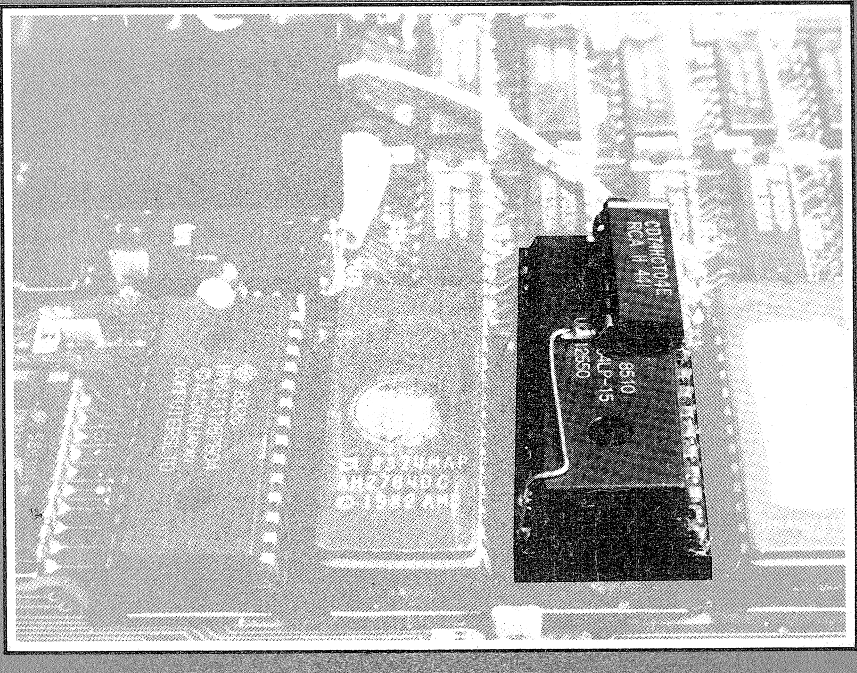
VOLUME 5 NUMBER 7 DECEMBER 1986 PRICE £1.30

BEEBUG

FOR THE BBC MICRO



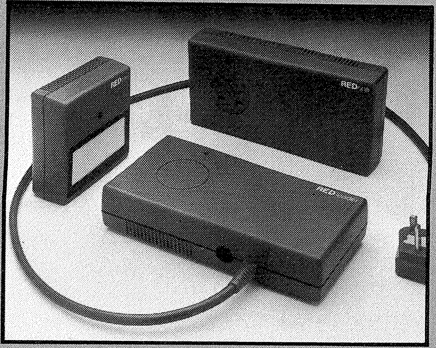
BUILD YOUR OWN SIDEWAYS RAM

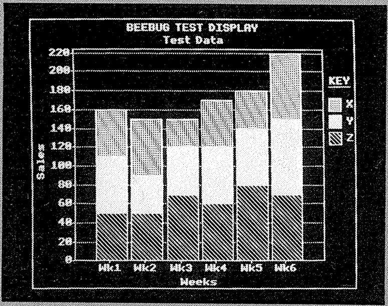
FILE SECURITY TECHNIQUES

BEEBUG

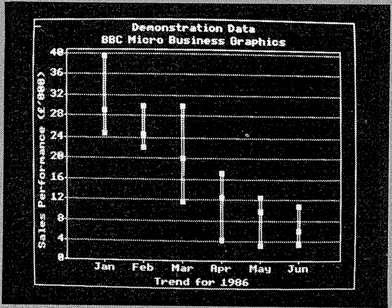
Volume 5 Number 7
December 1986

Red Boxes

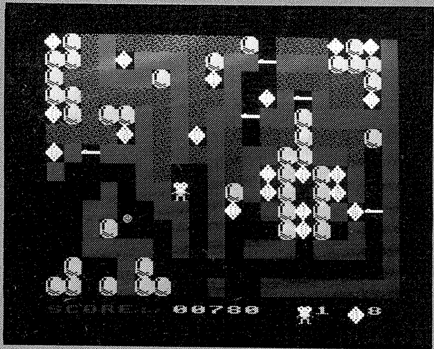




Business Graphics



Business Graphics



Pitfall Pete

FEATURES

File Security Techniques 8
Build Your Own Sideways RAM 10
Business Graphics (Part 2) 17
Sorting Data Files 21
Program Cruncher 24
The Master Series
64K Printer Buffer Utility 38
VDU Graphics Made Simple 41
Master Hints 42
Master Clock Manager 43
First Course — Character Formation Part 2 44
BEEBUG Workshop — Data Driven Routines 50
Pitfall Pete 57

REVIEWS

Red Boxes to Control 6
Computer Aided Design 14
Educad Micad Novocad
Acorn's ViewPlot Reviewed 26
Games Reviews 52
Stryker's Run Chart Challenge
Mikie Cluedo
ROM Round-Up 54
Advanced 1770 DFS
Advanced Disc Investigator
Floppywise Master BROM+

REGULAR ITEMS

Editorial Jottings 3
News 4
BEEBUGSOFT Forum 5
Supplement 28-37
Postbag 48
Hints and Tips 49
Points Arising 61

HINTS AND TIPS

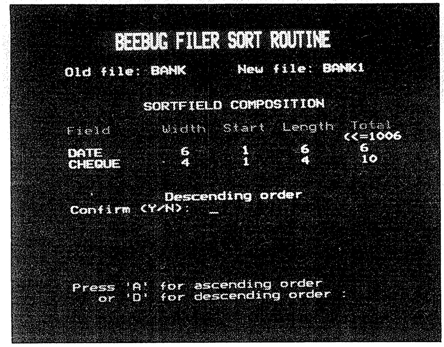
GENERAL

Speech Translation
Disabling DFS's
Cursor Turn off
Star Command Workspace
BEEBUG Editor Extensions
Vertical Printing

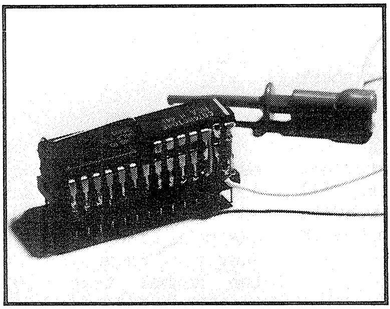
MASTER

Checking Available RAM Banks
Displaying and Saving Function Keys
ADFS Shortcuts
ADFS Wildcard Wipe
ADFS *ACCESS

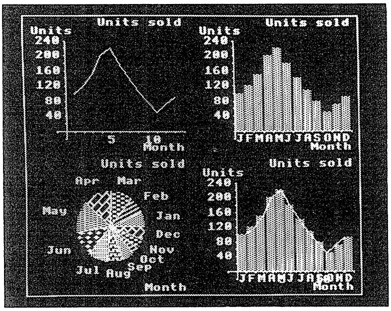
Sorting Data Files



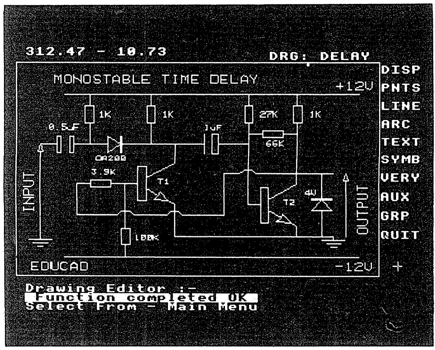
```
BEEBUG FILER SORT ROUTINE
Old file: BANK        New file: BANK1

        SORTFIELD COMPOSITION
Field       Width  Start  Length  Total
                                  <<=1006
DATE          6      1      6       6
CHEQUE        4      1      4      10

          Descending order
Confirm (Y/N):  _


Press 'A' for ascending order
   or 'D' for descending order :
```



Building Sideways RAM



ViewPlot Reviewed



```
Drawing Editor :-
 Function completed OK
Select From - Main Menu
```

Computer Aided Design

EDITORIAL JOTTINGS

CONTRIBUTING TO BEEBUG

Most of what you read in BEEBUG has been contributed by BEEBUG members. Some are long-standing authors whose names you will find in many past issues of the magazine. Equally, many of the articles and programs that we publish arrive out of the blue, and it is true to say that many of our best programs and articles have appeared in this way.

We would like to encourage more members to consider sending in their best programs for possible publication. All programs are looked at carefully, before any decision is made, and all material used is paid for on publication, at up to £40 per page. We are particularly interested in applications of all kinds, but all good quality programs are welcome. If you have a program which you feel would be of interest to other BEEBUG members, why not send it in, preferably with an accompanying article. Use previous issues of BEEBUG as a guide to length and style. We also have some "Notes of Guidance for Contributors" which are available on receipt of an A5 SAE.

NEW MAILING SYSTEM

Many of you will have noticed the change in the form of mailer used to distribute the magazine. As with many other magazines, we are now using a transparent plastic mailer which allows the mailing process to be more effectively automated, and integrated with the printing. This system has not yet been introduced for those who also get a renewal notice, but this part of the mailing will be included at some time in the future. As a result, we hope that we shall be able to greatly streamline the mailing process.

