled. This comes in useful when using the embedded codes within a Wordwise document, as it will not then pointlessly execute during a preview.

Other commands are also provided for printing characters, usually reserved for the foreign accented characters, all squeezed up together to form a logo. One command prints characters 0-31 as two rows of 16 characters. The other uses 8-31 in a format of 3 rows of 8, compatible with Watford's NLQ Designer *LOGO format. Fontaid also includes a simple Basic downloader for incorporation in your own Basic programs. A special View printer driver is also included which allows any star command (including the downloaders) to be issued from a View document. The last facility of Fontaid is a program to convert a font file into the printers' ROM format for blowing onto an EPROM and installing permanently in the printer instead of the RAM chip.

Fontaid is a useful buy for any Beeb user with one of these printers. Fortunately, creating your own fonts is simple enough to render the overpriced font discs redundant. With this software an entirely new face can be easily and reliably given to any document.


CJE Micros'
FONTAID
is a clever ROM to make
the most out of a Canon
PW-1080 or Taxan
KP-810/910 printer.
It allows a wide range of
alternative NLQ fonts to
be defined and used with
any wordprocessor or from
Basic programs of your
own.

1st course
Using the Function Keys (Part 2)

C.P. Yu concludes his look at the function keys with an examination of associated FX calls, and a discussion of ways to incorporate complete programs within a single key definition.

THE FX CALLS

There are seven major FX calls associated in one way or another with the function keys, and I will be taking a look at these in turn in this concluding article. Some of this month's material goes beyond the strictly elementary, but it is hoped that even users who have had little to do with function key definitions to date will find something of interest here.

FX18

This is a really simple FX call to begin with. It serves just to clear the function keys of all contents. Like all FX calls it may be issued from within a program, or from so-called immediate mode, by just typing at the keyboard. If the FX call is to be issued from within a Basic program, you must ensure that nothing appears after the call on the particular line of Basic concerned. Thus:

```
    1ØØ *FX18:REM FX CALL
```

will not work, though the following is perfectly acceptable:

```
    1ØØ PRINT A%:*FX18
```

You can of course clear the function keys individually by using (n is key number):

```
    *KEYn <Return>.
```

FX4,2

This FX call sets up the Copy key and the four cursor keys as function keys 11 to 15. They are programmed in exactly the same way as keys Ø to 9, or indeed key 1Ø, the Break key. Of course, if they are in "function key mode" they no longer act as Copy or cursor keys, and if you want to perform normal cursor key editing, you must first reinstate them with:

```
    *FX4  (or *FX4,Ø)
```

FX 225-228

This important set of four FX calls provides the user with a measure of overall control of the function keys. FX225 controls the use of the function keys on their own, while the others control their use in conjunction with the Shift and/or Ctrl keys.

*FX225,n	Function keys alone
*FX226,n	Shift-function keys
*FX227,n	Ctrl-function keys
*FX228,n	Shift-Ctrl-function keys

Table 1

Each of these four calls takes a single parameter, as follows:

n=Ø	Disable keys
n=1	Normal use as function key
n=2-255	Add n to key number to provide ASCII code

Thus if you execute:

```
    *FX225,Ø
```

the function keys will be completely disabled. Pressing them will have no effect whatsoever. Any definitions which they may hold are not however lost, and the keys may be reinstated by executing:

```
    *FX225,1
```

Using *FX225,n with a value of n between 2 and 255 has a different effect. This time, instead of disabling the keys, the call sets them up to generate single characters. The value returned by key number x after issuing *FX225,n is ASCII x+n. In other words, if you execute

```
    *FX225,65
```

key fØ will produce ASCII 65 (65+Ø), an upper case A, key f1 will produce ASCII 66 (65+1), an upper case B, etc.

This particular way of using the function keys has many applications, one of which we glimpsed at last month. There, the ASCII values generated by the keys (in conjunction with Shift or Ctrl) were used to produce teletext control codes to highlight program listings. We can make the function keys perform in a similar way on their own by issuing the call:

```
    *FX225,129
```

This will cause key fØ to return the value 129 (i.e. 129+Ø) which is the teletext code for red, f1 will give green, etc.

The range of values of n given above have exactly the same effect with the other three FX calls, except of course, that they affect the way in which the function keys will behave in combination with Shift and Ctrl, rather than on their own. Thus to make the shifted function keys return the normal function key strings, just issue:

```
    *FX226,1
```

If you now program key Ø with any string, you will find that this string appears on screen when you press Shift-fØ (i.e. hold down the Shift key, and touch function key fØ).

Unfortunately, you cannot program the function keys with one set of definitions, and the shifted function keys with another. Only a single set of definitions is allowed, and these are accessed by the keys alone, or in combination with Shift and/or Ctrl depending on the setting of FX225-228. To take this to extremes, if you issue the following four calls:

```
    *FX225,1
    *FX226,1
    *FX227,1
    *FX228,1
```

you will find that the function keys alone, or with Shift and/or Ctrl all generate the same set of user-defined strings.

If you want to reset the keys to their default settings, you only have to press Break. Ctrl-Break, which will clear any stored key definitions, is not required, though this will also reset them. The default settings are worth noting, and are as follows:

```
    *FX225       1
    *FX226     128
    *FX227     144
    *FX228       Ø
```

Note that the excellent Advanced User Guide for the BBC Micro apparently has a small slip here. It gives the default for Shift-Ctrl as 16Ø. It is actually zero.

FX138

This FX call is somewhat different from those which we have looked at so far in that its operation is not specific to the function keys. However, when used with the function keys it becomes a very powerful programming tool. Its effect is officially defined as inserting value v into buffer n when called with a pair of parameters as follows:

```
    *FX138,n,v
```

This sounds really rather unexciting. But when you set n=Ø to designate the keyboard buffer, and v=128 to specify function key zero, issuing the call will now simulate the pressing of function key zero. To test it out, try the following:

```
    *KEYØ "Key fØ pressed"
    *FX138,Ø,128
```

As soon as you have pressed Return on the FX call, you will see the message printed to the screen, indicating that a press of key fØ has indeed been simulated.

This technique immediately provides a number of interesting possibilities. For a start, it can be used to run a piece of code from within a program which would normally only operate from immediate mode. For example, we have occasionally provided a so-called move down routine with some of our longer programs, so that they may be loaded in from disc, and then moved down over the disc work space to give them enough room in which to run. This relies heavily on the use of FX138. Table 2 gives the listing for such a routine.

```
1 IF PAGE<&EØ1 THEN 1Ø
2 *K.Ø *T.|MFORA%=ØTO(TOP-PAGE)STEP4:
  A%!&EØØ=A%!PAGE:NEXT|MPAGE=&EØØ|MOL
  D|MRUN|M
3 *FX138 Ø 128
4 END
```

Table 2

It first checks to see if a move-down is necessary, and if so, it programs key fØ with the complete move-down routine (line 2). Line 3 then calls FX138 to execute the routine held in fØ. The use of a function key for this purpose is quite essential, because the move down could not be run from the program itself: that would be like trying to lift yourself up by pulling at your own boot straps. But it works when called from the function key.

The END statement on line 4 is also very important, because FX138 does not take effect until the END of a program. This makes sense, since Basic will not expect an immediate mode command until any running program has finished. Once the move down has taken place, the program in fØ then executes a RUN command. But now, PAGE has been set to &EØØ, so line 1 makes no attempt to re-execute the move down, and the program proceeds from line 1Ø.

COMPLETE PROGRAMS IN FUNCTION KEYS

Both the above example, and one or two of the examples given last month actually consist of complete programs executed from within a function key. In fact the function keys can be seen as offering the user a second level of programming for short utility programs. A program held in a function key can operate completely independently of any program held in user memory, and may even be called to operate upon such a program. For example a very useful "find string" utility could be set up in this way.

SINGLE KEY BINARY TO HEX

As a further example of a complete function key program, I shall use a binary to decimal and hex conversion program. This in itself can be very useful under certain circumstances, and has none of the attendant complications of a utility such as a string search. The complete key definition is given in Table 3. Before it becomes at all legible, you need to mentally decode the abbreviations which have been used. They are as follows:

I.	INPUT	N.	NEXT
F.	FOR	P.	PRINT
M.	MID$(		

The last one is the only really fiendish one. It effectively enters the code for MID$, which includes the left hand bracket. This means that the bracket count in the key definition appears to be quite wrong. But when you press f0, it all works ok, and you will be prompted for a binary number. Any string of ones and zeros of up to 31 characters in length may be entered. In fact any character other than "1" is taken as a zero for the purposes of the program. The result is then printed out as a decimal and hex equivalent.

```
10 *KEY0 I."Bin "A$:N=0:L=LEN(A$):F.A=1T
OL:N=N-2^(L-A)*(M.A$,A,1)="1"):N.:P.SPC1
5;N;" dec";SPC5;"&";~N;" hex"|M
```
Table 3

There is one major point to note with this program: even though the program is only one line long, it is able to stop for user input. This would not work, however, if the individual statements which make up the program were separated with |M instead of a colon. In that case, the statement immediately after the INPUT statement would be taken as user input, with a consequent foul-up of major proportions. This is also the case when using GET within a function key (or indeed an EXEC file, which suffers from a similar problem of an un-stoppable input stream).

ENHANCEMENTS

When you press any function key, it prints its contents to the screen, and only then, if it is terminated with |M, will it begin to act upon them. Sometimes it is vital that the key's contents should be displayed. With complex definitions such as the binary converter, this is just a nuisance. Getting rid of the surplus display is an easy matter, providing you are not in mode 7. You just insert a couple of COLOUR statements at the very start of the definition as follows:

```
*KEY0 CO.0|MCO.7:I."Bin "A$ etc.
```

You may conceivably also wish to insert a mode change at the very start to ensure that you are not in mode 7. The colour change works because the colour is immediately set to black because of the |M following the CO.0. It is then reset to colour 7, so that text printed by the key definition is legible to the user, but this does not take effect until the key definition has been printed out in black across the screen.

As a second enhancement, you may want to put an audible prompt at the INPUT statement. This may be accomplished by inserting the following:

```
SO.1,-9,20,3:
```

immediately before the INPUT statement I."Bin " etc.

```
10 *KEY0 CO.0|M:CO.7:REP.:SO.1,-9,20,2:I
."Bin "A$:N=0:L=LEN(A$):F.A=1TOL:N=N-2^(
L-A)*(M.A$,A,1)="1"):N.:P.SPC15;N;" dec"
;SPC5;"&";~N;" hex":U.0|M
```
Table 4

Finally, you may wish to make the program auto-repeat. The simplest way to achieve this is to place the routine within a REPEAT UNTIL loop. Table 4 gives the enhanced version, containing the modifications suggested here. To escape from the new program, just press Escape.

Again we have all too quickly reached our allotted space limit. It is hoped that some of the ideas covered here will encourage the more effective harnessing of one of the Beeb's under-used resources.

WATFORD'S 512 Co-processor Interface

Watford Electronics has released yet another interface for the model B and B+. All you need then is a 512 co-processor to turn your Beeb into an IBM PC compatible. A gimmick or a breakthrough? Peter Rochford reports.

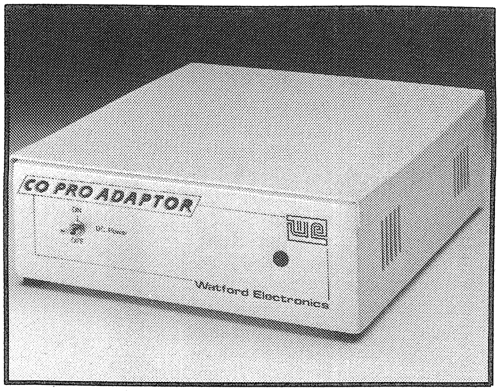
Product : Co-Pro Adaptor
Supplier: Watford Electronics
Jessa House, 250 High Street
Watford, Herts WD1 2AN.
Tel. (0923) 37774
Price : £63.25p inc. VAT and p & p.

There must be a fair number of model B owners who would welcome the chance to be able to connect one of the Master co-processors to their machine. This is the thinking behind the release of the latest Beeb add-on from Watford Electronics, called the Co-Pro Adaptor.

What this unit does, is to allow either the Master 512 co-processor or the Master Turbo co-processor to work with a model B. In addition, the well-heeled Master owner could have a triple processor machine, with for instance, a Turbo inside the machine and a 512 residing in the Co-Pro Adaptor.

The Co-Pro Adaptor is a free-standing, mains operated unit housed in a sturdy BBC-beige metal case, measuring 220mm x 200mm x 65mm. The front panel features a DC power on/off switch and LED indicator. At the rear is a captive mains lead complete with plug and a ribbon cable measuring approximately 11 inches for connection to the TUBE port.

The top of the case is a hinged lid and provides easy access to the interior for fitting and removal of a co-proces- sor. Installing a co-processor is quite simple. The pins on the underside of the co-processor board plug into two sockets on the main Co-Pro Adaptor circuit board. This does need reasonable care as it is only too easy to mis-align pins and socket, as there is no locating keyway.



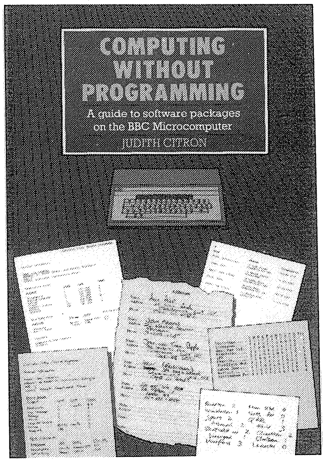
Once installed correctly, all that is required is for the ribbon cable to be connected to the TUBE port. This can be REALLY fiddly as the ribbon cable is so short. With everything installed and connected, powering up the adaptor and then pressing Ctrl-Break, activates the co-processor.

In use the Co-Pro worked perfectly and there were no system crashes or anything else untoward. I did test the unit on both my Master and a model B. With the Master, I had my Turbo board inside the Master and the 512 installed in the adaptor. For me, the ability to do this is a godsend as I can now utilise both my co-processors without having to plug and unplug them inside my Master.

Model B owners can now buy a Turbo board and the Co-Pro adaptor for less than the price of the old Acorn 6502 second processor. The Turbo of course, apart from being cheaper, is also faster. The use of a 512 with the model B I find a bit dubious. There is no numeric keypad on a model B, and the 512 software configures this to provide certain key-functions normally found on IBM and MS-DOS machines.

In conclusion, I think that the unit is excellent and should find favour with both model B and Master owners alike. I should add, that at the time of writing, I hear that Acorn are to release their own adapator for co-processors. It will be more expensive than the Watford unit at £86.25 (inc. VAT). I hope to obtain one as soon as possible for review.

COMPUTING WITHOUT PROGRAMMING


COMPUTING WITHOUT PROGRAMMING
A guide to software packages on the BBC Microcomputer
JUDITH CITRON

At last a book for the user rather than the programmer. "Computing without Programming" is primarily aimed at the educational market, but Mark Sealey finds much of interest to all users.

'Computing Without Programming' by Judith Citron, published by Chapman and Hall at £9.50

This book is intended for teachers and students working on Information Technology and Computer Appreciation courses in TVEI, CPVE, for RSA qualifications and for GCE or GCSE. It also contains much relevant information for all users of the application packages described. The idea behind it is to lead the complete beginner through the six software packages examined, and into tackling a complete and extended project. In that the book is not just a guide but sets out to be a source for project ideas, it is a success.

The six packages examined comprise word processors (Wordwise Plus and Edword2), databases (Quest and BEEBUG's Masterfile II) and spreadsheets (BEEBUG's Quickcalc and Ultracalc 2). In all cases, individual ideas for project work are examined after outlining the features, tricks and rationale of each package. Each section ends with a very useful, exhaustive and accurate semi-technical comparison between the two items of software. Applicability to the B+ and Master series is included.

Each section, on a different application, begins with an outline of the general principles involved. Only then is there any detailed examination of specific syntax. Criteria that place learning first are clearly uppermost - as in this on page 195: "Educationally, spreadsheets encourage an understanding of the relationship between numbers, and promote questioning and demonstrate the need for estimation skills". And the chapters on spreadsheets make it easy to do this; for example the clear way in which the author holds your hand as you enter the results of football matches with Ultracalc 2 and then revise them once you are confident with the basics.

The book uses designed frames rather than photographs to show what should be on the screen. This is especially useful and well done, for instance, in the word processing section, and amply illustrates the three advantages that the author sees of using such software (physical: neatness, educational: planning without drudgery, and social: flexibility and collective creativity). Indeed, the instruction side of the book is admirable. The proverbial difficulties, for instance, of simultaneity for beginners in Control key pressing and Shift-Break are well dealt with! The 'Special Features' of Masterfile II are arranged (as are other chapters on advanced facilities) so that you can skip them without harm if your needs are more basic. Recourse can later be made to the book's adequate index.

There are no exercises as such and the 'Compendium of Ideas' for each package could well have been a little fuller. It is good, though, to see mention made of presenting the same data in more than one of the three formats - for example recipes in both the word processor and the database, or holiday bookings in both spreadsheet and database form.

All in all, a good buy this. Over 25Ø pages for almost a tenner. Perhaps a little over-priced for the student market but 'Computing without Programming' deserves to become a standard textbook for Schools and Colleges teaching the relevant syllabus using these packages. And in that they are somewhat standard and established packages, this is a clear, handy, readable book which I can thoroughly recommend.

Keep your finger on the pulse of the Electron and BBC Micro market

at the

ELECTRON & BBC MICRO USER SHOW

Royal Horticultural Hall Westminster London SW1

Nearest Tube: Victoria

Friday to Sunday, May 8 to 10

Doors open at 10am each day
Close 6pm Friday and Saturday, 4pm Sunday

Much has happened in the six months since the last London show. Don't miss this opportunity to keep right up to date with all the new products now available – plus lots more bargains at rock bottom prices!

It's the 16th record breaking BBC show

SAVE 50p

Bring this coupon to the Show to get 50p per person off the normal admission price of £3 (adults), £2 (children).

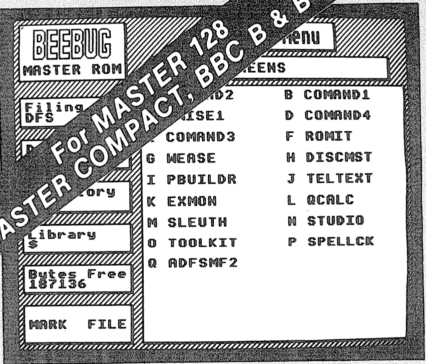
Valid for up to four people.

50p OFF

10am-6pm, Friday, 8 May
10am-6pm, Saturday, 9 May
10am-4pm, Sunday, 10 May
Royal Horticultural Hall, Westminster, London SW1

BUG

For MASTER 128
MASTER COMPACT, BBC B & B+



```
BEEBUG           Menu
MASTER ROM       EENS
Filing           D2         B COMAND1
DFS              ISE1       D COMAND4
                 COMAND3    F ROMIT
                 G WEASE    H DISCMST
                 I PBUILDR  J TELTEXT
ory              K EXMON    L QCALC
                 M SLEUTH   N STUDIO
Library          O TOOLKIT  P SPELLCK
$                Q ADFSMF2
Bytes Free
187136
MARK   FILE
```

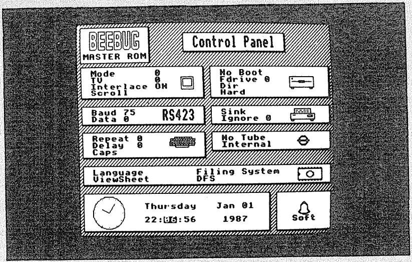
DISC MENU

A powerful disc menu giving complete control over disc files, and especially useful to ADFS users for easy movement through ADFS directories.

- Auto-run Basic and machine code programs.
- Automatically load Wordwise, View and Interword files.
- Scan ADFS directories, and step back to previous directory levels.
- Mark files for deletion, copying, re-naming etc.
- DFS and ADFS compatible.
- 40 or 80 column display.

DIARY & ALARM

A computerised diary for keeping track of appointments and important dates. Includes a monthly calendar display, allowing a diary to be kept for each day of the year. You can step backwards and forwards through the months and years, and set alarms for any dates.



```
BEEBUG        Control Panel
MASTER ROM
Mode      0             No Boot
TV        0             Fdrive 0
Interlace ON            Dir
Scroll                  Hard
Baud 75    RS423        Sink
Data 0                  Ignore 0
Repeat 0                No Tube
Delay  0                Internal
Caps
Language        Filing System
ViewSheet       DFS
     Thursday    Jan 01
     22:00:56    1987        Soft
```

CONTROL PANEL

A complete visual display of computer status allowing a whole range of settings to be examined and edited.

- Analogue clock display allowing clock to be adjusted.
- Comprehensive ROM/RAM display with options to insert, unplug and save ROMs.
- Alter configuration settings held in CMOS RAM (Master 128 only).
- Load/save status settings to disc.
- 40 or 80 column display.

SIDEWAYS RAM COMMANDS

A selection of commands to allow you to make more use of your sideways RAM. Use it as a printer buffer so that you can continue to use your computer for other work at the same time as printing long documents. It may also be used as a silicon disc, to load and save Basic or machine code programs to sideways RAM. A combination of both printer buffer and silicon disc can be selected.

DISC COMMANDS

The Master ROM contains a selection of disc commands to provide many useful features normally only available on the utility disc. Included are commands to backup ADFS discs, selectively copy files between the ADFS and DFS, catalogue whole ADFS discs, format, verify, and much more.

Although the Master ROM was designed for the Master 128, most features are available on the Master Compact, BBC B and B+. To make full use of the Master ROM, your BBC B or B+, should be fitted with the ADFS and sideways RAM.

BEEBUGSOFT

Master ROM

NEW

The Master ROM is a powerful new ROM developed to enhance the ADFS, sideways RAM and real time features of the Master 128. It has been further enhanced to enable users of the BBC Micro to benefit from its wide range of utilities.

Price £39.00

MEMBERS PRICE £29.25

Supplied on ROM with spiral bound manual.

BEEBUGSOFT

Command

the ultimate communications ROM for the BBC Micro and Master 128.

Command is a very special command driven communications ROM, with a powerful extended instruction set. All major features are available at the press of a key, but because the ROM may be command driven, it is very easy to link instructions together to create your own customised communications software.

COMMAND has to be one of the most comprehensive and flexible pieces of modern software around...

It works impressively well, without any problems...

A & B COMPUTING MAY 87

...has an amazing 50 commands in its repertoire.

...it's one of the best Beeb terminal packages to have been released in a couple of years, and full of potential.

MICRONET

It is very flexible and powerful...

Despite its incredible power it is easy and simple to use.

EDUCATIONAL COMPUTING MARCH 87

VIEWDATA TERMINAL

A full feature Prestel Terminal offering:

- ☆ pull down help screen
- ☆ real time clock
- ☆ Epson screen dump
- ☆ telesoftware downloader
- ☆ frame tagging
- ☆ mailbox send facilities
- ☆ frame load/save

SCROLLING TEXT TERMINAL

Ideal for accessing Telecom Gold and Bulletin Boards.
Features include:

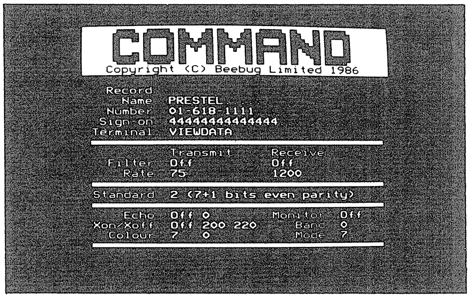
- ☆ 40 or 80 column operation
- ☆ pull down help and status screens
- ☆ user to user communication
- ☆ spool or print incoming text
- ☆ split screen operation
- ☆ XMODEM file transfer
- ☆ Xon/Xoff protocol supported

SPECIAL FEATURES

Telephone Directory — allows name, numbers and modem settings to be stored on disc for easy re-call.

Viewdata Editor — offers a full range of editing commands, a pull down help panel, and a pixel editor.

Auto-Answer facility — ideal if you wish to operate your own bulletin board.



```
COMMAND
Copyright (C) Beebug Limited 1986

   Record
     Name  PRESTEL
   Number  01-618-1111
  Sign-on  444444444444444
 Terminal  VIEWDATA

           Transmit    Receive
   Filter  Off         Off
     Rate  75          1200

 Standard  2 (7+1 bits even parity)

     Echo  Off 0       Monitor  Off
 Xon/Xoff  Off 200 220    Band  0
   Colour  7   0          Mode  7
```

Command Telephone Directory

Price £39.00

MEMBERS PRICE £29.25

COMMAND is supplied on ROM with a 76 page spiral bound manual, and function key strip.

COMMAND works with all BBC compatible modems, although the Auto-Dial/Auto-Answer facilities are designed for the Beebug Magic Modem and other similar models (e.g. Kirk Enterprise, Watford Apollo, Voyager 7).

COMMAND INSTRUCTION SET

☆ANSWER	☆DOWNLOAD	☆MON	☆SAY	☆TXRATE
☆BAND	☆ECHOFF	☆PAUSE	☆SDUMP	☆UPLOAD
☆BCLEAR	☆ECHON	☆PCLOSE	☆SEND	☆V21A
☆BDUMP	☆FTP	☆PROFF	☆SPOFF	☆V21O
☆BLOAD	☆GCLOSE	☆PRON	☆SPON	☆V23A
☆BSAVE	☆GRAB	☆PSCREEN	☆SPCLOSE	☆V23A/O
☆CALL	☆GROFF	☆RETRY	☆STANDARD	☆VEDIT
☆CONNECT	☆GRON	☆RINGS	☆STAT	☆VIEWDATA
☆DIRECT	☆GSCREEN	☆RXFILTER	☆TEXT	☆XOFF
☆DISCONNECT	☆LISTEN	☆RXRATE	☆TXFILTER	☆XON
☆DISPLAY	☆MOFF			

Personal Ads

BBC B with double density disc interface, 40/80 track 400K drive, ATPL ROM expansion board with Wordwise Plus, Wordease, Watford NLQ, Spellcheck III, Dumpout and Toolkit. Lots of disc software including Fleet Street Editor, Masterfile II and many games. Books, manuals and all back issues of BEEBUG. Total cost over £1250, will accept £375 ono for the lot. Tel: (05827) 67157 evenings (St.Albans area).

TANDON 80 track double sided fast setting drive, cased £87.50 (carr inc). Beebugsoft: Toolkit Plus ROM £22.50, Masterfile I cassette £10, Design disc £15. Lightpen and utilities disc, unused £20. IFEL sideways RAM and utilities disc, printer buffer etc £16. Salamander Turbo Basic mini compiler tape £10. All in good condition, original manuals etc. Tel: 01-863 6641 (evenings).

CANON colour printer, as new, any sensible offer near £200 - original cost £400. Tel: 01-304 5290.

MICROPOLIS 5.25 inch disc drives, double sided, 40/80 track. New and tested, £65 plus p&p. Tel: 061-861 8255 (eves) or 061-236 3311 ext. 2982 (day).

ACORNSOFT revs game, condition as new £7 ono. Tel: 01-952 7244.

ACORN Bitstick £235, boxed, as new. Global products P8000 industrial EPROM programmer £275. Tel: Paul, Derby (0332) 362798.

TOOLKIT Plus ROM £18. Sleuth ROM £12. Both complete with manuals. Recent versions. Various BBC B programs (tape and disc). Available as a result of upgrade to Master. Cumana CS100 40 track disc drive in original box with manual and 10 discs £70. Tel: (0302) 744005.

BBC keyboard for sale. Ideal as spare for any BBC owner £30 ono.

COMMODORE PET with twin disc drives £100. Two RML 480Z link machines £120 each. Tel: Paul 01-521 6444.

ARIES B32 RAM board as new, purchased just before Christmas complete with manual £55. Solidisk dual disc interface (1770 and 8271) new and completely unused, purchased mid January £25 no offers. Tel: Ashley (0322) 64761 after 7 p.m.

SHUGART 100K 40 track disc drive £45. Tel: Basingstoke (0256) 26699.

VIEW 2.1 word processor complete with manuals £30 ono. Spellcheck III 80 track £20 ono. White Knight Mk 12 cassette £5. Tel: (0795) 873082.

SOFTWARE on ROM and disc for BBC B. Books for BBC B. 20 items must go. Tel: (0269) 851035 after 6p.m. and weekends.

SYSTEMATICS stock control and invoicing (business duo) with manual £50. Wordwise Plus ROM with manuals and tape £20. Movie Maker, 2 ROMs and manual £15. Quickcalc, Spellcheck, Design, Teletext, Starter Pack, Discmaster all £5 each. BEEBUG back issues and magazine discs. Tel: Mark (0702) 331802.

WORDWISE Plus (latest), brand new with a blank registration card, manuals, fkey strip £30. Tel: 021-552 2670.

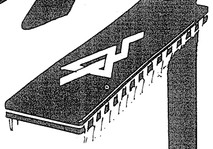
FOR SALE: Disc Doctor, Caretaker, Graphics Extension and Communicator ROMs complete with manuals £10 each. Gemini Beebcalc tape £5. Also quantity of books at half price or less, please ask for list. Tel: Derek Preece on Royston (0763) 42593 evenings.

ATPL with 16K RAM and DFS shift disc £25. Watford ZIF and Help ROM £12. Sleuth and Exmon II £12 each. Spellcheck II 40 track £17.Toolkit PLus £15. Tandem 40 track SS D/Drives plus two identical for spares £60 (buyer collects or £10 p&p). Tel: Newbury (0635) 62391 eves.

WORDWISE Plus ROM, manuals and typing tutor £26. Forth cassette and book £9. '30 Hour Basic' book £4. 13 single sided 48tpi discs £10. Tel: Bedford (0234) 218918.

→34

create programs ten times... Faster



Imagine, 150 extra commands instantly at your disposal to enable you to write programs in a fraction of the time normally taken. They can be fully relational with five files open SIMULTANEOUSLY. Well, it's a reality!

Your programs will even run faster and more efficiently. We wrote a CARD INDEX program using SYSTEM DELTA which outdates all other databases with its revolutionary features. It's fast, it's highly efficient and we are giving it away FREE OF CHARGE with each SYSTEM DELTA.

We also wrote several support applications, such as MAILSHOT, REPORTER and INTER/VIEW LINK to make the software the only flexible, modular, fully integrated office system available today. What's more it can even be customised for individual requirements by any BBC BASIC programmer!

Imagine, a powerful, fast database capable of finding any single record from a file of over 8000 records in less than 3 seconds! Or search for a group matching any multiple criteria which may even be any BBC BASIC expression or evaluation. Load in MAILSHOT to produce labels up to 4 across or link with a wordprocessor for personalised letters. Use REPORTER to create any report which is not available within standard databases, eg. totals, averages or even debit and credit columns. It's even more powerful than dBASEIII Report Writer!

If you require a graph then create it with SYSTEM DELTA in a short (13 line) BASIC program. Non-programmers can simply use the INTER/VIEW LINK to transfer the data in ANY way required including sum, averages or even the Top Ten number of occurances. Data can also be transferred into a spreadsheet in the same way.

We also used SYSTEM DELTA to write the most flexible, usable, integrated ACCOUNTING SYSTEM available today. Invoice layouts can be defined, it copes with Mail Order and Credit customers archiving all details for future mailshots, stock is managed with useful reports such as economic order quantities and even pareto graphs, cashbook entries can be output into a spreadsheet for cashflows, etc. Again, each module can be customised to suit individual requirements.

We also created specialist applications such as SCHOOL ADMINISTRATOR, HOTELIER, VIDEO RENTAL and NEWSAGENT which are also flexible and may be customised. All files are compatible so they may be loaded into Mailshot, Inter/View Link, etc.

Each person requires something different from their software and the only answer is flexible, programmable software. What's more the only available solution is SYSTEM DELTA - the only truly relational Data Management System. If you are not convinced and want to know more complete the box below and we will be pleased to send you full specifications by return.

Please send me by return ☐ or send full details ☐

SYSTEM DELTA with free CARD INDEX			£64.95	MAILSHOT	£19.95
INTER/VIEW LINK	£19.95	REPORTER	£19.95	NOMINAL LEDGER	£55.95
ORDERS/INVOICES	£45.95	SALES LEDGER	£45.95	PURCHASE LEDGER	£45.95
STOCK MANAGEMENT	£45.95	NEWSAGENT	£54.95	VIDEO RENTAL	£54.95
SCHOOL ADMINISTRATOR	£69.95	HOTELIER	£69.95	PROGRAMMERS MANUAL	£19.95

Cheque/Access No.......................Signature..............
NAME..
ADDRESS...
....

MINERVA SYSTEMS, 69 SIDWELL ST, EXETER, DEVON. EX4 6PH. Tel. 0392 37756

←32

WANTED: Payroll program on 40 or 80 track disc to run on BBC B+. For up to 25 employees on weekly payments. Please contact R.S.Ireland, 8 Moughland Lane, Runcorn, Cheshire WA7 4SE or Tel: Runcorn 74041.

UPGRADING to a Master so must sell BBC B with Acorn DFS, Wordwise and Disc Doctor. Will accept £220 ono. Tel: Roy (0247) 473002 evenings or weekends.

WANTED: ROMs - Interchart, Intersheet and Interword and manuals. Also Acorn Teletext Adaptor (N.B. Micro User offer new £74.95). Fleet Street Editor for sale 80 track, new/unused £30. Tel: Robin, Lincoln (0522) 752458 evenings.

DISC drives: Cumana 80 track £50. Canon 40/80 track £65. ROM Software: Acornsoft GXR, View; Beebugsoft Toolkit Plus, Sleuth, Exmon II; Computer Concepts Accelerator, Extension Graphics, Speech. All original £10 each. Aviator for BBC B

£5. ATPL ROM expansion board with battery backup £25. Tel: Dorking (0306) 889647 evenings.

BBC B with Acorn DFS and Zenith green screen monitor £300. Torch Graduate Second Processor (IBM compatible) with twin 360K 5.25" floppy disc drives. Original software and manuals for MSDOS and Psion Xchange (database, spreadsheet, wordprocessor and graphics) package. GDFS software to allow drives to be used with BBC software £300.

SOLIDISK FourMeg RAM/ROM expansion board and DFS ROM £35. Wordwise Plus £21. All with handbooks and original information. Mr.W.Jackson, 17 Glenwood Park, Conway, Dunmurry, N.Ireland BT17 9DT.

ISLAND Logic Music System 40/80 track discs for the BBC B (system disc and sound library) £20. Tel: (0272) 843409.

BBC B, Acorn DFS, Aries B32 shadow RAM board £290 the lot. Possible split. Tel: Sheffield (0724) 587658 after 6pm.

WANTED. Micro-Pulse ROM box. G.Sears, 33 Old Hall Lane, Longsight, Manchester M13 0TH. Tel: 061-224 2828.

WORDWISE Plus, new with registration card etc. £30. Spellcheck II, 80 track dictionary £20. Commstar £20. Tel: 01-514 2684 after 6pm.

ORIGINAL ROMs with manuals. Computer Concepts Graphics Extension (1.03), Sleuth (1.06) £12 each. Tel: 061-439 9665.

VIEW A2.1 ROM complete with original manual £25. DFS 1.2 ROM £10. Watford View printer driver generator for NLQ (disc) £5. Tel: Guildford (0483) 576466 (room 11) or send to B.Pokrud, Surrey Court, University of Surrey, Guildford, Surrey GU2 5XH.

WANTED: Grafpad2, old type with E/S switches, for digit image analysis. Tel: (04427) 72092.

TOOLKIT Plus £25. Studio 8 40/80 track disc £10. Discmaster 40/80 track £10.

Adrian Mole game 80 track disc £5. Citadel 40 track disc £5. Thrust 40/80 track disc £5. Contact: Ben Hallifax, 8c Spencer Hill, London SW19 4NY.

BRAND new - never used. Panasonic KX-P1080 printer, plus paper and cables £150 ono: Cumana CS400 40/80 track plus 20 discs £140 ono: Interword £35 ono: All in original packing. BBC model B with numerous software (games and teaching) £300 ono. Tel: Phil (04946) 6161 ext. 160 (day) or 01-200 0480 (evening).

BBC B, DFS, SWR board, books, mags (Acorn and Micro User '84 to '87), Replay ROM, Mini Office II, Fontwise +, and many more ROMs and utilities. Free software with orders. For prices Tel: Frank, Erith (03224) 30179.

ACORN Z80 Second Processor, almost new (proof of purchase date). Complete with all software and manuals. Best offer around £180 secures. Tel: Folkstone (0303) 56469.

WS2000 modem, BBC B interface cable and manual. All as new. Ideal comms package - only £99. Tel: Andrew (0803) 526058

Business Ads

QUESTIONNAIRE EVALUATION: Schools and researchers. A suite of programs designed to create, analyse and interrogate data files, providing printed and graphics output. Full documentation supplied. State 40/80 track BBC/Master £18 (inc. p&p). Q-EVAL, 432 Station Road, Dorridge, Solihull, West Midlands B93 8EU.

H-A-E-M For all biology students - and nurses!! 11 options, including: blood basics, blood cells, haemoglobin, blood groups, clotting, rhesus factor, multiple choice test and much more. Send for details or state disc (40 or 80) with cheque for £22 to:Mountain Software Resources, 27 Castlecroft Ave., Blackrod, Bolton BL6 5BA.

CHESS EXPERT SYSTEM. A complete storage, retrieval and analysis system for chess games with White Knight/Colossus interface. Master/Model B compatible. Available with 350K openings database. Send £1 (stamps accepted) for demonstration disc to Bernard Hill, Hawthorn Bank, Scott's Place, Selkirk TD7 4DP.