BEEBUG VOUCHERS

We are now able to offer BEEBUG vouchers in £1, £5 and £10 denominations. There are no extra charges (other than 50p p&p if ordered alone) and the vouchers are redeemable against any purchase from BEEBUG including hardware, software, membership and renewals. The ideal Christmas present for the BBC micro enthusiast. Order yours now (or get someone to buy them for you).

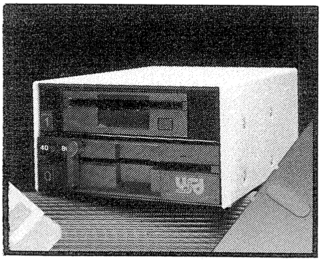
PROGRAM/REVIEW CLASSIFICATION

We hope that the new classification symbols introduced last month for programs and reviews clarify matters with regard to the growing variety of Acorn Systems. The complete set of symbols is shown below. These show clearly the valid combinations of machine (version of Basic) and filing system for each item (and Tube compatibility). Remember that a single line through any symbol indicates partial working (normally a few changes will be needed) and a cross over a symbol shows total unsuitability. For reviews we do not distinguish between Basic I and Basic II, but merely put a 'B', for the model B.

COMPUTER SYSTEM			FILING SYSTEM	
Master	(Basic IV)	M	ADFS	A
Compact	Basic IV)	C	DFS	D
Compact	(Basic IV)	C4	Cassette	⌧
Model B	(Basic II)	II	TUBE COMPATIBILITY	
Model B	(Basic I)	I	Tube	⊖
Electron		E		

News News News News News News Ne

More Disc Drives



Akhter has launched two new disc drives for the BBC micro. For hard disc users there is the 'Archive' combined floppy disc drive and tape streamer unit. The two drives are built into a bridge type monitor stand to fit over the Beeb, and provide normal floppy disc storage (8Ø track, double sided) and tape cartridge backup for a hard disc connected to the 1MHz bus. Backup software is also provided. Archive costs £999.95.

Akhter is also offering a 'Combo' combined 5.25 in. and 3.5 in. disc drive. Now that software is available in both formats for the older BBC machines and the new Master Compact, users may wish to read both types of disc. The two units are built into a standard twin disc drive case and the 5.25 in. drive is provided with a 4Ø/8Ø track switch. Software can be transferred from one format to the other with the dual format unit. The Combo costs £249.95 without a power supply, and another £1Ø with power supply, from Akhter on Ø279-443521.

Speak Out

R & D Speech Technology has released a speech recognition system for the BBC micro. Micro-voice plugs into the 1MHz bus and comes complete with a microphone. The device is 'trained' to recognise ten words at a time which it can then respond to within half a second.

R & D claims just about any program can incorporate the speech recognition system to make a wide range of tasks voice controlled. The Micro-voice costs £171 (that's £17 per word!). Further details from R & D on Ø326-7529Ø.

Tea Break

Although a cup of tea or coffee while working at the micro keyboard is a good idea for the user, it can be dangerous for the micro. A Viziflex seal prevents any damage from spilt drinks but allows the micro to be used almost as normal. Moulded to the shape of the Beeb's keyboard, the clear Viziflex cover sticks in place and seals the micro from dust, liquids, or any other unwanted add-ons. A BBC micro Viziflex costs £19.95 from Ceratech Electronics on Ø42Ø-88674.

Wordwise Training

The TVEI Wordwise Training Pack is published by Muse with the support of Computer Concepts. The training pack is a complete introduction to using the Wordwise word processor and contains study material, interactive exercises on disc, a reference book and glossary, sixty reference cards and a 'prompt' card for quick keyboard-side key and command reminders. The pack is intended for users of both Wordwise and Wordwise Plus and costs £28 from Jordanhill College Publications on Ø41-959 1232.

Scientific Words

The first word processing package for the Acorn Cambridge Workstation and 32Ø16 second processor has been produced by Vuman. Vuwriter Scientific offers the normal text editing functions and two character sets, including Greek and scientific symbols (5ØØ characters in all), and an overprint capability for producing more complex characters on the screen. Vuwriter costs £11Ø from Vuman on Ø61-969 4778.

Oaks and Acorns

Oak Computers is offering a range of business-style micros based on the Master 128. The PCM range features a detached keyboard with a main system box holding the Master circuit board, disc drives and various optional extras. Prices range from £495 for a PCM without disc drives to £4,375 for a system with twin drives, 32Ø16 second processor and 4ØMb hard disc drive. Versions with Z8Ø second processor and V21/V23 modem are also available. Details from Oak on Ø532 5Ø2615.

BEEBUG SOFT FORUM

New Service to Members

We are being increasingly asked by members for advice on which computer hardware they should buy. There are so many different altern- atives to choose from that even experienced users are often confused by the selection. Also, in a number of instances, we are hearing of members who have bought equipment only to discover that the product is not suitable for the task they required. We are therefore offering a new service to help members decide on what equipment to purchase. We have produced seven fact sheets covering all the major topics, and these are available free of charge. The subjects to be covered are:

1 Compact and Master Computers
2 Disc Drives
3 Disc Filing Systems
4 Printers
5 RAM/ROM Boards
6 Monitors
7 Second Processors

Please write to our St Albans address and mark the envelope 'Hardware Fact Sheets' enclosing a stamped addressed envelope. Please quote your BEEBUG membership number and the numbers of the sheets you require.

ADFS Masterfile Compatibility

Many members are expressing an interest in ADFS Masterfile II, but are concerned that their existing data files will not be compatible with the ADFS version. Existing data files can be transferred to the ADFS by using the Copy utility on the ADFS utility disc. Please note that ADFS Masterfile is compatible with the official Acorn ADFS and the BBC model B, B+, Master 128 and Compact.

Dumpmaster ROM

A new version of the Dumpmaster ROM is now available offering improvements for the Master 128. Screens can now be dumped and saved from shadow RAM and extra screen dumps are provided for the following printers:

Epson JX8Ø
Paper Tiger (IDS44Ø/445)
Walters WM2ØØØ/4ØØØ

For further details, please contact our technical department on (Ø727) 6Ø263.

BEEBUG Disc Prices Down

We are pleased to pass on from the manufacturer, a significant reduction in the price of BEEBUG blank discs. These are the same high quality discs from a leading manufacturer, and represent even greater value for money. BEEBUG discs are individually tested and guaranteed for life by BEEBUG, and are supplied in a free storage box. Please check the Dynamic Disc advert for full details of the new prices.

New Magic Modem Software

Beebugsoft will soon be releasing a new communications ROM for the BEEBUG Magic Modem. The ROM includes over 5Ø commands covering all the usual features such as Download, Text Terminal, Viewdata Terminal, but also offers many special features, for example:

* All features are available from * commands, allowing Basic programmers to write their own communications software with ease.

* A powerful telephone directory is included in ROM, allowing you to enter all your important numbers. No need to remember numbers any more, just specify the name you require e.g *CALL 'prestel'.

* Comprehensive teletext editor with facilities to save a number of screens in one file, dump screens, load screens and produce a carousel display.

* Other special features include an auto-answer facility, a real time clock, frame grabbing, and much more!

The software is supplied on a 16K ROM with a comprehensive manual and function key strip. A disc containing a selection of utility programs is also being prepared. See our adverts for prices and availability.

RED BOXES To Control

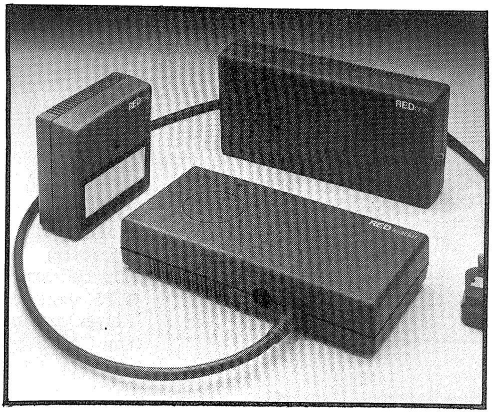
Look what Chris Curry, one of Acorn's original founders has dreamed up now — Red Boxes to control your house and home. Ian Waugh just couldn't resist the temptation and reports on his experiences.

Product : Red Boxes
Supplier : Electronic Fulfilment Services,
Chesterton Mill, French's Road,
Cambridge CB4 3NP.
Tel. (0223) 323143
Price : £129.00 (plus £4 p&p) for
starter kit of Red Leader,
Red One and Red Two.

In the early days of home computing, one of the most talked-about areas of application was computer control. A common topic was security: switching lights on and off at random, and controlling sensing devices to activate a burglar alarm. Another favourite was environmental control, monitoring temperature for example and switching appliances on or off as required.