MARKS AND STATISTICS for handling students' marks. Widely used in schools, colleges. Spreadsheets format, calculates totals, sorts, analyses, normalizes etc. Prints lists, histograms. BBC B, Master. 40/80 track. £17. In-Form, 73 Woodfield Park, Colinton, Edinburgh EH13 0RA.

EURO TEXT & COMMUNICATIONS EDITOR

the all-in-one text editor

Provides editing, printout and communications facilities all integrated in a single easy to use program. Plus extended character set for foreign languages or symbols, fast file search for mailmerging or vocabulary searches, and comprehensive printer routines giving close control over printout.

- ★ Full *wysiwyg* screen editing in 80 col. mode typically with max. 3950 chars. current text.
- ★ Built-in communications for scrolling data bases (e.g. TGOLD) or user to user.
- ★ File search and incorporate routine (search speed 1000 chars. per sec.)
- ★ Insertion of printer codes anywhere on a line.
- ★ Printout on any daisy or matrix printer (Epson compatible matrix printer required for full range of user defined characters).
- ★ Pause during printout, enter text or codes directly from keyboard, or abort printout.
- ★ In 16K ROM for BBC B with DFS.

A truly functional tool for text and message handling

£28.75 INC VAT

EDINBURGH TRANSLATIONS LTD.

14 Stafford Street, Edinburgh EH3 7AU

Tel. 031-225 8965

Advertising in Beebug

For advertising details, please contact

Yolanda Turuelo
on (0727) 40303

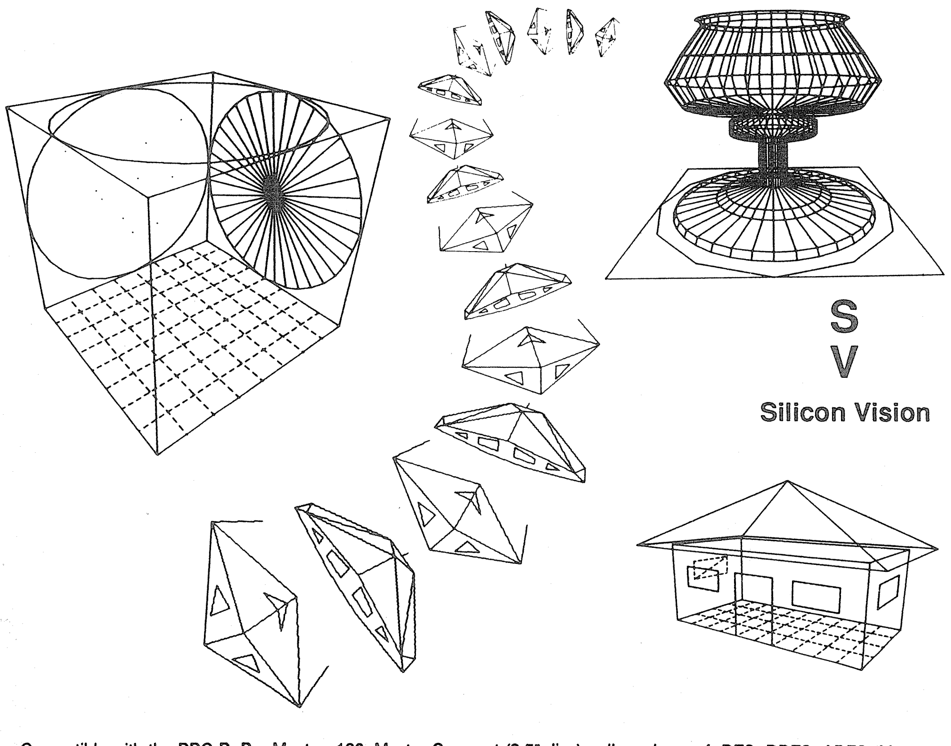
or write to:
Dolphin Place
Holywell Hill
St. Albans
Herts. AL1 1EX

REALTIME GRAPHICS SYSTEM

32k GRAPHICS ROM

"there is no other software to rival it" - BEEBUG

The *REALTIME GRAPHICS SYSTEM* provides a sophisticated design tool for creating 3D objects of any complexity. Colour hardcopy of the designs can then be produced on a range of popular pen-plotters or printers for professional results. Unlike other packages, this system also includes a unique *32k REALTIME GRAPHICS LANGUAGE (RGL)* rom allowing you to produce stunning 3D animation from your own Basic or Assembler programs. The package consists of a 32k Graphics Rom, five discs and a 115 page manual. The design system takes all the hard work out of designing complex 3D objects by providing *curves and surface generators* that can create any solid of revolution (not limited to circles and cylinders), *recursive macros* to allow a 'building block' design approach, *dynamic* 3D viewing, *Printer Driver, 3D Multi-Plotter Driver* currently supporting Plotmate, Penman, Epson HI-80 and Sakata, *Library* facilities and a *Data convertor* for interfacing to other CAD systems and applications. The *RGL* provides 44 star commands for high speed flicker-free 3D animation. These include 3D rotation, scaling, translation, perspective/isometric projections, orbiting, 3D Turtlegraphic and Geometric calculation commands for mathematical and scientific plots. Also includes a unique 35,000 pixels/sec line generator (considerably faster than Acorn's 9000/sec) for high speed drawing. The *RGL* is compatible with all the BBC graphics screen, plotting and colour modes including Shadow screen. The Prisma 2, Cadsoft workstation, Pluto and CMS colour cards can also be driven for line generation in excess of 1,000,000 pixels/sec.



S
V
Silicon Vision

Compatible with the BBC B, B+, Master 128, Master Compact (3.5" disc), all versions of DFS, DDFS, ADFS (Master 128/Compact), 6502 2nd Processor/Turbo (optional), AMX Mouse (optional). Available from your local dealer or send cheque/P.O. for £79.95 (inclusive) to: SILICON VISION LTD, 47 DUDLEY GARDENS, HARROW, MIDDX, HA2 ODQ.
Telephone: 01-422 2274

THE MASTER PAGES

Devoted to the Master Series Computers

This month, apart from the usual crop of Master hints, we have two very handy utilities. The first is an archiving facility for transferring files from DFS to ADFS.

Our second program will be a boon to Master users of sideways RAM: it enables you to initialise ROM images without pressing Ctrl-Break. Compact users do not have this problem, but they may still benefit from the program's ability to initialise the UNPLUG and INSERT status of their ROMs.

David Graham

Contents

MASTER SERIES

DFS to ADFS multi-file transfer

Use this utility by Alan Webster to archive copies of all your old DFS discs.

One of the major advantages of the ADFS over the DFS is its ability to store very large numbers of files. This is a consequence both of its unlimited number of directories, and its greater capacity. Because of this it can often be used to store the contents of a considerable number of DFS discs. The Master series machines provide no very easy way of backing up a DFS disc on to the ADFS, and this is where the present utility, Archive, comes in handy. Whether you are using it to save disc space by storing the contents of ten magazine discs all on a single ADFS disc, or just to copy a handful of a friend's DFS programs on to your ADFS, you will find this utility a great time-saver.

USING ARCHIVE

First of all, type in the program, and save it to your ADFS disc. When the program is run it asks you to put the DFS source disc into drive one, then press any key. When you have done this, Archive's front panel will appear. This displays the catalogue of side one of the DFS disc, with a letter against each filename. A number of options are also presented.

Before any copying is carried out, you should ensure that the desired ADFS destination directory has been selected. This is achieved in one of two ways. The default option is for the destination to be the currently selected ADFS directory on drive zero. You may change the currently selected directory (but not the drive) by issuing a star command from Archive's front panel. Just use:

```
    *DIR $.Fred
```

to put you in $.Fred. You may wish to create new directories for the copying process. Again use the star command facility from the front panel. Using:

```
    *CDIR $.George
```

will create $.George. But remember to follow this with:

```
    *DIR $.George
```

if you wish to establish this as your currently selected directory. The other way of establishing the ADFS destination directory is to use option 1 on the menu to select your destination directory specifically, regardless of the currently selected ADFS directory. To do this, key 1, then type:

```
    $.Henry  or whatever
```

Once you have established your destination directory, copying may commence. To copy single files, just key the letter against the required filename. If you want to perform a global copy, then key 2. You are then asked if you want a "selective" global copy or not. If you reply in the negative, every file on the DFS disc will be transferred in sequence, with a green marker indicating the file currently being transferred. If you enter a "Y", then you will be prompted for a "Y/N" response before each file is transferred. Even locked files are transferred, though not the special locked files on dual format discs.

There are four further options on the front panel not yet covered. The Tab key is used to swap surfaces of the DFS disc, and display the catalogue of the newly selected surface; while the Copy key allows you to change DFS discs - though the front panel is too crowded to display this option. Keying 3 from the front panel displays the amount of free space on your ADFS disc - an obviously useful option, since you may well be trying to cram as many DFS files on to a single disc as possible. Lastly, key 4 is used to quit the program.

PROGRAM NOTES

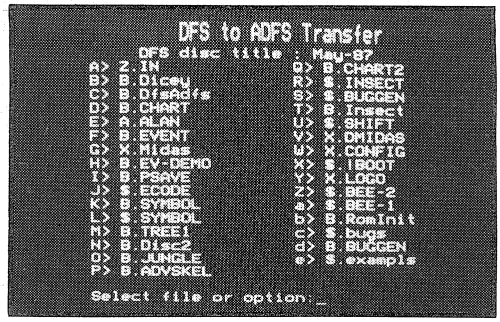
The program uses *LOAD and *SAVE to copy programs between the two filing systems, using a buffer of length &5000 set up in user RAM. This is much quicker than using the versatile *MOVE. However, this method is not suitable for particularly long files. A check is therefore made of the length of each file to be transferred, and if this exceeds 20K, the much slower *MOVE is invoked.

If Archive hits a disc error it will beep and display a message, and will return you to the ADFS root directory - so remember to reset your destination directory after a disc error of any kind.

DFS directory letters are appended to the beginning of corresponding filenames before copying to the ADFS unless the files are in DFS directory $. Three examples are given to clarify this:

$.TEXT4	becomes	TEXT4
B.PROG1	becomes	BPROG1
C.TEST	becomes	CTEST

Using Archive, you should find that you can recycle quite a number of DFS discs. Our record so far is 24 DFS discs on a single ADFS disc. If you do store the contents of any quantity of DFS discs on to a single ADFS disc, it is even more important to take a backup, since the potential loss from a corrupted catalogue or a "broken directory" is that much greater.



```
              DFS to ADFS Transfer
          DFS disc title  : May-87
  A> Z.IN                  Q> B.CHART2
  B> B.Dicey               R> $.INSECT
  C> B.DfsAdfs             S> $.BUGGEN
  D> B.CHART               T> B.Insect
  E> A.ALAN                U> $.SHIFT
  F> B.EVENT               V> X.DMIDAS
  G> X.Midas               W> X.CONFIG
  H> B.EV-DEMO             X> $.!BOOT
  I> B.PSAVE               Y> X.LOGO
  J> $.ECODE               Z> $.BEE-2
  K> B.SYMBOL              a> $.BEE-1
  L> $.SYMBOL              b> B.RomInit
  M> B.TREE1               c> $.bugs
  N> B.Disc2               d> B.BUGGEN
  O> B.JUNGLE              e> $.exampls
  P> B.ADVSKEL

  Select file or option:_
```

```
   10 REM Program DFS to ADFS transfer
   20 REM Version B1.8
   30 REM Author  Alan Webster
   40 REM BEEBUG  May 1987
   50 REM Program subject to copyright
   60 :
  100 D$="@."
  110 MODE 7:DIM buffer 512,setup 10,F$(
31),program &5000,prog 15,prog2 10
  120 MODE 7:cat=FALSE:drv=1
  130 ON ERROR GOTO 220
  140 PROCmessage
  150 ON ERROR GOTO 220
  160 OSCLI("FX4,1"):REPEAT:PROCtransfer
  170 IFG%<>9 THEN 200
  180 IF drv=1 drv=3 ELSE drv=1
  190 cat=FALSE:PROCreadcat
  200 UNTILG%<>9:MODE7
  210 OSCLI("FX4"):END
  220 ON ERROR OFF:OSCLI("FX4"):*ADFS
  230 global=FALSE:message=TRUE
  240 CLS:IF ERR=17 AND cat THEN 150
  250 IF ERR>128 REPORT:PRINTCHR$7''TAB(
4)"Press any key to return to menu";:IFG
ET
  260 IF ERR>128 AND cat THEN 150 ELSE I
F ERR>128 THEN 130
  270 MODE7:REPORT:PRINT" at line ";ERL
  280 END
  290 :
 1000 DEFPROCmessage
 1010 FORM%=0TO1:PRINTTAB(1,M%)CHR$129;C
HR$157;CHR$131;CHR$141TAB(9)"DFS to ADFS
 Transfer";TAB(38);CHR$156:NEXT
 1020 VDU28,0,24,39,2
 1030 global=FALSE:message=TRUE:CLS
 1040 PRINTTAB(0,8)CHR$134"Place ADFS di
sc in drive 0 and DFS disc"'CHR$134"in
drive 1. This program will copy"'CHR
$134"files from drive 1 (DFS) to dri
ve 0"'CHR$134"(ADFS)."TAB(7,20)CHR$131"P
ress any key for menu";:IFGET
 1050 OSCLI("FX202,0,32"):*FX118
 1060 PROCreadcat:*ADFS
 1070 ENDPROC
```

```
1080 DEFPROCtransfer
1090 CLS:PRINTTAB(3,0)CHR$132CHR$157CHR
$135"DFS disc title : ";FNtitle;TAB(34)C
HR$156
1100 PROCkeys:A%=buffer
1110 FORF%=1TO(?(buffer+&105))/8
1120 A%=A%+8:F$(F%)=FNread(A%)
1130 PROCdisp(F$(F%),F%):NEXT
1140 IF global THEN G%=R%:R%=R%+1:GOTO1
260
1150 PRINTTAB(0,18)CHR$131"Select file
or option:";
1160 G%=GET:IFG%=9 ENDPROC
1170 IFG%=135 $&7C00=D$:CLEAR:D$=$&7C00
:GOTO110
1180 IFG%=ASC("4") ENDPROC
1190 IFG%=ASC("3") PROCfree:GOTO1090
1200 IFG%=ASC("*") PROCstar:GOTO1090
1210 IFG%=ASC("1") PROCdirect:GOTO1090
1220 IFG%=ASC("2") PROCglobal:GOTO1090
1230 IFG%>96 AND G%<102 G%=G%-6
1240 G%=G%-64:IF G%<1 OR G%>F%-1 THEN 1
160
1250 PRINTTAB(0,18)STRING$(30,CHR$32);
1260 IF NOT message PRINTTAB(0,18)CHR$1
31"Copying..."
1270 IFG%>(?(buffer+&105))/8 THEN globa
l=FALSE:message=TRUE:GOTO1090
1280 R$=F$(G%):R2$=LEFT$(R$,1):IFR2$="$
" R2$=""
1290 R2$=R2$+RIGHT$(R$,LENR$-2):*FX21
1300 IF NOT message THEN 1320
1310 PRINTTAB(0,18)CHR$131"Transfer ";R
$;" as ";R2$;" (Y/N):";:REPEAT:G=GET:UNT
IL G=78 OR G=89 OR G=110 OR G=121:VDUG,1
3,10:IF G=78 OR G=110 THEN 1090
1320 PROCinfo(R$):IF L%>&5000 THEN 1390
1330 REM *LOAD and *SAVE
1340 OSCLI("DISC"):OSCLI("DR."+STR$drv)
1350 A$="L."+R$+" "+STR$~program
1360 OSCLI(A$):*ADFS
1370 A$="SA."+D$+R2$+" "+STR$~program+"
+"+STR$~L%+" "+E$+" "+RE$
1380 OSCLI(A$):GOTO 1410
1390 A$="MOVE -DISC-:"+STR$drv+"."+R$+"
-ADFS-"+D$+R2$
1400 OSCLI(A$):*ADFS
1410 A$="ACCESS "+D$+R2$+" WR"
1420 OSCLI(A$):GOTO 1090
1430 ENDPROC
1440 DEFFNread(B%)
1450 E$=CHR$((B%?7)AND&7F)+"."
1460 FORC%=B% TO B%+6
1470 E%=?C%:IFE%=32 C%=B%+7:NEXT:=E$
1480 E$=E$+CHR$E%:NEXT:=E$
1490 DEFPROCreadcat
1500 OSCLI("DISC"):OSCLI("DR."+STR$drv)
1510 X%=setup MOD 256:Y%=setup DIV 256
1520 ?setup=drv:setup!1=buffer
1530 setup?5=3:setup?6=&53:setup?7=0
1540 setup?8=0:setup?9=&22
1550 A%=&7F:CALL&FFF1
1560 IFsetup?10<>0 drv=1:?buffer=0:buff
er?1=129:$(buffer+2)="Not DFS Formatted"
+CHR$0:CALLbuffer
1570 cat=TRUE:OSCLI("ADFS"):ENDPROC
1580 DEFPROCdisp(S$,S%):C%=S%+64
1590 IF S%>16 T%=20:S%=S%-16 ELSE T%=0
1600 VDU31,T%,S%,131:IFC%>90 C%=C%+6
1610 VDUC%,62:IF(R%=C%-64 OR (R%>26 AND
R%=C%-70)) AND global VDU130 ELSE VDU13
5
1620 PRINTS$:ENDPROC
1630 DEFPROCstar
1640 VDU12,10,42:INPUT""A$:OSCLI(A$)
1650 PRINT''"Press Space to continue:";
1660 IFGET:ENDPROC
1670 DEFPROCdirect
1680 VDU13:PRINTCHR$131"Enter ADFS dest
ination (@ for current)"'CHR$131":";
1690 INPUT""D$:D$=D$+".":ENDPROC
1700 DEFPROCglobal
1710 global=TRUE:R%=1:*FX21
1720 VDU13:PRINTCHR$131"Selective mode
(Y/N) :";:message=TRUE:H%=GET
1730 IFH%=78 OR H%=110 THEN message=FAL
SE:ENDPROC ELSE ENDPROC
1740 DEFFNtitle
1750 T$="":FORM%=0TO7:T%=buffer?M%
1760 IF T%<32ORT%>126 T%=32
1770 T$=T$+CHR$T%:NEXT:FORM%=&100TO&103
1780 T%=buffer?M%:IFT%<32ORT%>126 T%=32
1790 T$=T$+CHR$T%:NEXT:=T$
1800 DEFPROCinfo(W$):*DISC
1810 OSCLI("DR."+STR$drv)
1820 A%=5:X%=prog MOD256:Y%=prog DIV256
1830 $prog2=W$:!prog=prog2:CALL&FFDD
1840 RE$=STR$~(!(prog+2)AND&FFFF)
1850 E$=STR$~(!(prog+6)AND&FFFF)
1860 L%=!(prog+10)AND&FFFF:*ADFS
1870 ENDPROC
1880 DEFPROCkeys
1890 PRINTTAB(0,20)CHR$131"1."CHR$134"A
DFS Destination Dir"CHR$135"<";:IFD$="@.
" PRINT"Current>" ELSE PRINTLEFT$(D$,LEN
D$-1);">"
1900 PRINTTAB(0,21)CHR$131"2."CHR$134"G
lobal Transfer";
1910 PRINTTAB(20)CHR$131"3."CHR$134"*FR
EE"TAB(30)CHR$131"4."CHR$134"Quit"
1920 PRINTCHR$131"*."CHR$134"OS Command
";TAB(15)CHR$131"Tab."CHR$134"DFS Drive
1/3"CHR$135"<";drv;">";
1930 ENDPROC
1940 DEFPROCfree
1950 VDU28,0,24,39,20,12,10
1960 OSCLI("ADFS"):*FREE
1970 PRINT'TAB(4)"Press any key to retu
rn to menu";:IFGET
1980 VDU12,28,0,24,39,2:ENDPROC
```

MASTER SERIES

Master Hints

The latest in hints and tips for the Master and Compact compiled by David Graham.

EXEC File Auto-Save

The Master Editor seems the obvious place to create and edit EXEC files on the Master. The following line, placed at the very top of any EXEC file provides an auto-save option from within the Editor, yet is completely ignored when the file is EXECed in:

```
    *|.>filename
```

You may add spaces between the "." and the ">" for clarity. Once in the Editor, pressing f3 (Save), followed by Return will save the file under the filename "filename". Pressing f2 (Load) then Return will attempt to load in a file of the same name.

The same technique can be used in a Basic program on a REM line, thus:

```
    10 REM Saved under .>filename
```

But this is less useful, since the Editor saves Basic programs in an untokenised form, and they cannot then be reloaded back into Basic. It is usually more useful to return to Basic before saving an edited program.

Printer Buffer Length

From your letters we know that Tim Powys-Lybbe's Master printer buffer published in the December issue has proved to be very popular. Users may like to know a quick way of returning the current size of the buffer. Tim writes that ADVAL(-4) can be used, just as with the Beeb's resident buffer. This can be useful, since it effectively tells you how many 16K banks have been allocated for buffer use. Just type:

```
    PRINT ADVAL(-4)
```

from immediate mode Basic. If the buffer is not engaged the result will be 63. If it is, then the number of banks in use may be obtained by dividing the result by 16384. If the buffer currently contains any text this will be reflected in the result returned. Of course, if you are in a word processor, you will have to exit first.

ADFS Directory Copy

The following will copy the whole contents of the ADFS directory $.TEXT on drive one to the currently selected ADFS directory:

```
    *COPY :1.$.TEXT.* @
```

The useful "@" symbol means "current directory". Of course, you can specify the block of files to be transferred much more precisely using wildcards if you wish. The following:

```
    *COPY :1.$.TEXT.view* @
```

will copy over only those files in the TEXT directory whose filenames begin with the prefix "view".

Unfortunately *COPY cannot be used to copy files between filing systems. To do this, you need to use *MOVE (as described in BEEBUG Vol.5 No.5), and this will not accept wildcards to imply a multiple file copy - hence this month's Master Series utility.

View Stripping

It is sometimes useful to be able to strip all the special codes from a View text file. For instance, when a View file is to be further edited, or printed out using Wordwise. The following sequence of commands from within the Master Editor will more or less do the trick:

```
    f5 Ctrl-Z/Return
    f5 $$/+++Return
    f5 $/SpaceReturn
    f5 Ctrl-//Return
    f5 +++/$$Return
```

where Ctrl-/ means hold down Ctrl then press "/". You will also need to manually remove View rulers using Delete or Ctrl-Copy, and you may need to manually insert a few carriage returns.

MASTER SERIES

Loading Sideways RAM without a hard break

Mike McNamara shows how to initiate sideways ROM images on the Master without using Ctrl-Break; and Lee Calcraft turns his idea into machine code.

If you have ever tried to use any of the Master's prodigious 64K of sideways RAM for holding ROM images, you will have realised that to "initialise" the image you need to perform a hard Break. This is very inconvenient since it upsets the machine completely, destroying function key definitions, resetting filing systems and halting the progress of any current program.

Acorn themselves soon realised the inconvenience, and have provided the Master Compact with a way to initialise ROM images as they are loaded in. In this article we present a solution to the problem for Master users. It comes in two alternative versions - Basic or machine code - and you can use whichever suits you best, though neither will work with the few ROMs which need to claim workspace on power-up.

THE BASIC SOLUTION

Both solutions revolve around the fact that the Master (like the Model B) holds a table in RAM (&2A1-&2B0) containing status information on each of the machine's 16 possible paged ROMs. There is one byte for each possible ROM slot, and this byte normally holds a copy of byte &8006 of the corresponding ROM. This is the byte in the header of every paged ROM which holds the so-called ROM type. The most common types are:

Service ROM	&82
Language & Service	&C2
Basic	&60

If a ROM is software-unplugged however, the table contains a zero at the corresponding point.

To initialise a ROM image, we just have to make sure that the ROM table contains a value appropriate to the type of occupying ROM. The simplest way to find out these values is to use the following:

```
    P.~(n?&2A1)
```

where n is the ROM socket number. The following program prints out all 16 against the socket number:

```
    10 FOR n=0 TO 15
    20 P.n,~(n?&2A1)
    30 NEXT
```

To initialise a given ROM image, just use:

```
    *SRLOAD filename 8000 n
```

to load it into socket n. Then

```
    ?(&2A1+n)=value
```

where "value" is the corresponding byte (usually &C2 or &82) read from the table at &2A1, as described above.

As an example of the use of this method, table 1 gives the contents of a !BOOT file which initialises two ROM images using this principle.

AUTHOR'S NOTE

Neither method described here is capable of initialising RAM images of ROMs which need to claim workspace. This is the case with filing system ROMs such as the DFS and ADFS, though very few other ROMs for the Master or Compact have such requirements. For the same reason, software-unplugged filing systems still need Ctrl-Break or a power-up after they have been reinstated with *INSERT.

USING MACHINE CODE

We can usefully extend this idea into machine code. The accompanying assembler listing provides a short piece of code which can be *RUN from disc, or CALLed from immediate mode Basic. Its effect is to initialise ROM images in any of the Master's 16 sideways slots. In addition it will implement any new INSERT or UNPLUG conditions, which normally also require a hard Break before they take effect.

To make use of the routine, type it in and save away the assembler listing. When the program is run you will be prompted to press function key f0 to save the assembled machine code, and key f1 to load it back in and run it. The code is saved under the name "init", and all you have to do to run the code once it has been saved is to type:

```
    *INIT     (or *RUN INIT cassette users).
```

Alternatively, if the machine code is already resident, you may use the following from Basic:

```
    CALL &DD00
```

As this suggests, the machine code is assembled and run from the transient program area at &DDØØ. If you wish to alter this, just change the value of R% in line 11Ø.

HOW IT WORKS

The machine code routine uses the same basic idea described above of manipulating bytes held in the table at &2A1. But because it has been written to initialise ROM images automatically, it must also carry out certain checks. The most important is a check that there is indeed a ROM or ROM image present in any given socket. If a bank of sideways RAM contained data of some sort, or just garbage, then copying a byte from it at &8ØØ6 into the table at &2A1 would hang the machine, so we must test for a valid ROM image before doing so.

The test for ROM validity is made in the subroutine "test" at lines 35Ø-47Ø. It is quite portable, and may well prove useful in other applications. The routine is entered with the Y register holding the socket number of the ROM to be tested. If there is a valid ROM at that position, the routine returns with the ROM type in the accumulator. If a valid ROM (or ROM image) is not present, it returns with zero in the accumulator.

A test is also carried out to see whether any of the sockets are software-unplugged. The information for this is held in two bytes of CMOS RAM (bytes 6 and 7), with one bit allocated to each ROM slot. If it is unset, the ROM slot is software-unplugged. If the initialise routine finds an "unplugged" socket, it inserts a zero at the corresponding byte in the ROM table at &2A1.

```
  1Ø REM Program Rom Initialise
  2Ø REM Version B 3.3
  3Ø REM Author  Lee Calcraft
  4Ø REM Beebug  May 1987
  5Ø REM Program subject to copyright
  6Ø :
 1ØØ MODE7
 11Ø R%=&DDØØ
 12Ø OSBYTE=&FFF4:OSRDSC=&FFB9
 13Ø temp=&7Ø
 14Ø FOR pass=Ø TO 1
 15Ø P%=R%
 16Ø [OPT pass*3
 17Ø :
 18Ø \**********Main routine
 19Ø LDA temp:PHA
 2ØØ LDA #161:LDX #7:JSR OSBYTE
 21Ø STY temp:LDY #&F
 22Ø .loop\     Check all 16 Rom slots
 23Ø LDA #Ø:ASL temp:BCC insert
 24Ø JSR test\  Test for Rom image
 25Ø BEQ next
 26Ø .insert\   Put byte in Rom table
 27Ø STA &2A1,Y
 28Ø .next:DEY:BMI finish
 29Ø CPY #7:BNE loop
 3ØØ PHY:LDA #161:LDX #6:JSR OSBYTE
 31Ø STY temp:PLY:BRA loop
 32Ø .finish:PLA:STA temp:RTS
 33Ø :
 34Ø \**********Test for Valid Rom
 35Ø .test:LDA #7:STA &F6
 36Ø LDA #&8Ø:STA &F7
 37Ø JSR OSRDSC:STA &F6
 38Ø JSR OSRDSC:BNE neg
 39Ø INC &F6:JSR OSRDSC
 4ØØ CMP #ASC"(":BNE neg
 41Ø INC &F6:JSR OSRDSC
 42Ø CMP #ASC"C":BNE neg
 43Ø INC &F6:JSR OSRDSC
 44Ø CMP #ASC")":BEQ ok
 45Ø .neg:LDA #Ø:RTS
 46Ø .ok:LDA #6:STA &F6:JSR OSRDSC
 47Ø RTS\       Return with Rom status
 48Ø :
 49Ø ]:NEXT
 5ØØ *KEYØOSCLI("SAVE init "+STR$~R%+"
"+STR$~P%)|M
 51Ø *KEY1*RUN init|M
 52Ø PRINT'"Use key fØ to save code as
init"
 53Ø PRINT'"Use key f1 to load & run in
it"'
```

*SRLOAD SPELL 8ØØØ 6 Q	Load Viewspell into socket 6
*SRLOAD BUFFER 8ØØØ 7 Q	Load the BEEBUG Printer Buffer
CH."FKEYS"	Program the function keys
MODE 131	Select shadow mode 3
*FX2Ø2,5Ø	Switch Shift and Caps lock off
?&2A7=&C2	Activate socket 6
?&2A8=&82	Activate socket 7
*WORD	Enter View
SETUP FJI	Set View to Format, Justify and Insert

Table 1

ACCOUNTS PLUS

Accounts-Plus is the fourth accounting package released by Micro-Aid for the BBC micro. John Wellsman has been trying this one out, and is less than enthusiastic.

Product : Accounts-Plus
Supplier: Micro-Aid
25 Fore Street, Praze,
Camborne, Cornwall TR14 0JX
Tel: (0209) 831274
Price : £58.05 inc. VAT and p & p.

The title of this program is misleading. It is not an accounts program. It is actually a form of cash account with all entries, whether paid or not, being credited or debited to a bank account and a V.A.T. account. It has analysing abilities in that it is possible to separate the payments into pre-determined groups, and it is therefore possible to construct a set of trial balances by noting manually the balances shown by the analysis. It is also possible to produce invoices, etc., in conjunction with a pre-entered mailing list.

On inserting the disc and booting up, a serious defect in the whole package becomes evident. The whole program is written in colour. Now it is perfectly possible to run a business with a monochrome monitor, and in fact most business computers use a monochrome monitor for its higher resolution and clarity. Unfortunately, the way in which colour is used in this package makes it unsuitable for monochrome monitors, and the brightness must be increased to maximum in order to be able to read everything. Even then red is barely visible. This can be really hard on the eyes and I don't suppose that it does the monitor any good either. Even the use of a colour television is not the proper solution as mode 3 is used for some screens, almost unreadable on a TV.

Another annoyance with the program is that every time that it is loaded it has to be told whether double sided or twin disc drives are being used. It surely would be possible to initialise to this once and for all.

The way in which data is entered also leaves a lot to be desired. Batch entering by date is standard, but it is not possible to enter an invoice number or reference. When entering the name of the debtor or creditor, three characters are prefixed to it to act as an analysis code; this is followed by another 'flag' character, either a space if the invoice has been paid, or an asterisk if it has not. Sales and purchase invoices are entered together; the only difference is that you are asked whether it is a 'bank credit' or a 'bank debit'. As this program is obviously aimed at very small businesses, there may well be some confusion over these terms.

The menu displays do nothing to help the user along. Often, the user selects an item by pressing a number, only to find that he is presented with almost the same menu and has to press the same number again. The options on the menu often do little to explain their purpose. And there still seems to be the odd bug. Towards the end of my exploration, I was unable to go further than entering the date. The computer buzzed angrily at me and refused to go any further.

Computer users often find that poor documentation is something to be expected, but in this case the documentation is terrible. The writer obviously knew how the program worked before he wrote it, but also assumed that everyone else did as well. There is no attempt whatever to lead the first user through the programs; for instance, an explanation of the credit control section comes before there is any attempt to explain how to enter data.

The use of colour throughout this program makes it uncomfortable to use, poor documentation and unhelpful menus make it difficult, and the price of £58.Ø5 makes it expensive. On top of all this, it is rather alarming to read in the instructions that the method of calculating V.A.T. used by the program is 'fround upon' (sic) by the Customs and Excise. As you would expect, I cannot recommend this program as a good buy.

BEEBUG WORKSHOP - Pointing The Way (pt 2)

Continuing last month's theme Surac shows how pointers can be used in complex data structures such as family trees. Several useful routines are listed to locate all those long lost relations!

In last month's workshop we looked at the idea of pointers and how they may be represented in Basic. In the examples given, the structure and ordering of information was comparatively simple, but applications not infrequently occur where the data naturally has a much more complex structure. The use of both data and pointer arrays to represent such a situation leads to the formation of a so-called data structure. Data structures are very important to many complex programming situations, and it is usually crucial to get the structure right before commencing any programming (data structures were covered more generally in Vol.4 Nos.4 & 5).

As an example of a more complex form of data structure I have given some thought to the storage of family trees. I have assumed that the basic unit of information is an individual person and that the details to be stored for each person are to be their name, sex, date of birth (dob), and place of birth (pob).

In order to store the relationships between the people in a family tree a number of pointers will be needed. After some experimenting I devised the following:

mother
father
spouse
next elder sibling
next younger sibling
eldest child

Thus we have created a data structure, as shown in figure 1, to store all the relevant information about an individual and his relationship with others. The data and pointers for each person can, in computing terms, be thought of as a single record.

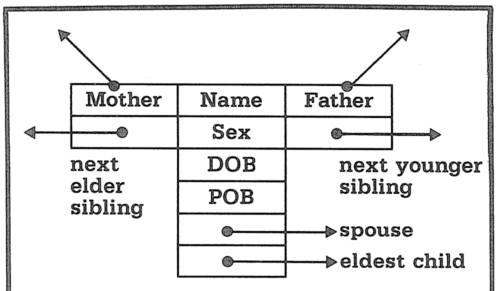


Figure 1

We will assume that the information is to be stored in main memory, in which case an array will be needed for each type of data and pointer. For simplicity, string arrays will be used for the person's details, and real arrays for pointers, though this is not the most efficient use of memory. The same position in each array will relate to the same individual (similar to the final example last month). The names of the arrays will be as already described (or an abbreviation - e.g. 'nes' for 'next elder sibling'). We will now write our first procedure to display the details of an individual 'r':

```
1000 DEF PROCdetails(r)
1010 PRINT"Name: ";name$(r)
1020 PRINT"Sex: ";sex$(r)
1030 PRINT"Date of birth: ";dob(r)
1040 PRINT"Place of birth: ";pob(r)''
1050 ENDPROC
```

Now let us suppose that we want to display the details of the parents of any individual 'r'. A procedure to do this is as follows:

```
1100 DEF PROCparents(r)
1110 IF mother(r) PROCdetails(mother(r))
1120 IF father(r) PROCdetails(father(r))
1130 ENDPROC
```

A zero pointer is assumed to indicate either that no information exists or that the relationship does not exist (e.g. no children), hence the tests in lines 1110 and 1120. Now we can use this to write a procedure to display the details of all the grandparents of an individual:

```
1200 DEF PROCgrandparents(r)
1210 PROCparents(mother(r))
1220 PROCparents(father(r))
1230 ENDPROC
```

If we wanted to locate not just the parents, but the grandparents, great-grandparents and so on going back through the generations, this is an ideal situation for a recursive procedure:

```
1300 DEF PROCancestor(r)
1310 IF mother(r) THEN
              PROCancestor(mother(r))
1320 PROCparents(r)
1320 IF father(r) THEN
              PROCancestor(father(r))
1330 ENDPROC
```

The above procedure is directly akin to the traversal of a binary tree. The three lines 1310, 1320 and 1330 may be in any order, but each order will produce a different traversal of the tree-like structure, and the order in which the details of ancestors are displayed will be different in each case. So far we have looked at procedures starting with any individual and then locating parents and grandparents etc. Let us look now at children and grandchildren, first of all a procedure to display details of all the children of an individual. In this case we use the pointer from an individual to their eldest child (ec), and then the chain of pointers from each child to the next youngest child (sibling) until all children have been located.

```
1400 DEF PROCchildren(r)
1410 IF ec(r) THEN r=ec(r):REPEAT:
     PROCdetails(r):r=nys(r):UNTIL r=0
1420 ENDPROC
```

We can now immediately follow this up with a further procedure to find grandchildren:

```
1500 DEF PROCgrandchildren(r)
1510 IF ec(r) THEN r=ec(r):REPEAT:
     PROCchildren(r):r=nys(r):UNTIL r=0
1520 ENDPROC
```

Again, it is possible to turn the last procedure into a recursive form which will locate an individual's children, their children's children and so on.

```
1600 DEF PROCheirs(r)
1610 IF ec(r) THEN r=ec(r):REPEAT:
     PROCdetails(r):PROCheirs(r):
     r=nys(r):UNTIL r=0
1620 ENDPROC
```

As before, the exact ordering of progeny will be determined by the detailed coding. The examples given here are intended primarily to show how to write procedures to move around a complex database, rather than to satisfy the detailed needs of expert genealogists.