Theoretically, at least, it should have been an easy matter to put just about any electrical device under computer control but early control units had severe limitations. Most required an elaborate interface involving trailing wires, and none had the intelligence which is built into the Red Box system. Such devices also had to be permanently connected to the host computer, increasing the cost considerably, and relegating the computer to the role of a simple timer.

Red Boxes, developed by Chris Curry's new company, General Information Systems (GIS), takes a different tack and one which is so simple and obvious you may well wonder why it hasn't been done before. The Boxes plug into the standard 13 amp ring main and communicate with each other using a mains-borne 129 kHz carrier, a frequency set aside by the British Standards Institution for just such a purpose. The devices to be controlled plug into the Red Boxes and this takes care of the safety aspects (e.g. protection from the mains voltage) as well as removing the need for lots of trailing wires. They will operate over a distance of approximately 200 metres so you can scatter them liberally throughout a fair-sized mansion.



The starter system includes a Red Leader, a Red One and a Red Two. Red Leader is the master control unit and is equipped with its own 6502 based computer and 8K of RAM. All communication is directed through this unit. Red One is an on/off switch and Red Two is an infra-red movement detector. Red Leader connects to the RS423 port of your Beeb, and a terminal program downloads upon switching on. This is simply to allow communication between Red Leader and your computer; the actual control program runs inside Red Leader, not the host micro. The beauty of the system is that after programming Red Leader you can unplug your computer and Red Leader will monitor the other units and carry out quite complex sets of instructions on its own.

The system is very security conscious. New Red devices are introduced to Red Leader by their security number (quite a lengthy sequence of digits) of which there are over 16 million combinations. As well as making the system virtually tamper-proof, it will let several Red Box systems operate on the same wiring without interference with each other, or other systems.

Once downloading has taken place - a matter of some 15 seconds - a menu appears on the screen. You can use this to give the Boxes simple instructions, such as times to switch on and off, and one device can be made to control another. For example, the sensor (Red Two) could switch on an alarm plugged into Red One. You can also control and monitor the Red Boxes in real time (for example, remotely switching on or off a TV or a light - great fun), but the real advantage is that once set up, you can disconnnect your Beeb and leave Red Leader to look after everything itself. Using the Beeb in this way makes set up and control of the Red Boxes very easy whilst keeping the cost of the Red Boxes down to a reasonable level.

Although you could probably set up a simple security system using only the menu, the possibilities expand enormously when you delve into Red Basic, the programming language for the Red Boxes. It is near enough to 'ordinary' Basic to ensure quick familiarity, but has several new commands for full control of the Red Boxes. I wrote a couple of small programs in a fairly short time.

Although the manual lists all the Red Basic commands, a little more help would have been welcomed, and perhaps some practical examples. In particular, it doesn't tell you how to get your program running in Red Leader before disconnecting from your Beeb - just press the reset button on Red Leader. I also had problems downloading the terminal program into my Master. Acorn, you see, saw fit to set the Master's RS423 default baud rate to 1200 baud, not the 9600 of a standard Beeb. Not GIS's fault, but I wasted a lot of time trying to track down this 'fault'.

Among the more interesting Red Basic instructions are EVERY, TELL and WHEN. EVERY is rather like a FOR-NEXT loop but provides a simple way of repeating a set of instructions at specified time intervals. TELL and WHEN could well be used together: WHEN one device sends a message (or signal) TELL another device to do something (switch on for example). There are several other commands and functions all designed to make the writing of your own control programs a relatively simple task.

Red Boxes go much further than simple on/off and timing devices. A lot of emphasis is placed on the security aspects and intelligence of the system. When messages are sent to Red Leader - say a sensor has been tripped - Red Leader acknowledges receipt of the message and if no acknowledgement is received the originating box repeats the message. Conventional systems would just send a single message, and leave it at that, and thus could fall foul of a mains spike, or whatever.

It is also worth mentioning the new Red Box modules under development. Of prime interest, I would suspect, to all Red Box users is a real-time clock which will fit into Red Leader. This will have a battery back-up so you can unplug Red Leader once it's been programmed and re-situate it. I think this should have been fitted as standard, but it will cost around £10-£15 as an extra.

We can soon expect to see analogue devices for temperature measurement and light dimmer control, window contact switches, pressure mats and an alarm. This will be battery-backed so once triggered it can't be silenced by unplugging it. Additional Red Ones and Twos cost £34.95 (plus £2 p&p) and it is hoped that the other modules will cost about the same.

The analogue devices will enable the most fastidious control over room and house temperature and lighting. Heating can be switched on and controlled while there is movement in a room (don't fall asleep watching TV) and you could even monitor and record the temperature of a room during the course of a day or a week and program your heating system accordingly.

Red Boxes have pulled together the technology for implementing a home or security control system, and packaged it so that everyone can use it. The only adverse factor I can see is the cost of fitting a house with Red Boxes, but with clever programming the system could control both an alarm and environment.

I think Chris Curry has another winner here: an affordable and eminently practical method of computer control. A matter for discussion in '81 - a reality in '86.

We are selling Red Boxes with 5% members' discount - see Mail Order insert.

File Security Techniques

David Graham presents an encryption machine that will hide program and data files alike away from prying eyes.

With the rise of the hacker as high tech hero of the eighties, the question of computer security has received the full glare of media attention. But what can Jo Micro do to keep his own data secure? The program presented here goes some way towards a solution.

Essentially it will encode any computer file stored on disc; whether this be a program, a text file, or some other data file. The file once encoded will be completely unreadable, and as far as I can see, completely uncrackable (?), unless you have the encryption program together with a unique 'password' to decode it. The file may then be stored, or sent in the post on disc, or passed over a telephone line using a modem to another user, without the fear that it could be read by anyone except the intended recipient.

First of all, type in the program and save it away. When you run it, it asks for the filename of the file to be encoded. You may precede this with a drive number and directory name(s) in either DFS or ADFS format if you wish. If the program cannot find the file, it will inform you, and offer to catalogue drive zero. Once an acceptable filename has been given, you are prompted for a new filename to be used for the encrypted file.

You are then asked whether you require "Random or Text coding". Essentially, two different types of encoding are catered for, and these will be discussed in a moment. Depending on which you select, you will be prompted for either a numeric or a text code. This will be the password which will subsequently decode your file when required, so do not forget it!

Once this is entered, encoding will begin, a byte at a time, and each byte read (except control codes below ASCII 32) will be scrolled in a screen window. As you will have gathered, Encoder produces a completely separate coded file, leaving the source file unchanged. For the utmost security, the source file should be deleted, and the disc compacted. But once you have done this you are completely at the mercy of your password.

THE ENCODING TECHNIQUE

At the heart of the encoder/decoder is the exclusive-OR operator (EOR). This has the extremely useful property that if you EOR two numbers A and B together, the result, C, will be unique, such that C EOR A will regenerate B. This means that the same password can be used to both code and decode a message. All we have to do is to make sure that the password, represented by A in this case, is not easily found.

The encoder also uses a certain amount of filtering so that it does not encode (or attempt to decode) characters whose ASCII codes lie in the range Ø to 31. These are often used as control codes, and if encoded would mean that the encrypted file could not be used on public networks which expect the user to supply text only messages. For some purposes you may find that you wish to reduce the encryption range still further. If you change line 14ØØ to read:

```
14ØØ IF bytein>31 AND bytein<128 BPUT#
W%,FNcode:VDU bytein ELSE BPUT# W%, by
tein
```

then only characters in the range 32 to 127 will be altered. But you should note that you can only alter this input filter at distinct bit positions (e.g. 32, 128 etc.). And incidentally, you may notice that to make even filtering of this kind possible, not the whole of the character to be encoded is EORed with the code, only the lower five bits in fact: selected by an AND operation with &1F.

To render the encoded file as difficult to crack as possible, one should not use the same byte as a code for each character of the source file. The Text option provides a simple way around this. You enter a text string as the password, and the characters from this are repeatedly used in sequence to provide the coding byte. The longer the password that you enter, the more secure will be the encoded file - but of course, the harder it will be to remember the password.

The other menu option uses an interesting property of the random number generator in the BBC micro to generate a different coding byte for EACH byte of the source file. The password given at the start of the program is used as a seed for the random number generator. The generator is then called for each byte of the source file. Fortunately, as soon as the generator is re-seeded with the same number it will produce an identical sequence of numbers, which can then be used to decode the file.

The range of possible seeds at your disposal is very large - somewhere between 1Ø^-9 to 1Ø^9, with no likelihood of repetition for a given seed, even with very long files. In tests, I found no repetition pattern even after 4 million calls for a given seed. But if you wish to encrypt still further, you can pass a file through the encoder twice with different passwords. When decoding after multiple encryption, the order in which the passwords are used is not important - again this is a consequence of the properties of the EOR operator.