As a final and quite interesting example, consider the problem of finding all the cousins of a particular person. It is necessary to treat cousins on the mother's side separately to those on the father's side, giving a procedure in two mirror halves. On the mother's side for example, find the eldest sibling of the mother, and then proceeding from eldest to youngest sibling, display the details of the children of each sibling in turn, not forgetting to omit those of the mother herself, or you would find that someone was their own cousin!

```
1700 DEF PROCcousins(r):p=0
1710 IF mother(r) THEN p=mother(r):
     IF nes(p) THEN
     REPEAT:p=nes(p):UNTIL nes(p)=0
1720 REPEAT
1730 IF p<> mother(r) THEN
              PROCchildren(p)
1740 p=nys(p)
1750 UNTIL p=0
1810 IF father(r) THEN p=father(r):
     IF nes(p) THEN
     REPEAT:p=nes(p):UNTIL nes(p)=0
1820 REPEAT
1830 IF p<> father(r) THEN
              PROCchildren(p)
1840 p=nys(p)
1850 UNTIL p=0
1860 ENDPROC
```

You may well find it interesting to consider alternative, maybe better ways of writing some of the procedures I have given here. You may also like to try your hand at some different ones as well. How about a procedure to find a person's nephews and nieces, or maybe their

→ 52

Vectored entry into Sideways RAM

Bernard Hill demonstrates how to coax last month's 'key blipper' into sideways RAM.

PAGED ROMS AND VECTORS

Last month we gave the code for a stand-alone keyboard blip routine which gave audible feedback from the computer keyboard as an aid to reducing typing errors. That routine ran from user RAM: in this article, we shall be coaxing it into sideways RAM. This is a more convenient location for regularly used machine code, and additionally permits it to be put into EPROM, from where it is always available at switch-on. Installing this kind of routine into sideways RAM is not as simple as one might at first think, because of its use of the "read character" vector. And it is with the system vectors that we begin this article.

THE VECTOR TABLE

The BBC micro is a very versatile machine indeed. Not least in the hardware which is tailored for expansion, but, equally importantly, in the software. As designed, it has been left with many documented "hooks" onto which program writers can hang their add-ons. Thus, rather than being frozen in the system ROM, the BBC micro holds the addresses of the most commonly used pieces of code ("the vectors") in RAM.

While this sounds like a dangerous thing to do (as RAM can be accidentally changed!), it is in fact an extremely useful feature, allowing us to add to or replace much of the machine's operation. In fact all filing system ROMs (DFS, ROMIT etc) do precisely this: when the machine is first powered up, the operating system writes the addresses of the tape filing system routines for all vectors concerned with files, but when the disc filing system is initialised (whether on Break or explicitly, with *DISC), these values are overwritten with values which that particular filing system ROM requires.

These addresses are located in a table at &2ØØ-235 on all current members of the BBC micro family, and cover many essential operations together with a few spares for the inventive user to hang his own features on. Table 1 covers the most popular, and I have numbered the vectors Ø to 26, so that the address of number Ø is in &2ØØ-2Ø1, number 1 in &2Ø2-2Ø3 etc. As always in 65Ø2 assembler, the low byte is held in the lower address.

Some of the Commonly Used System Vectors
(addresses in hex)

No	Name	Add-ress	Vec-tor	Conts OS1.2	Usage
4	CLIV	FFF7	2Ø8-9	DF89	Star command interpreter
5	BYTEV	FFF4	2ØA-B	E772	OSBYTE handler
6	WORDV	FFF1	2ØC-D	FF2A	OSWORD handler
7	WRCHV	FFEE	2ØE-F	EØA4	Write char
8	RDCHV	FFEØ	21Ø-1	DEC5	Read char
9	FILEV	FFDD	212-3	F27D	File SAVE/LOAD
11	BGETV	FFD7	216-5	F4C9	BGET from a file
12	BPUTV	FFD4	218-9	F529	BPUT to a file
14	FINDV	FFCE	21C-D	F3CA	OPENIN/OUT/UP
16	EVENTV	--	22Ø-1	FFA6	Event handling

TABLE 1

VECTORED CALLS

In a great many machine code programs you will find a need to write to the screen or read from the keyboard. This is performed with a command such as:

```
    JSR OSWRCH
or  JSR &FFEE
```

But what exactly happens when you JSR to the address &FFEE? At this address is found just one instruction: JMP (&2ØE): i.e. a jump to the address which is stored at vector number 7.

In this article we're going to concentrate on vector number 8, called RDCHV, which reads a byte from the current input stream (normally the keyboard). The keyboard blip routine listed last month uses this vector to direct the operating system to a short piece of code which produces a blip every time that a key is pressed.

All we had to do to make this happen was to redirect vector number 8 to point to our piece of code rather than the operating system's code. This was achieved in lines 150 and 160. At the end of the blip routine the operating system was then directed back to the original contents of vector 8 (line 320).

SIDEWAYS ROM IMPLICATIONS

All this is easy and very standard, and routines which do this have been published often before. But if the code to produce the blip resides in sideways ROM or RAM then the situation is rather more complicated, as in addition to giving the address of the routine, we also need to stipulate which ROM it resides in.

The designers of the BBC operating system have determined that in the case of ROM-resident routines we must operate a similar but quite different system:

1. Instead of the vector holding the start address of the new routine, it must hold the value &FF00+3*V, where V is the vector number.

2. The real address of the routine, together with its ROM number, must now be placed in a second table at a particular place in memory. For OS 1.2 this table is at &D9F.

3. The address within the table for the data for any given vector number V is given by:
   table=&D9F + 3*V
   And the three-byte contents of the slot are as follows:
   table?0=(routine's address) MOD 256
   table?1=(routine's address) DIV 256
   table?2=ROM number

What actually happens is that the operating system has arranged that some code is present at &FF00+3*V which consults the table at &D9F to find out which address to jump to, and which ROM to use.

This information is used in the accompanying program. It contains the full code for a sideways ROM header and command interpreter, as well as the key blip routine from last month. To get this working, first type in the listing. Then save the program away before running it.

When the program is run it will automatically save the assembled machine code under the name "ROM". Once you have done this you just need to reload this into sideways RAM. On an ATPL board or similar, use:

```
*LOAD ROM 8000
```

Users of other makes of sideways board should consult their manual on loading ROM images. Once the image is loaded, press Break to initialise. On a Master, use:

```
*SRLOAD ROM 8000 n
```

where n is the number (4-7) of the destination RAM socket. Then initialise the ROM image by pressing Ctrl-Break, or use the Master Initialise routine published in this issue. On a Compact use:

```
*SRLOADI ROM 8000 n
```

which both loads and initialises the image.

This gives you two new star commands:
*BLIP
and *NOBLIP
Their effects should be obvious!

The format of the ROM header code and the command interpreter code is exactly as given in the series on Sideways RAM in BEEBUG Vol.5 Nos. 2-4. To save space, the command line interpreter has been compacted and renumbered, but the number sequence of all sections of the code remains unaltered. If you already have these procedures, then you only need lines 100-170 and from 18000 onwards.

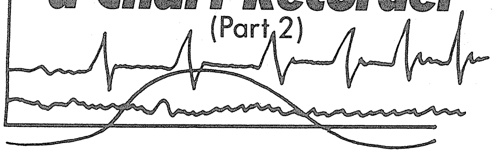
This month's listing should also prove useful as a further, and relatively simple, example of the way in which new star commands can be inserted into the format developed in the earlier series. By following the ideas set out in the series, it should be an easy matter to append the blip routine to other procedures and utilities already developed in ROM image format, rather than using up a whole ROM just for a single command.

Application Note

The key blip routines given both this month and last should be assembled on the system on which they are to be used. If the system is changed, such as by adding a Spell-Master ROM, which alters system vectors, or by upgrading to a Master, etc, you will need to re-assemble the code on the new system.

```
   1Ø REM Program Keyboard Blip ROM
   2Ø REM Version B 1.ØA
   3Ø REM Author  Bernard Hill
   4Ø REM Beebug  May 1987
   5Ø REM Program subject to copyright
   6Ø :
  1ØØ MODE 7:HIMEM=&5CØØ
  11Ø P%=&8ØØØ:O%=&5CØØ
  12Ø PROCromhead("Keyboard Blip ROM","V
1.Ø","BEEBUG, 1987")
  13Ø CLEAR
  14Ø PROCbuzz
  15Ø PROCendrom
  16Ø *SAVE ROM 5CØØ +2ØØØ 8ØØØ 8ØØØ
  17Ø END
  18Ø :
 1ØØØ DEFPROCromhead(name$,ver$,owner$)
 1Ø1Ø Q%=P% : R%=O%
 1Ø2Ø FOR opt=4 TO 7 STEP 3
 1Ø3Ø P%=Q% : O%=R%
 1Ø4Ø [ OPT opt
 1Ø5Ø EQUB Ø
 1Ø6Ø EQUB Ø
 1Ø7Ø EQUB Ø          \ not a language rom
 1Ø8Ø JMP serve       \ service entry pt
 1Ø9Ø EQUB &82        \ service rom
 11ØØ EQUB copyrite MOD 256
 111Ø EQUB Ø
 112Ø .title     EQUS name$
 113Ø .verstr    EQUS CHR$Ø+ver$
 114Ø .copyrite EQUS CHR$Ø+"(C) "+owner$
+CHR$Ø
 115Ø .serve
 116Ø ]
 117Ø NEXT opt
 118Ø ENDPROC
 119Ø
 7ØØØ DEFPROCcommandhandler(cmdtable)
 7Ø1Ø Q%=P% : R%=O%
 7Ø2Ø FOR opt=4 TO 7 STEP 3
 7Ø3Ø P%=Q% : O%=R%
 7Ø4Ø [ OPT opt
 7Ø5Ø CMP #4:BEQ starcmd:JMP notforme
 7Ø6Ø .starcmd
 7Ø7Ø LDA &A8:PHA:TXA:PHA
 7Ø8Ø TYA:PHA:LDX #&FF:STX &A8:INX
 7Ø9Ø .nextcmd
 71ØØ PLA:TAY:PHA:INC &A8:DEX:DEY
 711Ø .nextchar:INX:INY:LDA (&F2),Y
 712Ø CMP #ASC"a":BCC notlc
 713Ø CMP #ASC"z"+1:BCS notlc:AND #&DF
 714Ø .notlc
 715Ø EOR cmdtable,X :BEQ nextchar
 716Ø BPL nomatch:LDA (&F2),Y
 717Ø BEQ gotcmd:CMP #13:BEQ gotcmd
 718Ø CMP #32:BEQ gotcmd
 719Ø .nomatch
 72ØØ CPX #Ø:BEQ overcmd:LDA (&F2),Y
 721Ø CMP #ASC".":BEQ dotmatch
 722Ø .overcmd:LDA cmdtable,X:BMI eocm
 723Ø INX:JMP overcmd
 724Ø .eocm:INX:INX:LDA cmdtable,X
 725Ø BNE nextcmd:PLA:TAY:PLA:TAX
 726Ø PLA:STA &A8:LDA #4:JMP notforme
 727Ø .dotmatch:INX:LDA cmdtable,X
 728Ø BPL dotmatch:.gotcmd
 729Ø LDY &A8:LDA cmdtable,X
 73ØØ PHA:LDA cmdtable+1,X:PHA
 731Ø LDA cmdtable+1,X PHA
 732Ø RTS:.notforme
 733Ø ]:NEXT opt:ENDPROC
 734Ø :
 9ØØØ DEFPROCendrom
 9Ø1Ø [ OPT 7:RTS:]:ENDPROC
 9Ø2Ø :
18ØØØ DEFPROCbuzz
18Ø1Ø Q%=P% : R%=O% : V=8 :REM RDCHV
18Ø2Ø FOR opt=4 TO 7 STEP 3
18Ø3Ø P%=Q% : O%=R%
18Ø4Ø [ OPT opt
18Ø5Ø JMP cmdparse
18Ø6Ø .vectors:EQUS "BLIP" : EQUW (buzz-
1) MOD 256*256 + (buzz-1) DIV 256
18Ø7Ø EQUS "NOBLIP" : EQUW (nobuzz-1) MO
D256*256 + (nobuzz-1) DIV 256:EQUB Ø
18Ø8Ø :
18Ø9Ø .buzz  \ change vectors
181ØØ LDA &A9 : PHA \ need this for now
1811Ø LDA #3*V :STA &2ØØ+2*V
1812Ø LDA #&FF  :STA &2ØØ+2*V+1
1813Ø LDA #&A8  : LDX #Ø : LDY #&FF
1814Ø JSR &FFF4 \ get extended vec table
1815Ø STX &A8 : STY &A9 : LDY #3*V
1816Ø LDA #new MOD 256:STA (&A8),Y : INY
1817Ø LDA #new DIV 256:STA (&A8),Y : INY
1818Ø LDA &F4 : STA (&A8),Y \ and rom no
1819Ø PLA : STA &A9 \ restore and return
182ØØ PLA : TAY : PLA : TAX \ ..as usual
1821Ø PLA : STA &A8 : LDA #Ø : RTS
1822Ø :
1823Ø .new:PHP : PHA : TXA : PHA
1824Ø TYA : PHA : JSR flush
1825Ø LDA #7 : LDX #parms MOD 256
1826Ø LDY #parms DIV 256 : JSR &FFF1
1827Ø PLA : TAY : PLA : TAX : PLA : PLP
1828Ø .jmp:JMP !(&2ØØ+2*V) AND &FFFF
1829Ø .flush:LDA #21:LDX #4:LDY #Ø
183ØØ JSR &FFF4:RTS
1831Ø .parms
1832Ø EQUW Ø : EQUW -9 : EQUW 2 : EQUW 1
1833Ø .nobuzz
1834Ø LDA jmp+1 : STA &2ØØ+2*V
1835Ø LDA jmp+2 : STA &2ØØ+2*V+1
1836Ø PLA : TAY : PLA : TAX
1837Ø PLA : STA &A8 : LDA #Ø : RTS
1838Ø :
1839Ø .cmdparse : ]
184ØØ NEXT opt
1841Ø PROCcommandhandler(vectors)
1842Ø ENDPROC
```

Turn your BEEB into a Chart Recorder (Part 2)



David Peckett adds the finishing touches to last month's program, and completes the process of turning your Beeb into a four-channel chart recorder.

Last month's article contained the core listing for a program to convert a Beeb and a printer into a chart recorder. This month's listing completes the program by adding the routines to reprint data, to save data to disc and to use 'star' commands.

These additions overwrite the dummy procedures at the end of last month's program. Either load part one from last month and type in the additions, or enter these separately and then merge them with the original program by, for instance, using the *MERGE facility in the BEEBUG Toolkit ROM. You could also use the Part Merge/Save program from Vol.5 No.6. Do make sure when adding part two that you include line 4220 as this replaces a dummy procedure call in part one.

THE NEW FACILITIES

Once you have added the extra code, the top level menu of the program will look just the same as before, but all the options will now be operational.

Pressing 'P' allows you to plot any data which may already have been saved in memory by running the program. Because, by this stage, the system knows the full range of data which has been read, it allows you to set a real scale to the vertical axis. You can enter the real world values (e.g. 100 psi) which correspond to the maximum (65535) and minimum (0) values which the Beeb's ADC can supply. You can also decide whether to expand the vertical scaling of the chart so that the graphs fill it from top to bottom, or to keep it as it was originally. Furthermore, you can choose to print just part of the total data which was recorded.

The 'O' (for 'Old') option works just like 'P', except that it takes its data from a disc file rather than from memory.

With 'S', you can save data which is in memory, whether or not you saved it while you were reading the data. Finally, pressing '*' gives you access to the Beeb's 'star' commands; you can run any of them, but the common ones which would crash the program ask you to confirm before they proceed.

PLOTTING MATHEMATICAL FUNCTIONS

As with the earlier oscilloscope system, the program can be adapted to print mathematical functions. To do this you need, at the least, to amend line 2650 to define the function for each channel. For instance:

```
2650 V%=5000+10000*I%+10000*
        SIN(I%*N%/10)
```

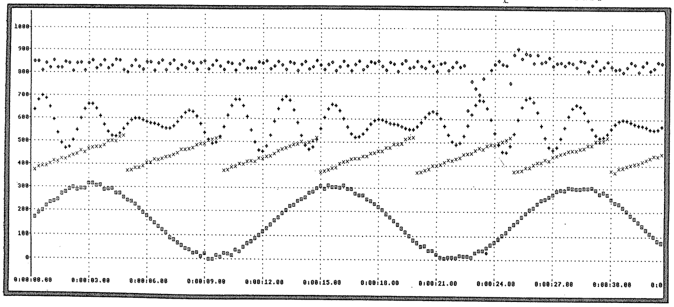
will plot a series of sine curves.

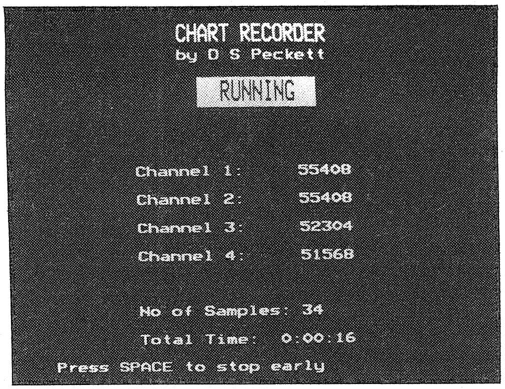
You can make things as complicated as you like, adding extra lines if necessary, and it is not difficult to produce some very complex plots.

USING NON-EPSON PRINTERS

Although the system is written for use with Epson and Epson-compatible printers, the following few notes should help you to adapt it for printers which use other command sequences.

Line Spacing. The printer normally prints at 6 lines per inch. An Epson printer can print, at a single pass, a line of graphics 1/8" high; the line spacing must therefore be set to 8 lines per inch.




```
CHART RECORDER
by D S Peckett

RUNNING



Channel 1:      55408
Channel 2:      55408
Channel 3:      52304
Channel 4:      51568


No of Samples: 34
Total Time:  0:00:16
Press SPACE to stop early
```

The command to drive an Epson printer to set printer spacing to n/216 inches is:

VDU 2,1,27,1,51,1,n,3

The '2' selects the printer, the '1's send the next character to the printer only, while the '3' merely de-selects the printer. Thus line 278Ø sets spacing to 24/216" (or 1/8"), while lines 148Ø and 479Ø reset the printer to print 6 lines per inch (normal spacing).

Driving Graphics. Graphics are printed as 8 parallel lines of dots or spaces across the page. The dots can be set to several horizontal spacings and the program uses the basic spacing of 48Ø groups of dots across the page for draft quality, and 96Ø groups (each group printed twice) for 'quality' printing.

To print 'n' groups of 8 dots in draft quality, an Epson requires the sequence:

VDU 1,27,1,ASC("K"),1,n MOD 256,
1,n DIV 256

while:

VDU 1,27,1,ASC("L"),1,n MOD 256,
1,n DIV 256

does the same for 'quality' printing. Line 339Ø does this job for you, using the value of 'QPlot' to select between the modes.

PROGRAM NOTES

Here are some brief notes about the workings of the main procedures and functions used in the program.

PROCInit: The variable 'Free' identifies the maximum amount of RAM available to store data, regardless of the model of the computer or its configuration.

PROCMenu: Sets up the top-level menu and returns the selected option.

PROCRead: Controls the reading, plotting and real-time saving of data from the ADC.

PROCParams: Reads in and validates the parameters which describe how much data is to be read, and sets up the main flags which control the program.

PROCRun: Times the reading-in of data and displays the raw values as they are read.

PROCHandleData: Actually reads in the data, saves it to RAM and/or disc, and plots it.

PROCHeader: Sets the printer to print at 8 lines per inch. It then prints the graduations at the left-hand side of the chart. It also sets up the vertical scaling factor for the plot.

PROCGraphIt: Positions the symbols representing the input data, calculates and, if necessary, sets the time markings at the bottom of the chart.

PROCGetFont: Uses an OSWORD call to read a character's font so that it can be plotted sideways.

PROCPrtLine: Uses the printer's graphics facilities to print 48Ø (QPlot=FALSE) or 96Ø (QPlot=TRUE) sets of 8 dots across the page to give a vertical line of the plotter's output.

PROCPosChar: Positions character 'ch$', printed sideways, at position 'pos' across the printout.

FNGetch(str$): Waits for one of the keys defined in 'str$' to be pressed.

FNTime(tim): Converts the value in 'tim' into an 'hr:min:sec.dec' string.

FNRunTime: Reads the run time entered in 'hr:min:sec' and converts it to seconds.

PROCSave: Saves the data in RAM to a disc/cassette file.

PROCPlot: Sets up to replot the data already in RAM.

PROCOld: Sets up to plot data from a disc file.

PROCRePlot: Reads existing data from RAM or disc and plots it.

PROCrange: Identifies the y-axis graduations for the plot, and whether the plot's vertical scale is to be expanded to fill the paper.

PROCLimits: Reads the real-world values to be plotted and sets up system to display data correctly.

PROCStar: Gives access to '*' commands, trapping dangerous ones.

PROCOSCLI: Gives Basic I access to OS commands. The procedure is redundant in later versions of Basics (in which case line 5140 should be:

```
    5140 IF OK THEN PRINT':OSCLI(star$)
```

```
 4220 :
 4230 DEF PROCOpenInFile
 4240 PRINT TAB(5,8);:INPUT "Name of dis
c File? "FilName$
 4250 PRINT TAB(5,18) "Insert disc,"CHR$
136"SPACE"CHR$137"when ready ";
 4260 REPEAT UNTIL INKEY-99
 4270 F%=OPENUP(FilName$)
 4280 INPUT#F%,N%,Tsamp,Tmon,hival,loval
,IP(1),IP(2),IP(3),IP(4)
 4290 PTR#F%=60
 4300 ENDPROC
 4310 :
 4320 DEF PROCSave
 4330 LOCAL I%
 4340 PROCTitle("SAVE EXISTING DATA")
 4350 IF NOT RAM THEN ENDPROC
 4360 PROCOpenOutFile
 4370 FOR I%=0 TO P%-2:BPUT#F%,DBuff?I%:
NEXT
 4380 PROCCloseOutFile
 4390 ENDPROC
 4400 :
 4410 DEF PROCPlot
 4420 LOCAL FromDisk,I%
 4430 PROCTitle("PLOT DATA FROM RAM")
 4440 FromDisk=FALSE
 4450 PROCRePlot
 4460 ENDPROC
 4470 :
 4480 DEF PROCOld
 4490 LOCAL FromDisk,N%,Tsamp,Tmon,hival
,loval,IP(1),IP(2),IP(3),IP(4)
 4500 PROCTitle("PLOT DATA FROM DISC")
 4510 FromDisk=TRUE
 4520 PROCOpenInFile
 4530 PROCTitle("PLOT DATA FROM DISC")
 4540 PROCRePlot
 4550 ENDPROC
 4560 :
 4570 DEF PROCRePlot
 4580 LOCAL I%,J%,P%,All%,st%,fin%
 4590 PROCLimits
 4600 PRINT TAB(5,10) "There are ";N%;"
samples." TAB(5,11) "Print them all? ";
 4610 All%=FNYes
 4620 IF NOT All% THEN PROCRange ELSE st
%=1:fin%=N%
 4630 PRINT TAB(5,15) "Draft or Quality
Print (D/Q)? ";
 4640 QPlot=(FNGetch("DQ")="Q")
 4650 PRINT TAB(11,18) CHR$136 CHR$130 C
HR$157 CHR$135 "RUNNING  " CHR$156
 4660 PROCHeader(loval,hival,loval*(RWhi
-RWlo)/65536+RWlo,hival*(RWhi-RWlo)/6553
6+RWlo,st%-1)
 4670 PRINT TAB(0,20) "Press"CHR$136"SPA
CE"CHR$137"to stop early";
 4680 I%=st%
 4690 P%=2*(1-st%)*(IP(1)+IP(2)+IP(3)+IP
(4)):IF FromDisk THEN PTR#F%=60+P%
 4700 REPEAT
 4710 FOR J%=1 TO 4
 4720 IF IP(J%) AND FromDisk THEN Inval(
J%)=BGET#F%+BGET#F%*256
 4730 IF IP(J%) AND NOT FromDisk THEN In
val(J%)=(DBuff!P% AND &FFFF):P%=P%+2
 4740 NEXT
 4750 PROCGraphIt(loval)
 4760 I%=I%+1
 4770 UNTIL I%=fin% OR INKEY-99
 4780 IF FromDisk THEN CLOSE#F%
 4790 VDU 2,1,12,1,27,1,51,32,3
 4800 ENDPROC
 4810 :
 4820 DEF PROCRange
 4830 LOCAL OK
 4840 REPEAT
 4850 PRINT TAB(5,12) SPC30 TAB(5,12) "F
irst sample (1-";N%-2;")?";
 4860 INPUT " "st%
 4870 OK=st%>0 AND st%<N%-1
 4880 IF NOT OK THEN VDU7
 4890 UNTIL OK
 4900 REPEAT
 4910 PRINT TAB(5,13) SPC30 TAB(5,13) "L
ast sample (";st%+2;"-";N%;")?";
 4920 INPUT " "fin%
 4930 OK=fin%>(st%+1) AND fin%<=N%
 4940 IF NOT OK THEN VDU7
 4950 UNTIL OK
 4960 ENDPROC
 4970 :
 4980 DEF PROCLimits
 4990 PRINT TAB(0,4) "Recorded data valu
es from"';loval;" to ";hival;" in a poss
ible"''"range of 0 to 65535"
 5000 INPUT "Real-world value equiv. to
0? "RWlo
```

```
 5010 INPUT "Real-world value equiv. to
65535? "RWhi
 5020 PRINT "Expand the vertical axis? "
;
 5030 IF NOT FNYes THEN loval=0:hival=65
536
 5040 ENDPROC
 5050 :
 5060 DEF PROCStar
 5070 PROCTitle("STAR COMMANDS")
 5080 PRINT TAB(5,8);:INPUT "*"star$
 5090 Lstar$=LEFT$(star$,2)
 5100 OK=(Lstar$<>"CO" AND Lstar$<>"BA")
 5110 OK=(OK AND Lstar$<>"L." AND Lstar$
<>"LO")
 5120 OK=(OK AND Lstar$<>"R." AND Lstar$
<>"RU")
 5130 IF NOT OK THEN VDU7:PRINT '''"THAT
MAY ERASE THE PROGRAM!"''"Do you wish to
continue (Y/N)? ";:OK=FNYes
 5140 IF OK THEN PRINT:PRINT:PROCOSCLI(s
tar$)
 5150 IF VPOS>18 THEN PRINT:PRINT
 5160 PRINT TAB(0,20) "Press"CHR$136"SPA
CE"CHR$137"for main Menu ";
 5170 REPEAT UNTIL INKEY-99
 5180 ENDPROC
 5190 :
 5200 DEF PROCOSCLI(str$)
 5210 LOCAL X%,Y%
 5220 $osclibuf=str$
 5230 X%=osclibuf MOD 256
 5240 Y%=osclibuf DIV 256
 5250 CALL &FFF7
 5260 ENDPROC
```

THE RECURSIVE UNIVERSE

This fascinating book by William Poundstone provides an in-depth exploration of the world of Life, the game devised by John Conway. John Yale, a keen 'Lifer', steers a course through a mass of information.

'The Recursive Universe' by William Poundstone published by Oxford University Press at £5.95.

Neither the title nor the subtitle 'Cosmic Complexity and the limits of Scientific Knowledge' reveal the main interest of this book for computer owners. About half of the book's 250 pages are devoted to the 'Game of Life' (see BEEBUG Vol.1 No.10). This game developed a cult following in the 60s and 70s, and many versions were implemented on computers large and small.

All of the well known 'Life' objects are here from blocks and blinkers to glider guns and puffer trains, with much biographical information about their discovery. Many new objects are also to be found including one pattern of 13 gliders which collide to form a glider gun!

The book explores the possibility that there are "living" objects in the Life universe which can reproduce themselves, react with the environment, evolve into more complex "organisms", and even become intelligent. Along the way, the book has chapters on Maxwell's Demon, Information and Structure, the Big Bang and Heat Death, and Self Reproducing Machines.

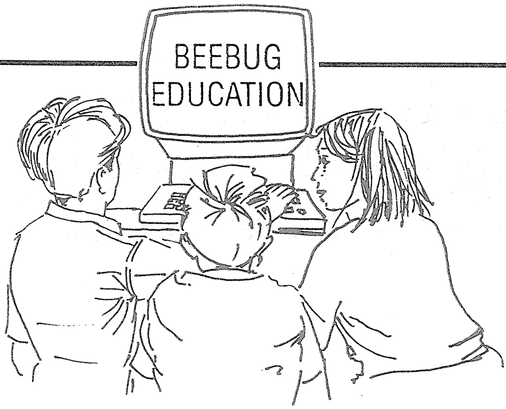
The book contains two simple programs for exploring the Life universe, one in Basic and the other in Assembler suitable for the IBM PC. BEEBUG members would do better to get the program from Vol.1 No.10, or a more sophisticated commercial program available from Software Production Associates (0926 22959).

This book is essential reading for any 'Life' fans. For those interested in the origins and fate of the universe the book is a good read but rather expensive for a standard size paperback.

←49 WORKSHOP

brothers and sisters. It is also quite interesting to consider what would be the most useful set of primary procedures which would allow most other requirements to be built up from these primitives. It is a fascinating area, and one which can prove very interesting to explore.

The magazine cassette/disc for this issue contains a complete demonstration of this month's procedures, including a mini-database of 32 people. You could even substitute the members of your own family if you so wished.



Logotron has come a long way since their version of the Logo language appeared in 1984. Their Logo has become pre-eminent in education, but Logotron has much more to offer users than just a language, as Mark Sealey reports.

Since BEEBUG first looked at the Logo language for the BBC micro (Vol.3 No.1Ø), one version, Logotron's, has virtually become a market leader in what is arguably the most important educational use of the micro. It is certainly the one whose advocates and enthusiasts are most vocal. The MESU (Micro-electronic Education Support Unit) has made the Logotron chip standard in machines that it is issuing, and this version of Logo is part of the software bundled by Acorn (who had their own implementation) with the Master Compact.

This edition of BEEBUG Education is devoted to an an overview of the new additions to the Logotron range. By using Sprites, Music and Control Logo, and the various extensions, the work that pupils of all ages can tackle on the BBC micro, Master and Compact takes them into exciting, educationally significant and technically advanced areas; non-educational users, too, will find Control Logo an accessible introduction to robotics.

CONTROL LOGO

Software to control peripherals attached to the Acorn family of computers has been available in many forms for a long time. So my criteria in evaluating these items have been these:

1) What is the range of commands and effects on offer? Could an experienced user want more or different?

2) How easy and accessible are they to young users?

3) How readable and informative is the documentation that goes with them?

4) How homogeneous a set of facilities do they make?

Logotron's Control Logo scores straight away. Once set up - by loading an extension from disc - the environment in which you are working will consistently follow certain rules. Here, the user cannot but appreciate that the computer is a controller of input and output. And it is clearly important for children to see WHERE, HOW and WHY robot arms, lights and thermostats, say, relate to their own programming. Almost a quarter of the 38 primitives added by Control Logo contain either the word 'IN' or 'OUT'.

For example, the following simple procedure tests continually to see if bit 7 of the user port (the default) is set. If so, it turns on bits Ø, 1 and 2 of the printer (the default) port. This could well be connected to warning lights in a model level-crossing with the user port line connected to a sensor under the rails that is activated by a train:

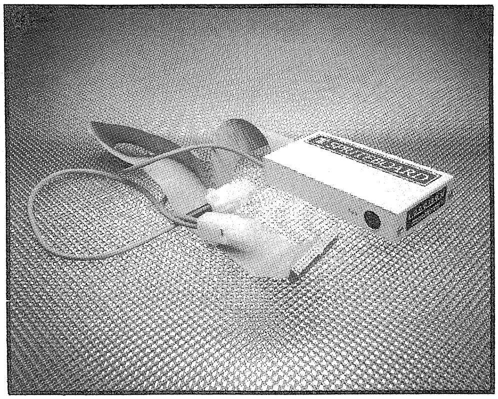
```
    TO WARN
    .SETOUTPORT PRINTER; Printer port set
to current output port
    .SETINPORT USER; User Port set to
current input port
    IF IN? 7 [TURNON [Ø]]
    IF NOT IN? 7 [TURNOFF [Ø]]
    WARN
    END
```

All operations on ports (User, Printer, Analogue and 1MHz) can be specified in bits, in hex, or in decimal. Logical operators (NOT, AND, OR etc) are available but must be in prefix form. If you wanted to begin an exploration of control on the BBC micro but were not familiar with Logo, I think it would be hard to better the use of this package. A chapter in the documentation (definitely not aimed at younger children: but well written and with lots of good ideas) details some sources and uses of hardware.

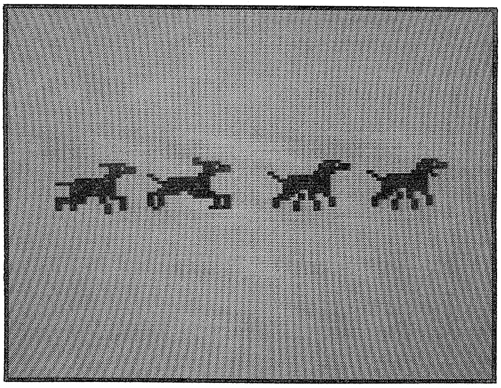
SPRITES AND EXTENSIONS

It is nothing new to say that turtle graphics on their own are overworked. The best implementations of Logo now go much further. Since so many children are used to drawing on a screen, to be able to explore mathematical and scientific worlds is a must. The behaviour of speed and velocity is one such. How do acceleration and deceleration operate? Which mathematical rules determine when two turtles travelling at different rates will collide? For work of this nature, sprites are essential. Logotron's Sprites is exemplary. Not only can you design your own, have 32 of them all moving at the same time, but you can control them and their colours and shapes individually.

The demonstration which comes with Sprite Logo is an inspiration, and so effective, that many children see animation first and control only second. The excellent documentation gives examples of how to write your own games. The primitives (for example, those to detect collisions etc) are easy to use and versatile. The Sprite board must be connected directly into your monitor because it uses a separate video control chip (Texas Instrument's TMS 9918/9929) and is rather expensive. As a first add-on, though, it is well worth it.



The Extension Disc has 8 separate modules. Space does not permit a detailed descripton, but attention must be drawn at least to integral dumps for Epson compatible printers, turtle drivers (Jessop and Valiant) as well as a set of primitives to extend operations on lists and allow properties to be defined. I regret that no attempt has been made to render the debugging process more flexible. It is to be hoped that Logotron will admit the need children have to keep tabs on every step of their procedures.



MUSIC LOGO

By now I hope the point will have been accepted that Logo is NOT chiefly about simple sequences of graphics commands but about a programming environment that actually encourages explorative programming. Music Logo comprises 50 additional primitives to control the pitch, duration, tempo and form (via Envelope) of the BBC's sound chip.

But the point not to be missed is that once you start using words like BREVE, CROTCHET and A# or Bb, which all fit so neatly into the familiar Logo landscape, this version's admitted lack of sophistication ceases to matter. Children of any age would certainly get a lot from turning the Beeb's keyboard into a simple

```
    TO PLAY.NOTE
    MAKE "durations [BREVE CROTCHET MINIM
SEMIBREVE SEMIQUAVER]
    MAKE "chance RANDOM 4
    MAKE "note.value ITEM (:chance+1)
:durations
    MAKE "pitches [C D E F G A B]
    MAKE "size RANDOM 6
    MAKE "pitch.value ITEM (:size+1)
:pitches
    PR SE :note.value :pitch.value
    RUN SE :note.value :pitch.value
    DISPLAY.NOTE (:chance+1) (:size+1)
    PLAY.NOTE
    END

    TO DISPLAY.NOTE "col "size
    PU RT RANDOM :size*100 FD :size*100 PD
SETPC :col
    MAKE "size :size/5
    VDU SE [25 157 1 1 1] :size
    END
```

musical instrument FOR THEMSELVES, or being able to see their notes have visual effects. Imagine drawing the results of skeletal procedures like those shown.

The environment in which you are almost always working is that of the list. Lists are essential to Logo and often ill-understood and ill-used. Logotron Music Logo is an excellent introduction to lists in their own right, as each element is immediate (as with the note example just now). Look too at the use of the extended VDU parameters introduced on the Master and Compact, and via the Graphics Extension ROM elsewhere. On top of all this, octaves, tempi and durations can all be set and changed, and there are example programs in the manual for saving envelopes that you have defined yourself. Again, I can recommend this new software very highly. As with the other items this month, it definitely runs on the Econet but not with Second Processors nor on the Electron.

Lastly, apart from saying that I hope soon to return to Logotron's associated software, mention must be made of a Logo tool which has nothing to do with Logotron itself. Mike Blamires (author of Muddles: which I shall also review soon) has written Logotext which allows you to produce Teletext-like screens in Logo. Commands such as:

```
    PRINT BIG IN BLUE ON WHITE
```

need no explanation. As with the Logotron Pendown Toolkit and Concept Keyboard Editor, Logotext must be seen as putting computer power into the hands of both very young children, and adults and kids with special needs alike.

Special needs education with the Acorn family is a topic to which I shall return in the near future. Meanwhile please do drop me a line at the Editorial address if you are keen to have particular areas of educational computing examined and discussed in these pages.

Logotron Prices

Logotron Logo	**£60.00**
Logotron Extension Disc and Manual	**£15.00**
Control Logo	**£15.00**
Music Logo and Manual	**£15.00**
Sprite Logo	**£149.00**

VAT and carriage not included.

Logotron Ltd.
Dales Brewery, Gwydir Street,
Cambridge CB1 2LJ
Tel: (0223) 323656.

Mike Blamires
Redbridge SEMERC,
Dane Centre, Melbourne Road,
Ilford IG1 4HT.
Tel: 01-478 6363.

←15

freaks there are ample details and examples of how to manipulate the Eureka's memory map, but these are not for the faint-hearted.

The Eureka board is limited in that it will only work with certain versions of a limited range of software, and even then there may be further restrictions on use. Hopefully, the range of software which will work on the Eureka board will be expanded in the future (the View family and Computer Concepts' Inter series are apparently being adapted at present).