Finally, a word of warning to Master users. The random number generator in Basic IV is different from that of Basics I and II. This only causes a problem if you wish to pass randomly encoded files between Master and Model B. The simple solution is to use BAS128 on the Master (or Compact). This disc-based Basic is derived from Basic II, and uses the same random number generator as the Model B.

```
   1Ø REM Program File Encoder
   2Ø REM Version B Ø.2J
   3Ø REM Author  D.E.Graham
   4Ø REM Beebug  November 1986
   5Ø REM Program Subject to copyright
   6Ø :
  1ØØ ON ERROR GOTO 147Ø
  11Ø REPEAT:MODE7:PROCstart
  12Ø PROCfile:PROCseed:PROCencode
  13Ø PROCend:UNTIL A=81
  14Ø MODE7:END
  15Ø :
 1ØØØ DEFFNcode
 1Ø1Ø =(bytein AND &EØ) + ((bytein AND &
1F) EOR EVAL("FN"+name$))
 1Ø2Ø :
 1Ø3Ø DEFFNrnd:=RND(31)
 1Ø4Ø :
 1Ø5Ø DEFFNtext:A=1-A*(NOT(A=LEN(text$))
)
 1Ø6Ø =ASC(MID$(text$,A,1)) AND &1F
 1Ø7Ø :
 1Ø8Ø DEFPROCnofile
 1Ø9Ø VDU7:PRINT'"File not found"
 11ØØ PRINT"Press space to retry, C to c
at disc ";
 111Ø IF (GET AND &DF)<>67 CLS:ENDPROC
 112Ø PRINT:VDU14:*CAT
 113Ø PRINT'"Press space to re-try"
 114Ø VDU15:A=GET:CLS:ENDPROC
 115Ø :
 116Ø DEFPROCstart
 117Ø A=Ø:CLOSE#Ø:FOR B=2 TO 3
 118Ø PRINTTAB(9,B)CHR$131;CHR$141;"FILE
 ENCODER"
 119Ø NEXT:VDU28,Ø,23,39,6:ENDPROC
 12ØØ :
 121Ø DEFPROCfile:REPEAT
 122Ø INPUT"Read filename:  "read$
 123Ø R%=OPENIN(read$)
 124Ø IF R%=Ø PROCnofile
 125Ø UNTIL R%<>Ø
 126Ø INPUT"Write filename: "write$
 127Ø W%=OPENOUT(write$):ENDPROC
 128Ø :
 129Ø DEFPROCseed
 13ØØ PRINT'"Random or Text coding (R/T)
: ";
 131Ø REPEAT:C=GET AND &DF: UNTIL C=82 O
R C=84
 132Ø PRINTCHR$C
 133Ø IF C=82 name$="rnd":INPUT"Seed (an
y number): "seed:R=RND(-ABS(seed))
 134Ø IF C=84 name$="text":INPUT"text st
ring "text$
 135Ø ENDPROC
 136Ø :
 137Ø DEFPROCencode:VDU28,Ø,23,39,12
 138Ø REPEAT:bytein=BGET# R%
 139Ø IFbytein=13 VDU1Ø,13
 14ØØ IF bytein>31 BPUT# W%,FNcode:VDU b
ytein ELSE BPUT# W%, bytein
 141Ø UNTIL EOF#(R%):CLOSE#Ø:ENDPROC
 142Ø :
 143Ø DEFPROCend:PRINT''"Coding complete
"
 144Ø PRINT"Press space to re-run, or Q
to quit ";:A=GET AND &DF
 145Ø ENDPROC
 146Ø :
 147Ø ON ERROR OFF:MODE7
 148Ø REPORT:PRINT" at line ";ERL:END
```

BUILD YOUR OWN SIDEWAYS RAM

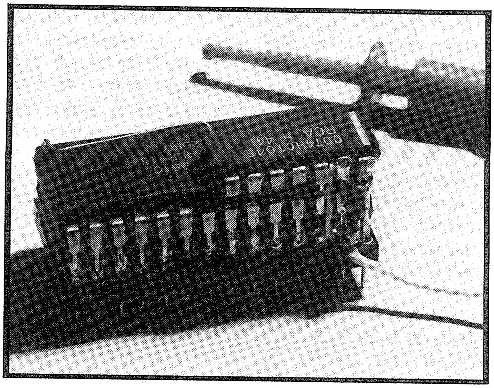
William Fulton explains how to add sideways RAM to your machine without the cost of an expansion board.

SIDEWAYS RAM

Sideways RAM has been an extremely popular add-on for the BBC micro, finding such applications as printer buffers, the storing of ROM images, the development of ROM software, and even sophisticated silicon disc filing systems, like that provided by the Romit ROM. And while many excellent sideways RAM modules are currently available, it is quite feasible to construct an 8 or 16K module, which will plug directly into any vacant sideways ROM socket in the Beeb. The RAM required for this, the extremely versatile 6264 8K static RAM, is obtainable at very low cost, and all you will need apart from this component are a couple of small hardware items, and an ability to solder.

CONSTRUCTIONAL DETAILS

The accompanying figures show the essential constructional details. As you can see, the 16K unit requires two RAM chips to be soldered one on top of the other. And while the 8K module can be plugged directly into an existing ROM socket, a carrier device is nevertheless recommended. The carrier is essential in the 16K version for the correct handling of address line 13, used to determine which of the pair of 8K RAMs is being accessed at any given time. This is handled by one section of a logic IC (74LS Ø4) mounted above the two RAM chips.



On the 8K version, only two pins (26 and 27) are not connected directly to the carrier, and these can either be cropped back or turned upwards to provide solder terminals. On the 16K version, pin 26 of the upper device and pin 27 of the lower must be similarly treated. Since all other pins are common, the two devices can effectively be soldered together piggy-back. All but pins 13 and 14 of the 74LSØ4 package are cropped off, with pin 7 linked to pin 14 of the 6264 using a short length of wire. The resistor between pins 27 and 28 of the RAM chips on either version should be 1Øk ohms, and ideally 1/8W size.

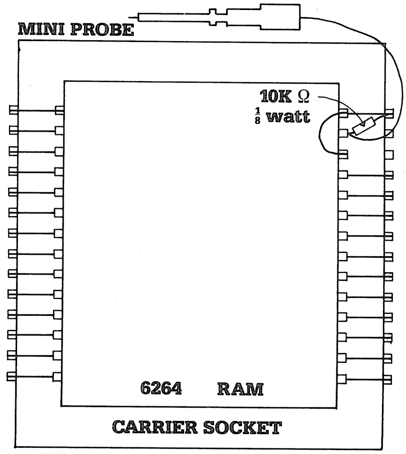


Fig. 1 Exaggerated perspective view from top of the 8K RAM module.

In both the 8 and 16K units, a flying lead must be attached to pin 27 of the RAM to facilitate Write operations. The Beeb's ROM sockets have no Read/Write line, because they were intended only for ROMs. The other end of this lead can be connected to any of the following three points to pick up the Write line:

IC77	pin 8
IC78 (8271)	pin 1Ø
IC73 (ADC)	pin 24

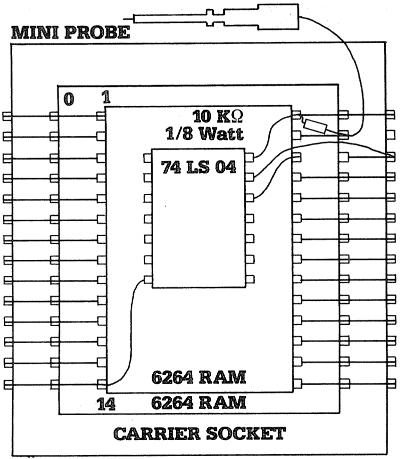


Figure 2 Exaggerated perspective view from the top of the 16K RAM module.

The connection may either be soldered, or you may use a suitable connector (such as RS 424-175 or Maplin YX57M) to clip onto an IC leg. When complete, the device can be installed in any ROM socket, though preferably at a lower priority than the prime language ROM. With the Write lead connected, it is ready for testing.

RAM-LOADING SOFTWARE

It is not possible to load images directly into the sideways RAM from disc, but this job is easily performed with the short accompanying routine. This should be typed in, and saved away before use. When the program is run, it requests a filename of a suitable ROM image, and loads this directly into user RAM (starting at &3C00). When a socket number is supplied (0-15), the RAM image is transferred to the required sideways RAM. The program then tests the first four bytes of the sideways RAM image against the image held in user RAM, and informs the user if the load operation has been successfully performed. The program then prompts you to execute a Ctrl-Break to initialise the ROM image. Performing *HELP should then reveal the presence of the new image.

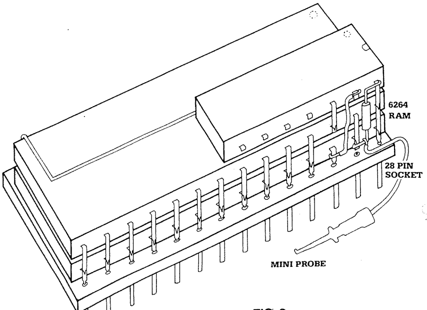


FIG.3

If you are not sure which socket you have put the RAM module into, remember that the rightmost socket (which should normally hold Basic) is number 15, with 14 to its immediate left, and so on. You should also note that some commercial ROM software is protected, and will not normally run from sideways RAM.