The extreme slowness of loading and saving programs/data can be annoying. However, when you consider how much memory is now at your disposal it is arguably a small price to pay.

Despite the limitations of the Eureka board, it is well worth having in your machine. The Eureka board has to be the ultimate memory expansion board so far produced for the Beeb, but at a price, and potential users may well wish to consider a 6502 second processor as a worthwhile alternative which also offers more speed from a proven product.

BEEBUG ON MICRONET

Prestel has now introduced keyword searching, so there is no longer any need for you to remember long page numbers. We have been allocated two keywords; *BEEBUG# will take you directly to the BEEBUG front page, while *DATABUS# will take you directly to our on-line magazine. And keyword searching can now be used for the whole of Prestel.

POSTBAG

POSTBAG

Discs in the Deep Freeze

What please does CRC ERROR mean, and what can cause the error? Faced with the above, I was able to retrieve 10 letters (text files) saved to disc by using the following method.

Place the disc in a large padded envelope and put both in the deep freeze for 15 minutes. Using the envelope to stop the disc warming up in transit to your Beeb, quickly insert the disc in a drive, catalogue the disc, and still at top speed load any file into memory and save to a new disc.

With luck there is still time to recover a second file; once I achieved three. Now return the disc in the envelope to the deep freeze and repeat the process, ensuring the disc is really cold (less than +5 degrees fahrenheit).

I suspect a shrinkage of disc material overcomes the CRC fault, but why?

H.Oak-Rhind

CRC refers to 'cyclic redundancy check' and is a technique used to check on the validity of data stored on disc. The error can arise in various ways, from trying to read 40/80 track *discs in 80/40track drives* through to total corruption of data (for whatever reason). The method for retrieving data is intriguing but there is no guarantee that it will work in every case, if at all. If a disc is corrupted, we would suggest a first approach is to use a file recovery program (e.g. BEEBUG Vol.4 No.8) before resorting to the deep freeze.

Puzzled

The new Spellmaster ROM would seem to be a major improvement on previous spell-check programs. I am the senior crossword compiler for a national daily newspaper, and possibly the only professional compiler in the country who uses a Beeb for the entire composition of crosswords from pattern formatting to final edit. My own program neither chooses words for the pattern, nor does it tailor acceptable cryptic clues, but it does do virtually everything else, and has saved me many hundreds of hours over the past five years.

Existing spell-check pro grams have proved of no real use in professional compiling, but Spellmaster looks promising.

Douglas St P. Barnard

In reviewing Spellmaster we have tested the *CHECK and *CROSSWORD commands, and found them to work extremely well for the composition of crosswords.

Summing up Wordwise

My work often involves typing estimates etc, and subsequently having to check the list of figures with a pocket calculator. The Wordwise Plus program below was borne out of necessity, and is useful for totalling the sums of money which are contained in a Wordwise file such as an estimate or invoice.

Load the program into segment 0. Use the main text area for entering your invoice, estimate, etc., preceeding each money sum with a pound sign. No further text should be entered on such lines. Save the file, and press Shift-f0. The program removes all decimal points, and after a pause types the total sum at the bottom of the text. Note this, reload the original file and insert the total.

As the method uses only integer arithmetic, the total must not exceed £655.35.

```
SELECT TEXT
REPEAT
CURSOR TOP
REPLACE"."""
UNTIL EOT
CURSOR TOP
B%=0
REPEAT
FIND "£"
CURSOR RIGHT
.stepright
IF GCT$=" " THEN GOTO
   stepright
CURSOR LEFT
A$=GLT$
A%=VAL(A$)
B%=B%+A%
UNTIL EOT
TYPE"Total= £"
TYPE STR$B%
CURSOR LEFT2
TYPE"."
IF(?&8F AND 1)=0 THEN
   *FX138,0,0,27
END
```

Raymond Ridgewell

HINTS and tips

More on Shifting Case

The hint 'Shifting Case' in BEEBUG Vol.5 No.8, whilst correct, does not describe the full range of *FX2Ø2 options. A fuller list is printed below:

*FX2Ø2	Shift Lock	Caps Lock	Shift Enable
Ø	On	On	No
16	On	Off	No
32	Off	On	No
48	Off	Off	No
128	On	On	Yes
144	On	Off	Yes
166	Off	On	Yes

P. Vincent

BBC Soft Teletext Bug

Under certain conditions, pressing Escape while using the Advanced Teletext System ROM can generate error code 27 (which is the ASCII code for Escape), rather than the true Escape error code of 17. The Escape action itself works normally, however. Try the short program below:

```
 1Ø ON ERROR GOTO 1ØØ
 2Ø *TTXON
 3Ø *PAGE 999
 4Ø *TRANSFER 3ØØØ
 5Ø END
1ØØ *TTXOFF
11Ø REPORT:PRINT" at line
    ";ERL'ERR
```

For those with sideways RAM, the cure is quite simple. Download the ROM image into main memory and search through the code until you find the word "Escape" (using a machine code monitor such as Exmon). Alter the byte which immediately precedes it from &1B (27) to &11 (17) and save this new image to disc or tape. You can now run the revised ROM image in sideways RAM.

Richard Sterry

New Character Set

A new and interesting character set can be produced by way of *FX155 to directly access the video ULA. The new characters are best viewed in mode 4 or 6 and can be produced by either *FX155,128 or *FX155,24Ø. The two commands produce different 'fonts', and adding a number between 1 and 15 to the values will produce the character set in colour. To reset the machine to normal, just issue a mode change command.

Alan Hatfield

Filling in a Form

The following short Wordwise Plus segment program will allow you to easily fill in a 'form letter'. To use it, enter your form letter in the usual way, but place a '\' character at each position where text is to be 'filled in' at a later stage and save the letter away.

Then load the segment program into a free segment and execute it by pressing Shift-fØ (if the program had been loaded into segment Ø). The screen will clear and you should now type the text that you wish to put in place of the '\' character, pressing Return at the end of each input.

When you have finished, you can re-load the 'blank' form to produce another version. The program can be improved further by providing a prompt for each input, but I will leave that up to others.

```
SELECT TEXT
CURSOR TOP
CLS
PRINT "Enter text"
REPEAT
FIND "\"
DELETE AT
A$=GLK$
TYPE A$
UNTIL EOT
```

T.K.Boyd

Program Protection

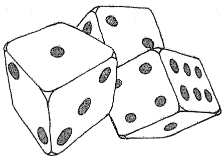
One good way of protecting a Basic program from being listed is to stop Basic from finding the end of a program in memory. A Basic program has a character code 255 at the end to signify the end of file. If Basic cannot find this byte then it prints the 'Bad program' message. To protect your program, load it in and type:

```
PRINT ~TOP
?(TOP-1)=Ø
*SAVE name AAAA BBBB
```

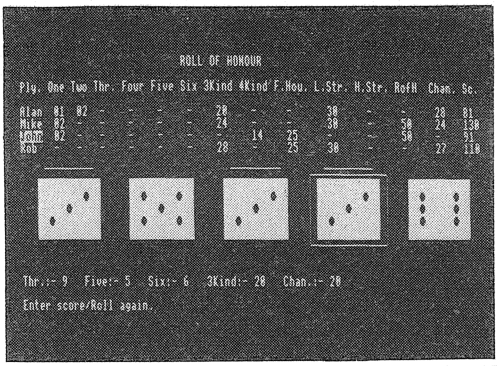
where AAAA is the current value of page and BBBB is the value for the top of the Basic program returned by the first statement. The program can then be loaded in at any time and RUN in the normal way. Any attempt to list will produce the 'Bad program' message.

Andrew Armstrong

ROLL OF HONOUR



After the frenzy of last month's game we bring you something more peaceful, an excellent implementation by Philip Thomas of the popular dice game Yahtze.



```
                    ROLL OF HONOUR
Ply. One Two Thr. Four Five Six 3Kind 4Kind F.Hou. L.Str. H.Str. RofH  Chan. Sc.
Alan  01  02  -    -    -    -   20    -     -     30     -      -    20   81
Mike  02  -   -    -    -    -   24    -     -     30     -     50    24  130
[illegible] 02  -   -    -    -    -    -   14    25     -      -     50    -   91
Rob   -   -   -    -    -    -   20    -    25     30     -      -    27  110

Thr.:- 9  Five:- 5  Six:- 6  3Kind:- 28  Chan.:- 28
Enter score/Roll again.
```

Roll of Honour is based on the popular dice game 'Yahtzee', combining strategy with an element of chance.

This is a game for one to four players in which the object is to score as many points as possible by throwing five dice. Thirteen different categories of 'hand' are presented and a score has to be entered into each category. Any or all of the five dice may be rolled a total of three times in completing each hand. After the first roll, the player is allowed to 'hold' any of the dice for the next throw, or to enter their score. The different hands and corresponding scoring are shown in the table.

Type in and save the game as usual. When you run the program, you first select the number of players and enter their names. For each player, the game proceeds as follows (the uppermost faces of the five dice are displayed on the screen).

Press 'R' to hold any dice. Then use the left and right cursor keys to move the box over the required dice and press the spacebar to hold or 'un-hold'. If a dice has a bar at the top, it means that it is held. Pressing the spacebar will remove the bar and cancel the hold on the dice. If a dice was held on the previous throw, then that dice may not be thrown again.

To score the current hand (or after a maximum of three throws) the 'S' key must be used to enter your score. Again the left and right cursor keys are used to move a cursor to the required score column and the spacebar is used to enter the score. Careful choice of category when scoring each hand is essential to achieve a high score. If, at any stage, you have a

SCORING HANDS

The categories of hand, and their scores, are as follows: ONES, TWOS, THREES, FOURS, FIVES and SIXES are scored as the corresponding face value multiplied by the number of times that face appears in the hand. For example, if you threw the sequence 1,3,5,2,3 then you could choose whether to enter this in the ONES category and score one point (1*1), or in the TWOS category and score two points (2*1), or in the THREES category and score six points (3*2), or in the FIVES category and score five points (5*1).

THREE OF A KIND is any hand where three of the dice show the same value. The score is the sum of all five dice. Similarly, FOUR OF A KIND is where four dice show the same value and the scoring is similar.

A FULL HOUSE is a hand consisting of three of a kind, plus a pair (e.g. 5,3,5,5,3). The score for a full house is 25 points.

LOW STRAIGHT is a run of four dice (regardless of the fifth), either 1,2,3,4 or 2,3,4,5 or 3,4,5,6. The score for this is 30 points. HIGH STRAIGHT is a run of all five dice, either 1,2,3,4,5 or 2,3,4,5,6 and scores 40 points.

ROLL OF HONOUR, the name of the game, is the hardest hand to achieve, being a set of five dice with the same face value. This is awarded 50 points.

The final category is CHANCE, where the score is the sum of the face values of all five dice.

hand which does not fit any of the remaining categories, then one of these must be selected and a zero score entered.

The game is completed after each player has entered a score in all thirteen columns, and the winner is the person with the highest score. As a target, a score of 200 is quite good, and a score in excess of 275 is very good.