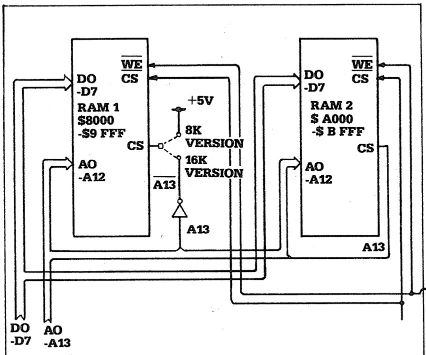
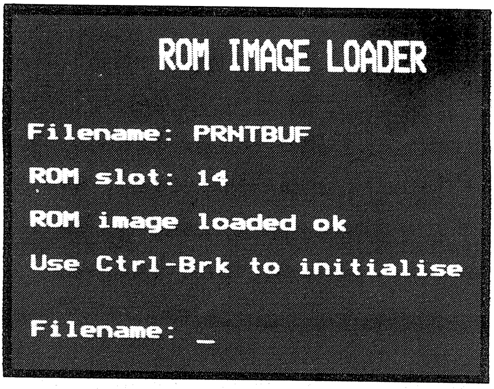


Figure 4 Schematic diagram of 8 & 16K RAM modules.

Unprotected ROMs may be copied to disc or tape using the ROM Controller program from Beebug Vol.5 No.3 or the extremely concise program by Bernard Hill in Beebug Vol.5 No.2 page 30, but these programs should not be used to break the copyright conditions on any ROM-based product.

If you have no suitable ROM image to hand, but would nevertheless like to test your RAM unit, you can still use the accompanying RAM Loader program. But when a filename is requested, just supply the name of any file that you have to hand, providing that it is not greater than 16K in length. You should get a Successful Load message if the RAM is ok, but the machine will probably hang after this,



ROM IMAGE LOADER

```
Filename: PRNTBUF

ROM slot: 14

ROM image loaded ok

Use Ctrl-Brk to initialise

Filename: _
```

because it does not like what you have put into the RAM. Just switch off the machine to clear it.

Once your RAM is functioning correctly, there are many uses to which it may be put, as already suggested. But if you are thinking of using it to store your own machine code utilities, we would recommend Bernard Hill's recent series on sideways RAM. See Beebug Vol.5 Nos 2, 3 and 4. And of course, why stop at just one bank of RAM?

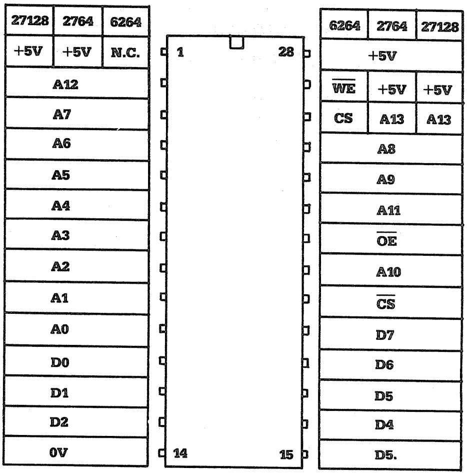
TECHNICAL NOTES

Most users will be familiar with the machine memory map (User Guide page 500), with one quarter of the memory map, from &8000 to &BFFF, allocated to sideways or paged ROMs. This means that each ROM in the machine resides at a common address, and a selection system, known as paging, is used to enable the currently valid ROM socket. The ROM selection circuitry of the Beeb is controlled by a four-bit latch (IC 76). This latch resides at the Sheila address &FE30, and is capable of defining sixteen unique values for the currently selected sideways ROM number. In the standard Beeb only two of the four bits are decoded by IC 20 giving a total of four pages, each of which contains four further possible pages. For example IC52 is normally considered to be paged ROM 15, but it can also be addressed as pages 11, 7 or 3.

Each ROM socket contains the necessary signals (address lines A0 to A13, data lines D0 to D7 and two control signals Page Select and Output Enable). Fortunately, as may be seen from Fig.5, the pin-out of the 6264 RAM is very close to that of the standard 2764 and 27128 EPROMs, which plug directly into any sideways ROM socket. The only major difference being the need to provide a control signal to enable the writing of data to the RAM.

8K SIDEWAYS RAM

As may be seen from Fig.5, there are only three pinout differences between the 6264 RAM and the 2764 EPROM: namely pins 1, 26 and 27. The socket pin 26 will normally be connected to address line A13 if the machine is set up for 27128 EPROMs (S32 and S33 west). This is incompatible with the RAM chip pinout, and although the ROM sockets can be changed in pairs to provide a suitable signal on pin 26 using S32/33, this does restrict the other socket of the pair to 8K capacity. A simpler solution, adopted on the 8K unit, is to isolate the pin from the socket and provide a suitable condition by connecting pin 26 to pin 28.



27128	2764	6264	Pin
+5V	+5V	N.C.	1
A12			
A7			
A6			
A5			
A4			
A3			
A2			
A1			
A0			
D0			
D1			
D2			
0V			14

Pin	6264	2764	27128
28	+5V		
	$\overline{WE}$	+5V	+5V
	CS	A13	A13
	A8		
	A9		
	A11		
	$\overline{OE}$		
	A10		
	$\overline{CS}$		
	D7		
	D6		
	D5		
	D4		
15	D5.		

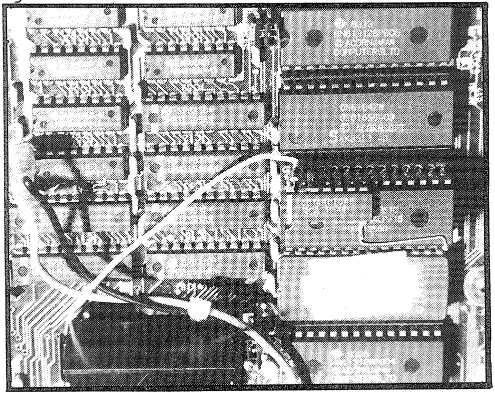
Figure 5 Pinouts of 6264 RAM, and 2764 and 27128 EPROMS.

16K SIDEWAYS RAM

To provide the RAM equivalent of a 27128 EPROM it is necessary to use two 6264 chips, as no suitable single device presently exists. The Chip Select signal on pin 26 of the RAM is used to provide selection between the two chips. One RAM device responds to addresses in the lower half of the sideways page (&8000 to &9FFF) when A13 is low, while the second device is active for A13 high (addresses &A000 to &BFFF).

Functionally this is achieved by taking the A13 line, present on pin 26 of the ROM socket, and feeding it directly to pin 26 on one RAM, then inverting it before feeding it to pin 26 on the other. See Fig.4. A suitable device for inverting the address line is the 74LS04 hex inverter, though technology buffs may prefer the lower power 74HCT04 high-speed CMOS part.

```
   10 REM Program ROM Image Loader
   20 REM Version B 0.3C
   30 REM Author  William Fulton
   40 REM Beebug  December 1986
   50 REM Program subject to copyright
   60 :
  100 MODE7:HIMEM=&3C00:DIM Q% 50
  110 FOR A=1 TO 2:PRINTTAB(7,3+A)CHR$13
 4CHR$141"ROM IMAGE LOADER":NEXT
  120 PROCassemble
  130 REPEAT:PRINT''CHR$131;"Filename:";
 CHR$134;:INPUT""F$
  140 PROCoscli("LOAD "+F$+" 3C00")
  150 REPEAT:PRINT'CHR$131;"ROM slot:";C
 HR$134;
  160 INPUT""B%:UNTIL B%>-1 AND B%<16:PR
 INT'CHR$131;"Size in K bytes (8 or 16):"
 ;CHR$134;:INPUT""S%:IFS%<>8 S%=16
  170 ?&70=0:?&71=&3C:?&72=0:?&73=&80:?&
 74=B%:?&75=S%*4:CALL copy
  180 IF FNtest PRINT'CHR$131"ROM image
 loaded ok":PRINT'CHR$134"Use Ctrl-Brk to
  initialise" ELSE VDU7:PRINT'CHR$133"Ima
 ge has not loaded"
```



```
  190 UNTIL FALSE
  200 :
 1000 DEF PROCassemble
 1010 FOR I%=0 TO 2 STEP 2:P%=&A00:[ OPT
 I%
 1020 .copy:LDA &F4:STA &76:LDA &74
 1030 STA &F4:STA &FE30:LDX #0
 1040 .cloop1:LDY #0
 1050 .cloop2:LDA (&70),Y
 1060 STA (&72),Y:INY:BNE cloop2
 1070 INX:INC &71:INC &73
 1080 CPX &75:BNE cloop1:LDA &76
 1090 STA &F4:STA &FE30:RTS
 1100 ] NEXT:ENDPROC
 1110 :
 1120 DEFFNtest
 1130 Y%=B%:?&F7=&80
 1140 FOR C%=0 TO 3
 1150 ?&F6=C%:C%?&77=USR(&FFB9)
 1160 NEXT
 1170 =(!&3C00=!&77)
 1180 :
 1190 DEFPROCoscli(Q$)
 1200 $Q%=Q$:A%=0:X%=Q%MOD256:Y%=Q%DIV25
6
 1210 Q=USR(&FFF7):ENDPROC
```

SIDEWAYS RAM OFFER

To assist members who wish to construct their own sideways RAM module as described in the accompanying article we have put together a complete kit of all the parts you will need - a suitable soldering iron and solder are the only extras. The list of parts is as follows:

Quantity	Description
2	6264 8K staic RAM
1	IC 74LS04 logic chip
1	Carrier board
1	10k Ohm resistor 1/8W
1	Connector
1	Length of wire

The complete set of parts as above is available for just £9.95 post free direct from BEEBUG.

In addition, for those who do not wish to construct the RAM module themselves, we are able to supply ready-made units at post-free price of just £13.95.

In both cases send your order to BEEBUG RAM Offer, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX. If paying by credit card please quote your card number and its expiry date.

COMPUTER AIDED DESIGN

Geoff Bains, always quick on the draw, compares the latest CAD packages to reach the Beeb market.

A Computer Aided Design (CAD) package superficially appears similar to many drawing packages found on the Beeb. However, a CAD package stores information about each mark on the screen rather than relying on the screen data itself. For example, a straight line would be stored as two sets of co-ordinates together with a code to distinguish the type of line used. Simple art packages just produce the marks on the screen and then save the screen RAM.

A Beeb CAD package may not compete with a professional system designed to run on a much more expensive system. However, for a small outlay a CAD package on a BBC micro is ideal for tasks such as designing circuit boards, and many others.

A good CAD package has several features which distinguish it from those of lesser quality. Accuracy is important, particularly if the package allows parts of drawings to be magnified. In addition, a CAD package should be flexible and easy to use.

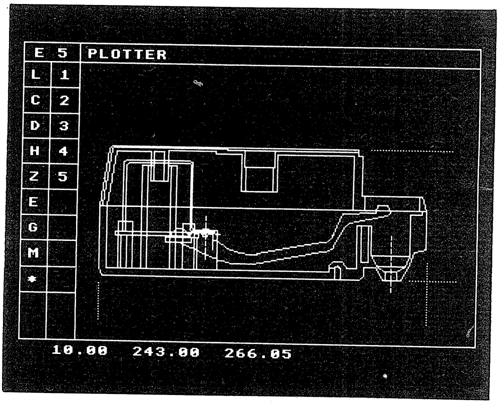
We will look here, at three CAD packages for the Beeb, one quite recent introduction (from Technomatic), and two aimed particularly at the educational market (Educad and Micad).

Title : **Micad**
Supplier : **Heinemann Educational Books Ltd.**
22 Bedford Square,
London WC1B 3HH.
Tel. 01-637 3311
Price : **£37.95**

Micad is the oldest (1984) of the software reviewed here, and is supplied on disc. It is really two programs in one; Micad 2D is used for conventional CAD work whereas Micad 3D is used for 3D wire frame images. Micad 3D is not really a 'CAD' package at all, but it is a nice inclusion for the exploration of technical drawing.

All the programs that go to make up Micad (except the printout routines) are written in Basic. This makes the whole package rather slow and flickery on the mode 4 screen used.

The different drawing options of Micad 2D are selected with the cursor from a menu of single letter commands down the side of the screen. Each command is followed by a number, again selected with the cursor. This process is very tedious. If single letters and numbers are to be used, it is much easier and quicker to enter them from the keyboard.



10.00 243.00 266.05

Micad has commands for drawing straight lines and circles (full or dotted), and hatching (shading) areas with one of five patterns. Text can also be added to a diagram and parts of the screen 'zoomed' for detailed work.

For editing a Micad 2D drawing a separate editing program is entered. This allows any point, line or arc to be moved or deleted. Again, the unfriendly nature of the menu system and the slow operation of Micad makes editing a process to avoid whenever possible.

Another program module allows the easy addition of linear and angular dimensions to the diagram. This is a useful feature and the only really professional facility of the package.

Once a diagram has been created and edited to your satisfaction it can be printed out. Micad can cope with both simple screen dumps to Epson compatible printers, and plots to Penman and Hewlett-Packard plotters. Although the manual doesn't cover it, it would be reasonably easy to link in your own screen dump routine.

Micad 3D is similar to 2D in operation, although movement and drawing is now in three dimensions, projected on the screen. The direction of view of the object is selected from preset directions on the function keys or keyed in as three-dimensional co-ordinates. A useful feature is the 'extrusion' facility which copies a shape in one plane to another, and joins the vertices to make a solid.

The 3D program uses single key commands from the keyboard - much better than 2D's menus. However, 3D is limited and the editing crude. It is easily outshone by Interactive 3D or 3D Zicon reviewed in BEEBUG Vol.4 No.10 and Vol.4 No.9 respectively.

The whole Micad package seems a strange combination of different ideas bundled together without much thought as to continuity or actual use.

Title : **Educad**
Supplier : **Edusoft,**
15 Tamworth Business Centre,
Amber Close, Amington,
Tamworth, Staffs B77 4DS.
Tel. (05436) 76939
Price : **£92.00**

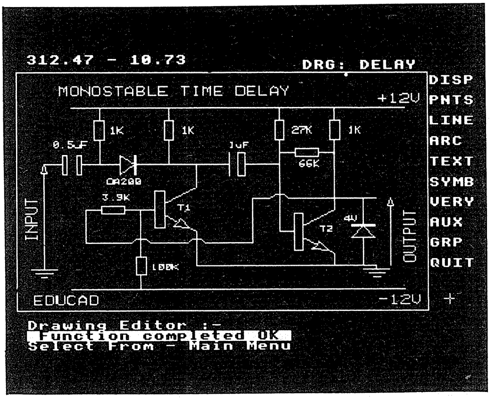
Although Educad is designed primarily as an introduction to CAD principles for schools, it has a decidedly home-made quality. The software is supplied on disc with a very poor 50 page manual. This lacks any useful structure and, for a product aimed at the educational market, this is a disaster.

Once the manual has been digested sufficiently to start using the package (it takes 6 pages to find CHAIN"EDUCAD" is needed), the Educad package is little better than Micad. Most of the software is in Basic, and again it is slow.

Educad pictures are produced in mode 4. The cursor is moved with either the cursor keys or joystick. The cursor accelerates when a cursor key is kept pressed down. This makes long movements fast but still allows delicate manoeuvring.

Like Micad, the diagram drawing process is menu-driven and suffers accordingly. There are levels of sub-menu as well, and it is quite easy to get lost in the menu structure.

Educad produces the requisite straight lines and arcs based on construction points and lines on the diagram which can be displayed or hidden as required. A large range of options for specific operations are catered for (e.g. drawing a line as a tangent to an arc), and all work on the diagram is automatically aligned with a (variable) grid to make the final drawing more accurate. For delicate work, a zoom facility is also provided, but the grid alignment makes work difficult at high magnifications.



Drawing Editor :-
Function completed OK
Select From - Main Menu

Menus aside, editing is reasonably easy. The cursor is moved over an element and the command to delete that element type issued. The software works out how to remove it from the drawing store.

Text of any size can be added. This is not just Beeb text but letters drawn out on the screen. Similarly, a range of preset symbols such as electronics components can be repeatedly positioned on a diagram. A character definer is supplied to produce your own fonts and symbols.

For computer aided manufacturing (CAM) facilities, Edusoft also have a 'EduCNC' package to take designs created with Educad, and generate data suitable for a CNC (Computerized Numeric Control) machine to actually make objects.

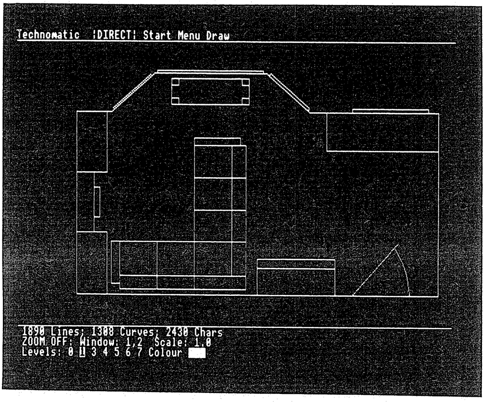
Educad uses a much more professional approach to producing technical drawings than does Micad. However, it costs over twice the price. For nearly £100 the quality of presentation of this program is well below standard.

Title : **Novacad**
Supplier : **Technomatic,**
17 Burnley Road,
London NW10 1ED.
Tel. 01-208 1177
Price : **£90.85.**

Novacad is also expensive. It requires both a 16K ROM and a disc. Novacad is the best of the bunch, both in terms of its ease of use and its flexibility.