Users with PAGE set higher than &E00 will have to append the following move-down routine to the start of the program before running.

```
1 IF PAGE<&E01 THEN 10
2 *K.0 *T.|MFORA%=0TO(TOP-PAGE)STEP4:
  A%!&E00=A%!PAGE:NEXT|MPAGE=&E00|MOL
  D|MRUN|M
3 *FX138 0 128
4 END
```

```
  10 REM Program Roll of Honour
  20 REM Version B0.4
  30 REM Author  P.Thomas
  40 REM Beebug  May 1987
  50 REM Program subject to copyright
  60 :
 100 ON ERROR GOTO 3120
 110 PROCinit:REPEAT:MODE7:PROCinputs
 120 MODE0:VDU23,1,0;0;0;0;
 130 GCOL3,1:PROCscreen
 140 FORloop%=1TO13:FORcurrent=1TOnum
 150 PROCname(0,129)
 160 FORX=1TO5:dice%(X)=1:NEXT:roll=0
 170 REPEAT
 180 FORX=1TO5:store%(X)=dice%(X):NEXT
 190 PROCrolldice:roll=roll+1
 200 PROCpossibles:PROCchoose
 210 UNTILexit%=2
 220 PROCname(1,128):NEXT:NEXT
 230 TIME=0:REPEAT UNTIL TIME>400
 240 MODE 7:PROCend:*FX21
 250 UNTIL G=78 OR G=110:*FX4
 260 VDU31,0,24
 270 END
 280 :
1000 DEF PROCinputs
1010 PROCheading
1020 PRINTTAB(7,10)CHR$134"Number of pl
ayer(s) : ";:A=GET:IF A<49 ORA>52 THENVD
U7:GOTO1020 ELSEnum=A-48:PRINT;num
1030 FORX=1TOnum:PRINTTAB(7,13)CHR$131"
Player ";X;" :-"
1040 PRINTTAB(7,16)CHR$134"Name of play
er : ";:INPUT""name$(X)
1050 L%=LENname$(X):PRINTTAB(25,16)STRI
NG$(L%," ")
1060 IFL%>4 THENVDU7:GOTO1040
1070 NEXT
1080 PRINTTAB(3,18)CHR$131"Do you want
the possible scores"'TAB(3)CHR$131"print
ed (Y/N) ";:A$=GET$:IFA$="Y" THENprint%=
1 ELSEIFA$="N" THENprint%=0 ELSEVDU7:GOT
O1080
1090 PRINT;A$:PRINTTAB(16,23)CHR$134"O.
K.":FORX=1TO450:NEXT
1100 FORX=1TOnum:FORY=1TO13:score%(X,Y)
=0:NEXT:NEXT
1110 ENDPROC
1120 :
1130 DEF PROCscreen:VDU19,0,4,0,0,0
1140 CLS:PRINTTAB(28,3)"ROLL OF HONOUR"
:PRINTTAB(0,5)score$:FORY=1TOnum:PRINTna
me$(Y):FORX=1TO14:PRINTTAB(x2(X),Y+6)"-"
:NEXT:NEXT
1150 ENDPROC
1160 :
1170 DEF PROCinit:*FX4,2
1180 DIMx1(13),name$(4),store%(5),x2(14
),dice%(5),val%(5),B$(4),die$(6),sort%(5
),score%(4,13),poss%(13),scr$(15)
1190 FORX=1TO13:READd:x1(X)=d:NEXT:FORX
=1TO14:READd:x2(X)=d:NEXT:FORX=1TO15:REA
Dd$:scr$(X)=d$:score$=score$+d$+CHR$(32)
:NEXT:VDU23,224,0,60,126,255,255,126,60,
0
1200 DATA5,9,13,18,23,28,32,38,44,51,58
,65,71
1210 DATA6,10,14,19,24,29,34,40,46,53,6
0,66,72,77
1220 DATA"Ply.","One","Two","Thr.","Fou
r","Five","Six","3Kind","4Kind","F.Hou."
,"L.Str.","H.Str.","RofH ","Chan.","Sc."
1230 D$=CHR$(10):B$(1)=STRING$(5," ")+C
HR$(224)+STRING$(5," "):B$(2)=CHR$(32)+C
HR$(32)+CHR$(224)+STRING$(8," ")
1240 B$(3)=STRING$(8," ")+CHR$(224)+CHR
$(32)+CHR$(32):B$(4)=CHR$(32)+CHR$(32)+C
HR$(224)+STRING$(5," ")+CHR$(224)+CHR$(3
2)+CHR$(32)
1250 die$(1)=D$+D$+B$(1)+D$:die$(2)=D$+
B$(2)+D$+B$(3):die$(3)=D$+B$(3)+B$(1)+B$
(2)
1260 die$(4)=D$+B$(4)+D$+B$(4):die$(5)=
D$+B$(4)+B$(1)+B$(4):die$(6)=D$+B$(4)+B$
(4)+B$(4)
1270 ENDPROC
1280 :
1290 DEF PROCrolldice
1300 P%=0:FORX=1TO5:IFdice%(X)=1 THENP%
=P%+9:val%(X)=RND(6):VDU28,((X-1)*16)+3,
17,((X-1)*16)+13,13:COLOUR0:COLOUR129:CL
S:PRINTdie$(val%(X));:SOUND1,-15,40+P%,1
1310 NEXT:VDU28,0,31,79,0:COLOUR128:COL
OUR1
1320 ENDPROC
1330 :
1340 DEF PROCpossibles
```

```
 1350 PROCconvert:PROCassess:PROCprintpo
ss
 1360 ENDPROC
 1370 :
 1380 DEF PROCconvert
 1390 FORX=1TO5:sort%(X)=val%(X):NEXT:FO
RY=1TO5:FORZ=1TO(5-Y):IFsort%(Z)>sort%(Z
+1) THENPROCswap
 1400 NEXT:NEXT
 1410 ENDPROC
 1420 :
 1430 DEF PROCswap
 1440 t=sort%(Z):sort%(Z)=sort%(Z+1):sor
t%(Z+1)=t
 1450 ENDPROC
 1460 :
 1470 DEF PROCassess
 1480 FORX=1TO13:poss%(X)=Ø:NEXT
 1490 PROCvalues:PROCchance:PROCthrees:I
Fcheck%=1 THENPROCfours ELSE1540
 1500 IFcheck%=1 THEN1520 ELSEPROCfullho
use
 1510 IFcheck%=1 THENENDPROC
 1520 PROCrofh
 1530 IFcheck%=1 THENENDPROC
 1540 PROCstraights
 1550 ENDPROC
 1560 :
 1570 DEF PROCvalues
 1580 FORX=1TO6:N=Ø:FORY=1TO5:IFX=sort%(
Y) THENN=N+1
 1590 NEXT:poss%(X)=X*N:NEXT
 1600 ENDPROC
 1610 :
 1620 DEF PROCchance
 1630 V=Ø:FORX=1TO5:V=V+sort%(X):NEXT:po
ss%(13)=V
 1640 ENDPROC
 1650 :
 1660 DEF PROCthrees
 1670 fhouse%=Ø:check%=Ø:IFsort%(3)=sort
%(1) THENfhouse%=1:check%=1
 1680 IFsort%(3)=sort%(5) THENfhouse%=2:
check%=1
 1690 IFsort%(2)=sort%(4) THENcheck%=1
 1700 IFcheck%=1 THENposs%(7)=poss%(13)
 1710 ENDPROC
 1720 :
 1730 DEF PROCfours
 1740 check%=Ø:IFsort%(1)=sort%(4) THENc
heck%=1
 1750 IFsort%(2)=sort%(5) THENcheck%=1
 1760 IFcheck%=1 THENposs%(8)=poss%(13)
 1770 ENDPROC
 1780 :
 1790 DEF PROCfullhouse
 1800 IFfhouse%=Ø THENENDPROC
 1810 check%=Ø:IFfhouse%=2 ANDsort%(1)=s
ort%(2) THENcheck%=1
 1820 IFfhouse%=1 ANDsort%(4)=sort%(5) T
HENcheck%=1
 1830 IFcheck%=1 THENposs%(9)=25
 1840 ENDPROC
 1850 :
 1860 DEF PROCrofh
 1870 IFsort%(1)=sort%(5) THENcheck%=1:p
oss%(12)=5Ø ELSEcheck%=Ø
 1880 ENDPROC
 1890 DEF PROCstraights
 1900 pair%=Ø:FORX=1TO4:IFsort%(X)=sort%
(X+1) THENpair%=pair%+1
 1910 NEXT
 1920 IFpair%>1 THENENDPROC
 1930 PROCcheckgaps:IFpair%=Ø THENPROCno
pairs
 1940 IFpair%=1 THENPROConepair
 1950 ENDPROC
 1960 :
 1970 DEF PROCcheckgaps
 1980 gflag%=Ø:gap%=Ø:FORX=1TO4:temp=sor
t%(X):IFsort%(X+1)>(temp+1) THENgap%=gap
%+1:IF X>1 ANDX<4 THENgflag%=1
 1990 NEXT
 2000 ENDPROC
 2010 :
 2020 DEF PROCnopairs
 2030 IFgap%=2 THENENDPROC
 2040 IFgflag%=1 THENENDPROC ELSEposs%(1
Ø)=3Ø
 2050 IFgap%=Ø THENposs%(11)=4Ø
 2060 ENDPROC
 2070 :
 2080 DEF PROConepair
 2090 IFgap%=Ø THENposs%(1Ø)=3Ø
 2100 ENDPROC
 2110 :
 2120 DEF PROCprintposs
 2130 IFprint%=Ø THENENDPROC
 2140 PRINTTAB(Ø,21);:FORX=1TO13:IFscore
%(current,X)=Ø ANDposs%(X)>Ø THENPRINTsc
r$(X+1);":- ";poss%(X);"   ";
 2150 NEXT
 2160 ENDPROC
 2170 :
 2180 DEF PROCchoose
 2190 IFroll=3 THENPRINT'''"Enter score."
 ELSEPRINT'''"Enter score/Roll again."
 2200 exit%=Ø:switch%=Ø:REPEAT
 2210 IFINKEY(-52) ORswitch%=1 THENPROCc
hoosedice
 2220 IFINKEY(-82) ORswitch%=2 THENPROCc
hoosescore
 2230 UNTILexit%>Ø
 2240 PRINTTAB(Ø,19);STRING$(255," ");ST
RING$(168," ")
 2250 ENDPROC
 2260 :
 2270 DEF PROCchoosedice
 2280 exit%=Ø:IFroll=3 THENENDPROC
 229Ø pl%=1:switch%=Ø:PROCdicemode
 23ØØ REPEAT
 231Ø FORX=1TO7Ø:NEXT
 232Ø IFINKEY(-26) THENPROCmovesquare(-1
,Ø,5,1Ø52,-256)
 233Ø IFINKEY(-122) THENPROCmovesquare(1
,6,1,28,256)
 234Ø IFINKEY(-99) THENPROCbar
 235Ø IFINKEY(-82) THENswitch%=2
 236Ø IFINKEY(-52) THENPROCcheckdice
 237Ø UNTILexit%=1 ORswitch%=2
 238Ø PROCdicemode
 239Ø ENDPROC
 24ØØ :
 241Ø DEF PROCdicemode
 242Ø xco%=28:FORX=ØTO4:IFdice%(X+1)=Ø T
HENMOVE(xco%+(X*256))+4Ø,634:DRAW(xco%+(
X*256))+176,634
 243Ø NEXT:xco%=28+((pl%-1)*256):PROCsqu
are
 244Ø ENDPROC
 245Ø :
 246Ø DEF PROCsquare
 247Ø MOVExco%,434:DRAWxco%,619:DRAW(xco
%+216),619:DRAW(xco%+216),434:DRAWxco%,4
34
 248Ø ENDPROC
 249Ø :
 25ØØ DEF PROCmovesquare(A,B,C,D,E)
 251Ø pl%=pl%+A:PROCsquare:IFpl%=B THENp
l%=C:xco%=D ELSExco%=xco%+E
 252Ø PROCsquare
 253Ø ENDPROC
 254Ø :
 255Ø DEF PROCbar
 256Ø IFstore%(pl%)=Ø THENVDU7 ELSEGCOLØ
,dice%(pl%):dice%(pl%)=dice%(pl%)EOR1:MO
VE(xco%+4Ø),634:DRAW(xco%+176),634:GCOL3
,1
 257Ø REPEAT UNTIL NOT INKEY-99
 258Ø ENDPROC
 259Ø :
 26ØØ DEF PROCchoosescore
 261Ø exit%=Ø:p2%=1:switch%=Ø:PROCcursor
(Ø,135)
 262Ø REPEAT
 263Ø REMFORX=1TO7Ø:N.
 264Ø IFINKEY(-26) THENPROCmovecursor(-1
,Ø,13)
 265Ø IFINKEY(-122) THENPROCmovecursor(1
,14,1)
 266Ø IFINKEY(-52) THENswitch%=1
 267Ø IFINKEY(-99) THENPROCscore
 268Ø UNTILswitch%=1 ORexit%=2
 269Ø PROCcursor(1,128)
 27ØØ ENDPROC
 271Ø :
 272Ø DEF PROCcursor(A,B)
 273Ø COLOURA:COLOURB:PRINTTAB(x1(p2%),5
);scr$(p2%+1)
 274Ø ENDPROC
 275Ø :
 276Ø DEF PROCmovecursor(A,B,C)
 277Ø PROCcursor(1,128):p2%=p2%+A:IF p2%
=B THENp2%=C
 278Ø PROCcursor(Ø,135)
 279Ø ENDPROC
 28ØØ :
 281Ø DEF PROCscore
 282Ø IFscore%(current,p2%)=Ø THENexit%=
2 ELSEENDPROC
 283Ø str$=STR$(poss%(p2%)):IFposs%(p2%)
>9 THENstring$=str$ ELSEstring$=CHR$(48)
+str$
 284Ø IFposs%(p2%)=Ø THENstring$="**"
 285Ø COLOUR1:COLOUR128:PRINTTAB(x2(p2%)
,(6+current))string$:IFposs%(p2%)=Ø THEN
score%(current,p2%)=255 ELSEscore%(curre
nt,p2%)=poss%(p2%)
 286Ø SC%=Ø:FORX=1TO13:SC%=SC%+score%(cu
rrent,X):IFscore%(current,X)=255 THENSC%
=SC%-255
 287Ø NEXT:PRINTTAB(x2(X),(6+current))ST
R$(SC%)
 288Ø ENDPROC
 289Ø :
 29ØØ DEF PROCend:CLS:PROCheading
 291Ø PRINTTAB(1Ø,1Ø)CHR$134"Score(s)":F
ORX=1TOnum:SC%=Ø:FORY=1TO13:SC%=SC%+scor
e%(X,Y):IFscore%(X,Y)=255 THENSC%=SC%-25
5
 292Ø NEXT:PRINTTAB(1Ø,12+X)CHR$131name$
(X)" : ";SC%:NEXT:*FX21
 293Ø PRINTTAB(Ø,2Ø)CHR$134"Another game
 (Y/N) :";:G=GET
 294Ø ENDPROC
 295Ø :
 296Ø DEF PROCname(A,B)
 297Ø COLOURA:COLOURB:PRINTTAB(Ø,6+curre
nt)name$(current)
 298Ø ENDPROC
 299Ø :
 3ØØØ DEF PROCcheckdice
 3Ø1Ø N=Ø:FOR X=1TO5:IF dice%(X)=Ø THENN
=N+1
 3Ø2Ø NEXT:IFN<5 THENexit%=1 ELSEVDU7
 3Ø3Ø ENDPROC
 3Ø4Ø :
 3Ø5Ø DEFPROCheading
 3Ø6Ø FORN%=1TO2
 3Ø7Ø PRINTTAB(1,N%)CHR$129CHR$157CHR$13
1CHR$141TAB(12)"Roll of Honour"TAB(38)CH
R$156:NEXT
 3Ø8Ø ENDPROC
 3Ø9Ø :
 31ØØ ON ERROR OFF:*FX4
 311Ø MODE7:IF ERR=17 END
 312Ø REPORT:PRINT" at line ";ERL
 313Ø *FX21
 314Ø END
```

BEEBUG MEMBERSHIP

Send applications for membership, membership renewals, membership queries and orders for back issues to the address shown. All membership fees, including overseas, should be in pounds sterling drawn (for cheques) on a UK bank.

MEMBERSHIP SUBSCRIPTION RATES

£6.90	6 months (5 issues) UK ONLY
£12.90	1 year (10 issues) UK, B.F.P.O., Channel Islands
£19.00	Rest of Europe
£26.00	Americas & Africa
£23.50	Middle East
£28.00	Elsewhere

BACK ISSUES (Members only)

VOLUME	SINGLE ISSUES	VOLUME SETS (10 ISSUES)
1	40p	single issues 7, 9 & 10 only
2	50p	£3.50
3	70p	£5.50
4	90p	£7.50
5	£1.20	£10.50
6	£1.30	–

DISCOUNT: Any 10 or more issues, deduct £1.50 from the total single issue price. 20 or more deduct £3.00, 30 or more deduct £4.50, etc.

Please add the cost of post and packing as shown:

DESTINATION	FIRST ISSUE	EACH SUBSEQUENT ISSUE
UK, BFPO + CH. IS.	40p	20p
Europe + Eire	75p	45p
Elsewhere	£2	85p

All overseas items are sent airmail (please send a sterling cheque). 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 that require an invoice. Note that there is no VAT on magazines.

Back issues are for members only, so it is ESSENTIAL to quote your membership number with your order.

BEEBUG
Dolphin Place, Holywell Hill, St. Albans, Herts. AL1 1EX

St. Albans (0727) 40303
Manned Mon-Fri 9am-5.00pm

(24hr Answerphone Service for Access/Visa orders and subscriptions)

If you require members discount it is essential to quote your membership number and claim the discount when ordering.

BEEBUG MAGAZINE is produced by BEEBUG Ltd.

Editor: Mike Williams
Technical Assistant: Alan Webster
Production Assistant: Yolanda Turuelo
Membership Secretary: Sara Anketell-Jones
Managing Editor: Lee Calcraft

Additional thanks are due to Sheridan Williams, Adrian Calcraft, John *Yale and Tim Powys-Lybbe.*

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 £40 per page, but please give us warning of anything substantial that you intend to write. A leaflet, 'Notes of Guidance for Contributors', is available on receipt of an A5 (or larger) SAE.

In the case of material longer than a page, we would prefer this to be submitted on cassette or disc in machine readable form using *"Wordwise", "View", or other means, but please ensure an adequate* written description of your contribution is also included. If you use cassette, please include a backup copy at 300 baud.

In all communications, please quote your membership number, and enclose a SAE if you require a reply.

BEEBUG
Dolphin Place, Holywell Hill, St. Albans, Herts AL1 1EX.

Produced by Head Office Design (0782) 286375 ISSN – 0263 – 7561

Magazine Cassette/Disc

CONTENTS CASSETTE/DISC MAY 1987

SELECTIVE BREEDING – delve into the principles of evolution and selective breeding with this high quality program.

VECTORED ENTRY INTO SIDEWAYS RAM – complete program, incorporating last month's key blip routine, for operation from sideways RAM.

THE MASTER SERIES - DFS to ADFS MULTI-FILE TRANSFERS – highly practical utility for Master users.

LOADING SIDEWAYS RAM WITHOUT A HARD BREAK – the solution to all those sideways RAM loading problems.

TURN YOUR BEEB INTO A CHART RECORDER – the complete program comprising both parts one and two.

ROLL OF HONOUR – excellent implementation of the Yahtzee dice game.

BEEBUG WORKSHOP – complete demonstration of a family tree database including details of 32 separate individuals.

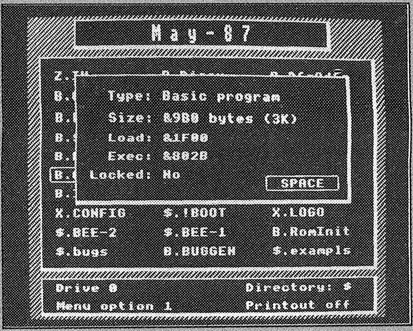
FIRST COURSE – enhanced function key definitions from this month's article.

READING *FX CALLS – two function key definitions to display information from *FX calls.

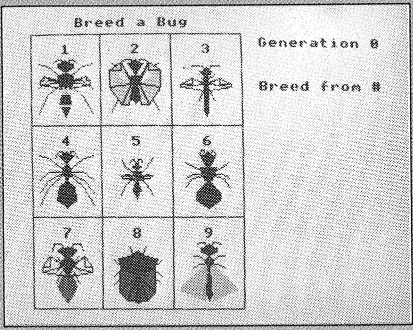
MIDAS: A UNIVERSAL DISC FRONT END – a highly sophisticated file management program for all your DFS discs.

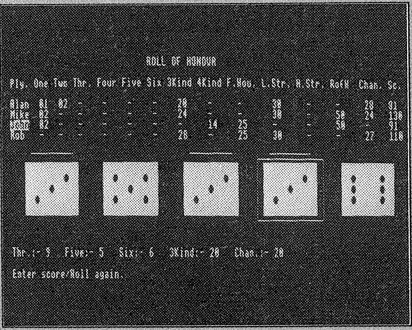
EXTRA FEATURES THIS MONTH

MAGSCAN – data for this issue of BEEBUG (Vol.6 No.1).


MIDAS Disc Front End

```
                May-87
 Z.
 B.     Type: Basic program
 B.     Size: &980 bytes (3K)
 B.     Load: &1F00
 B.     Exec: &8028
 B.   Locked: No
 B.                           SPACE
 X.CONFIG      $.!BOOT       X.LOGO
 $.BEE-2       $.BEE-1       B.RomInit
 $.bugs        B.BUGGEN      $.exampls

 Drive 0                Directory: $
 Menu option 1          Printout off
```


Selective Breeding


Roll of Honour

```
                       ROLL OF HONOUR
Ply. One Two Thr. Four Five Six 3Kind 4Kind F.Hou. L.Str. H.Str. RofH Chan. Sc.
Alan 01  02   -    -    -    -   28    -     -     30     -      -        28  91
Mike 02  -    -    -    -    -   24    -     -     30     -      50       24  130
MGSZ 02  -    -    -    -    -   -     14    25    -      -      50       -   91
Rob  -   -    -    -    -    -   28    -     25    30     -      -        27  110

Thr.:- 9  Five:- 5  Six:- 6  3Kind:- 28  Chan.:- 28
Enter score/Roll again.
```

All this for £3.00 (cass) £4.75 ($5\frac{1}{4}$" disc) £5.75 ($3\frac{1}{2}$" disc) + 50p p&p. ***(£1 overseas)***
Back issues ($5\frac{1}{4}$" disc since Vol.3 No.1, cass since Vol.1 No.10) available at the same prices.

Subscription rates (UK only)	$5\frac{1}{4}$" **Disc**	$3\frac{1}{2}$" **Disc**	**Cass**	**Subscription rates** (Overseas)	$5\frac{1}{4}$" **Disc**	$3\frac{1}{2}$" **Disc**	**Cass**
6 Months (5 issues)	£25.50	£29.50	£17.00	6 months (5 issues)	£30.00	£34.00	£20.00
12 Months (10 issues)	£50.00	£58.00	£33.00	12 months (10 issues)	£56.00	£64.00	£39.00

Prices are inclusive of VAT and postage as applicable. Sterling only please.

Cassette subscriptions can be commuted to $5\frac{1}{4}$" disc subscription on receipt of £1.70 per issue of the subscription left to run. ($3\frac{1}{2}$" disc £2 per issue)

All subscriptions and individual orders to:
BEEBUG, Dolphin Place, Holywell Hill, St.Albans, Herts. AL1 1EX.

Available only to BEEBUG MEMBERS

EXTENDED SPECIAL OFFER for Volume 6

BEEBUG

MAGAZINE DISC AND CASSETTE SUBSCRIPTION

ONE FREE DISC/TAPE when you subscribe for 6 months

TWO FREE DISCS/TAPES when you subscribe for 12 months

All the programs from each magazine ready to run.

- No more wasted time debugging programs.
- No more sore fingers.
- Mailed out at the same time as the magazine.

SUBSCRIPTION RATES FOR MONTHLY DISCS AND CASSETTES

SIX MTHS	UK	OV'SEAS
Cassette	£17.00	£20.00
5¼ disc	£25.50	£30.00
3½ disc	£29.50	£34.00
TWELVE MTHS	**UK**	**OV'SEAS**
Cassette	£33.00	£39.00
5¼ disc	£50.00	£56.00
3½ disc	£58.00	£64.00

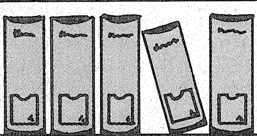
This offer applies to all new subscriptions and renewals received by the 31st May 1987. All new subscriptions commence with Volume 6 issue 1 (May 1987). Make sure you claim your free disc(s) or tape(s).

NOTE: Disc and Cassette subscriptions are available only to members and must be in addition to your BEEBUG membership.

CHOOSE: Six month subscription extended by ONE free monthly disc or tape; twelve month subscriptions extended by TWO free monthly discs or tapes.

ALTERNATIVELY: Select any previous discs or tapes quite FREE (1 for six month subscriptions, 2 for twelve month subscriptions), 5¼" discs from Vol.3 No.1, cassettes from Vol.1 No.10. All back issue discs will be dispatched as soon as your subscription has been processed (subject to stock availability).

PLEASE INDICATE YOUR CHOICE CLEARLY



BEEBUG BINDERS

BEEBUG magazine binders are in dark blue with "BEEBUG" in gold lettering on the spine. Each binder will accommodate the whole of one volume of the magazine together with index. Buy now to store your copies of Volume 5; buy now ready for the start of Volume 6.

Only **£3.90 each** (TWO for £7.50)
Post & Packing per binder £1.00 (UK), £1.50 (Europe), £2.50 (Elsewhere) — no VAT on binders

ORDER FORM

I wish to subscribe to the magazine cassette/disc. ☐ **6 months** ☐ Cassette ☐ 5¼ Disc ☐ 3½ Disc ☐ **12 months** ☐ Cassette ☐ 5¼ Disc ☐ 3½ Disc Cost £ ______

OR ☐ I wish to extend my subscription by one/two free issues.
☐ I wish to claim my free back issues: ______ Cost £ ______

BEEBUG magazine binders: Number required: ______ Post & Packing £ ______

Name ______ **Total cost of your order** £ ______

Address ______ I enclose cheque for £ ______ made payable to BEEBUG Ltd.

______ or please debit my Access/Visa Account Number Card Expiry Date

Postcode ______ **Access/Visa No.** ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature ______ **Expiry Date** ☐☐

Send your order to **BEEBUG TAPE/DISC Offer, Dolphin Place, Holywell Hill, St. Albans, Herts AL1 1EX**
OR call **St. Albans (0727) 40303** to place your order by phone
(quote your membership number and don't forget to claim your free discs or tapes).