Drawings can be in modes 0, 1, or 4 depending on the memory available. Novacad can use B+ and Master shadow memory and an Aries board.

The screen shows only a twentieth of a complete diagram at a time. It is a window onto the complete picture. You can move between windows by pressing the Ctrl key along with a cursor direction key.



```
Technomatic  |DIRECT| Start Menu Draw
```

```
1890 Lines; 1308 Curves; 2430 Chars
ZOOM OFF: Window: 1,2  Scale: 1.0
Levels: 0 1 3 4 5 6 7 Colour ■
```

The Novacad cursor is not the small 'x' of the other packages but a horizontal and vertical line running right across the screen, as used in most professional systems. This is moved with the cursor keys in large steps and can be slowed down with the Shift Lock key for fine work. In the mouse version of Novacad, control is much finer all the time.

There are eight 'levels' of a drawing, which can be displayed or hidden as required. The other levels can be used for construction lines or for, say, the reverse side design for a circuit board.

All the picture elements - line, rectangle and arc - are easily produced with a few easy-to-remember single-key commands. A useful feature is the recall of the last element at another position with the Copy key.

Novacad also has a zoom facility, but the most useful feature is the 'icon' system. Predefined sections of a diagram can be called up and deposited at any size and orientated four ways onto the screen. Even more useful, a section of the screen can be captured as a icon, stored to disc and then re-used as often as needed.

Text can be added to the diagram. Although this is only in the BBC micro's normal font and only in one size, the characters can be rotated.

As with Educad, deleting lines is a question of pointing at them with the cursor. However, Novacad uses just the Delete key to do this without the long menu sequences of Educad.

The one failing of Novacad is its production of hard copy. Only a simple screen dump is provided for dot matrix printers, but Novacad can drive a plotter to provide drawings of a high quality for those with access to such devices. This aside, Novacad is excellent. It is flexible, genuinely easy to operate without much practice, and a pleasure to use.

If money is of prime importance then Micad must be your choice. It is considerably cheaper than the others. However, you'd probably be better off with a good art package. For all the genuine CAD facilities you are likely to see or want on a BBC micro Novacad performs excellently, albeit for a high price.

BUSINESS GRAPHICS (Part 2)

Alan Dickinson adds a powerful screen editor to his business graphics program. Controlling and changing your charts and graphs is now simplicity.

I forget who coined the term "WIMPS" but they meant it in jest, implying that Windows, Icons, Menus, and Pointers were necessary for the wimps who couldn't cope with the traditional user interfaces, obscure and unfriendly as they were. The term seemed to stick, and fortunately so have easy-to-use interfaces.

Last month we published a program to produce high-quality graphs on the BBC micro. This month we are building on that program by adding a powerful menu-driven data editing system. This replaces the DATA statements of last month's program so that all aspects of the data display, including editing of the data, may be managed quickly and easily. It then becomes simplicity itself to modify some aspect of a display and rapidly re-draw the screen.

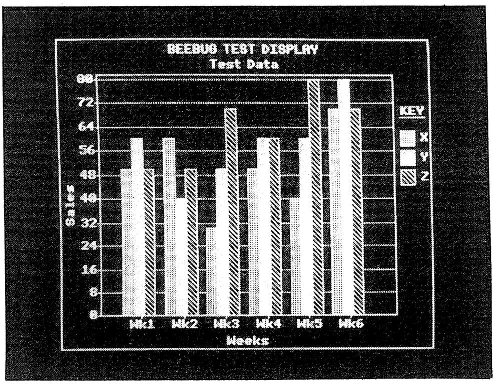
ADDING THE EDITOR

To add the editor to the original program, first type in this month's listing and save as for a normal program (just for safety) under the name GPROG2. Make sure, when entering the program, that you do not add any additional spaces, but do omit the space after each line number in the listing (included for readability), as the program is very tight on space. Basic I users should replace OPENUP by OPENIN in line 171Ø. Now, with the program still in memory, proceed as follows:

```
    *SPOOL TEMP    <Return>
    LIST           <Return>
    *SPOOL         <Return>
```

Now load last month's program:

```
    LOAD"GPROG1"   <Return>
    *EXEC TEMP     <Return>
    SAVE"GPROG"    <Return>
```



The editor has now been appended to the original program and the new version saved as GPROG ready for use. This process becomes almost ridiculously easy if you use the Part-Merge/Save Utility from the October issue of BEEBUG (Vol.5 No.5).

USING THE EDITOR

When you now run the Business Graphics program, you will be presented with a series of menus with up to 13 choices on any one menu. All the menus work in the same way. A pointer can be moved to any menu item using the cursor keys, and then selected by pressing the Return key, or you can type in the first letter of the desired option. When specific values must be input, such as the chart title, an input window is opened in the bottom right-hand corner of the screen. The required information is then entered, and terminated by pressing Return. With any menu, you may press the "*" key and open up a *-command window. This enables commands such as *CAT to be executed. The function keys could also be programmed to enter any repetitive sequence of menu selections, and can indeed be re-programmed from within the program using *KEY.

With all menus, pressing Escape will return you to the previous level of menu, while the key combination Ctrl/@ has been programmed to generate a true Escape condition. This returns to the highest level menu to provide the option of quitting the program. When first run, the program displays a menu with the options:

```
    File        Edit
    Display     Clear
```

We will look at the use of each of these in turn.

THE FILE OPTION

Four choices are provided, but only two of them, SAVE and LOAD, are available at this stage. These allow all the settings, including data, that you have entered via the editor, to be saved and re-loaded as required. The other two options, IMPORT and DUMP, will be described next month.

THE EDIT OPTION

This is the main choice and leads to a further menu allowing all aspects of the display to be entered and edited as required. Further sub-menus also appear in many instances. The Edit options (and sub-menus) are as follows:

Titles:	Title	Edit main title
	Sub-Title	Edit sub-title
	X-axis	Edit X-axis legend
	Y-axis	Edit Y-axis legend
Columns:	Legends	Edit column names
	Switches	Select which columns to display
Variables:	Legends	Edit variable names
	Patterns	Select pattern for variable (range Ø to 255)
	Switches	Select which variables to display
Numerical Data:		
	Colwise	Enter data for all variables, column by column
	Varwise	Enter data for all columns, variable by variable

Format of graph:

Stack	Cluster	Bar
Range	ThreeD	Line

Grid: Select Grid or No Grid

Display: Display selected graph

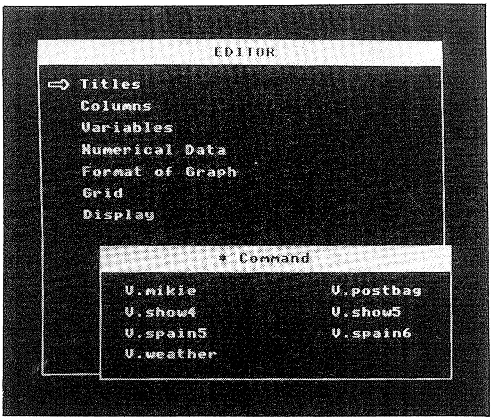
DISPLAY AND CLEAR OPTIONS

The main menu (and the Edit menu) both provide a Display option for convenience. The Clear option clears all current data and other settings.

GENERAL COMMENTS

You should read the above notes, and experiment thoroughly with the use of the editor, in conjunction with the description contained in part one published last month. All the graph facilities are exactly as first described, but are now controlled from screen menus. Remember that any chart display may have up to 13 columns, and each column may contain up to 7 variables. Which columns and variables are displayed are determined by the 'switch' settings (see above). Experimenting with all the options available will help you to make the most of the program's power and flexibility.

Next month, the final part of this series will deal with the two remaining graph options, 3D charts and pie charts, and provide a means for data to be imported to the Business Graphics program from other data files.



```
   1Ø REM Program GPROG2
   2Ø REM Version B1.2
   3Ø REM Author Alan Dickinson
   4Ø REM BEEBUG December 1986
   5Ø REM Program subject to copyright
   6Ø :
  1ØØ MODE4:PROCzi
  11Ø VDU23,24Ø,&88F8;&8888;&F8;Ø;23,241
,&D87Ø;&D888;&7Ø;Ø;23,242,&7Ø2Ø;&7ØF8;&2
Ø;Ø;23,243,&F8F8;&F8F8;&F8;Ø;
  12Ø DIMC$(12),C%(12),V$(6),V%(6),H%(6)
,N(12,6):p$=STRING$(2Ø," "):PROCclr2
  13Ø ONERRORPROCze
  14Ø REPEAT:REPEAT:Q%=FNzm("BEEBUG GRAP
H MODULE","File~Edit~Display~Clear",Ø)
  15Ø IFQ%=ØPROCfiles:ELSEIFQ%=1PROCedit
s:ELSEIFQ%=2PROCdis:A%=GET:ELSEIFQ%=3PRO
Cclr
  16Ø UNTILQ%<Ø
  17Ø UNTILFNzm("QUIT?","Yes~No",Ø)=Ø
  18Ø *FX4
  19Ø *FX22Ø,27
  2ØØ MODE7:END
  21Ø :
 1ØØØ DEFPROCfiles:LOCALQ%:Q%=FNzm("FILI
NG SYSTEM","Load~Save~Import~Display Dum
p",Ø)
 1Ø1Ø IFQ%=ØPROCload:ELSEIFQ%=1PROCsave:
ELSEIFQ%=2PROCimp:ELSEIFQ%=3PROCdump
 1Ø2Ø ENDPROC
 1Ø3Ø DEFPROCedits:LOCALQ%
 1Ø4Ø REPEAT:Q%=FNzm("EDITOR","Titles~Co
lumns~Variables~Numerical Data~Format of
 Graph~Grid~Display",Q%)
 1Ø5Ø IFQ%=ØPROCedtitles:ELSEIFQ%=1PROCe
dxdivs:ELSEIFQ%=2PROCedvars:ELSEIFQ%=3PR
OCeddata:ELSEIFQ%=4PROCedtype:ELSEIFQ%=5
PROCedgrid:ELSEIFQ%=6PROCdis:A%=GET
 1Ø6Ø UNTILQ%<Ø:ENDPROC
 1Ø7Ø DEFPROCedtitles:LOCALQ%
 1Ø8Ø REPEAT:Q%=FNzm("TITLES","Title~Sub
-Title~X-axis~Y-axis",Q%)
 1Ø9Ø IFQ%>=Ø$(&98Ø+Q%*32)=FNzc("Text",$
(&98Ø+Q%*32),32):Q%=Q%+1
 11ØØ UNTILQ%<Ø:ENDPROC
 111Ø DEFPROCedxdivs:LOCALQ%
 112Ø REPEAT:Q%=FNzm("COLUMNS","Legends~
Switches",Q%)
 113Ø IFQ%=ØPROCedxlegs
 114Ø IFQ%=1PROCedxsw
 115Ø UNTILQ%<Ø:ENDPROC
 116Ø DEFPROCedxlegs:LOCALQ%:REPEAT:PROC
atx:Q%=FNzm("LEGENDS",Z$,Q%)
 117Ø IFQ%>=ØC$(Q%)=FNzc("Column",C$(Q%)
,8):Q%=Q%+1
 118Ø UNTILQ%<Ø:ENDPROC
 119Ø DEFPROCedxsw:LOCALQ%:REPEAT:PROCat
x:Q%=FNzm("SWITCHES",Z$,Q%)
 12ØØ IFQ%>=ØIFC%(Q%)=ØC%(Q%)=1ELSEIFQ%>
=ØC%(Q%)=Ø:Q%=Q%+1
 121Ø UNTILQ%<Ø:ENDPROC
 122Ø DEFPROCatx:LOCALJ%:Z$="":FORJ%=ØTO
12:Z$=Z$+C$(J%)+"  ("
 123Ø IFC%(J%)>ØZ$=Z$+"ON)~"ELSEZ$=Z$+"O
FF)~"
 124Ø NEXT:ENDPROC
 125Ø DEFPROCedvars:LOCALQ%:REPEAT
 126Ø Q%=FNzm("VARIABLES","Legends~Patte
rns~Switches",Q%)
 127Ø IFQ%=ØPROCedvlegs:ELSEIFQ%=1PROCed
vpat:ELSEIFQ%=2PROCedvsw
 128Ø UNTILQ%<Ø:ENDPROC
 129Ø DEFPROCedvlegs:LOCALQ%:REPEAT:PROC
atv:Q%=FNzm("LEGENDS",Z$,Q%)
 13ØØ IFQ%>=ØV$(Q%)=FNzc("Variable",V$(Q
%),9):Q%=Q%+1
 131Ø UNTILQ%<Ø:ENDPROC
 132Ø DEFPROCedvpat:LOCALQ%:REPEAT:PROCa
tv:Q%=FNzm("PATTERNS",Z$,Q%):IFQ%>=ØH%(Q
%)=FNzn("Pattern",H%(Q%),Ø,255):Q%=Q%+1
 133Ø UNTILQ%<Ø:ENDPROC
 134Ø DEFPROCedvsw:LOCALQ%:REPEAT
 135Ø PROCatv:Q%=FNzm("SWITCHES",Z$,Q%)
 136Ø IFQ%>=ØIFV%(Q%)=ØV%(Q%)=1ELSEIFQ%>
=ØV%(Q%)=Ø:Q%=Q%+1
 137Ø UNTILQ%<Ø:ENDPROC
 138Ø DEFPROCatv:LOCALJ%:Z$=""
 139Ø FORJ%=ØTO6:Z$=Z$+V$(J%)+"  ("
 14ØØ IFV%(J%)>ØZ$=Z$+"ON"ELSEZ$=Z$+"OFF
"
 141Ø Z$=Z$+" p="+STR$(H%(J%))+")~"
 142Ø NEXT:ENDPROC
 143Ø DEFPROCeddata:LOCALQ%
 144Ø Q%=FNzm("DATA","Colwise~Varwise",Ø
):IFQ%=ØPROCedncol(-1)ELSEIFQ%=1PROCednv
ar(-1)
 145Ø ENDPROC
 146Ø DEFPROCedncol(v%):LOCALJ%,Q%
 147Ø REPEAT:IFv%=-1PROCatx:I$=Z$:ELSEI$
="":FORJ%=ØTO12:I$=I$+C$(J%)+" = "+STR$(
N(J%,v%))+"~":NEXT
 148Ø Q%=FNzm("COLUMN",I$,Q%)
 149Ø IFQ%>=ØANDv%=-1PROCednvar(Q%)
 15ØØ IFQ%>=ØANDv%>-1PROCeddata2(Q%,v%):
Q%=Q%+1
 151Ø UNTILQ%<Ø:ENDPROC
 152Ø DEFPROCednvar(c%):LOCALQ%,J%
 153Ø REPEAT:IFc%=-1:PROCatv:I$=Z$:ELSEI
$="":FORJ%=ØTO6:I$=I$+V$(J%)+" = "+STR$(
N(c%,J%))+"~":NEXT
 154Ø Q%=FNzm("VARIABLE",I$,Q%)
 155Ø IFQ%>=ØANDc%=-1PROCedncol(Q%)
 156Ø IFQ%>=ØANDc%>-1PROCeddata2(c%,Q%):
Q%=Q%+1
 157Ø UNTILQ%<Ø:ENDPROC
 158Ø DEFPROCeddata2(c%,v%)
 159Ø N(c%,v%)=FNzn("VALUE",N(c%,v%),-99
99999,9999999):V%(v%)=1:C%(c%)=1
 16ØØ ENDPROC
 161Ø DEFPROCedtype:LOCALQ%:Q%=FNzm("FOR
MAT","Stack~Cluster~Bar~Range~ThreeD~Lin
e~Pie",Z%):IFQ%>=ØZ%=Q%
 162Ø ENDPROC
 163Ø DEFPROCedgrid:LOCALQ%
 164Ø Q%=FNzm("GRID","No Grid~Grid",G%+1
):IFQ%>=ØG%=Q%
 165Ø ENDPROC
 166Ø DEFPROCclr:IFFNzm("CLEAR ?","Yes~N
o",Ø)=ØPROCclr2
 167Ø ENDPROC
 168Ø DEFPROCclr2:LOCALJ%:$&98Ø="":$&9AØ
="":$&9CØ="":$&9EØ="":F$="":f$="":D$=""
 169Ø FORJ%=ØTO12:C$(J%)="        ":C$(J
%)="":C%(J%)=Ø:NEXT:FORJ%=ØTO6:V$(J%)=ST
RING$(9," "):V$(J%)="":V%(J%)=Ø:H%(J%)=J
%:NEXT
 17ØØ Z%=Ø:G%=1:ENDPROC
 171Ø DEFPROCload:F$=FNzc("FILENAME",F$,
8):IFF$=""ENDPROC:ELSEA%=OPENUP(F$)
 172Ø INPUT#A%,$&98Ø,$&9AØ,$&9CØ,$&9EØ,Z
%,G%
```

```
 173Ø FORJ%=ØTO12:INPUT#A%,C$(J%),C%(J%)
:NEXT:FORJ%=ØTO6:INPUT#A%,V$(J%),V%(J%),
H%(J%):NEXT
 174Ø FORJ%=ØTO6:FORK%=ØTO12:INPUT#A%,N(
K%,J%):NEXT:NEXT
 175Ø CLOSE#A%:ENDPROC
 176Ø DEFPROCsave:F$=FNzc("FILENAME",F$,
8):A%=OPENOUT(F$):PRINT#A%,$&98Ø,$&9AØ,$
&9CØ,$&9EØ,Z%,G%
 177Ø FORJ%=ØTO12:PRINT#A%,C$(J%),C%(J%)
:NEXT:FORJ%=ØTO6:PRINT#A%,V$(J%),V%(J%),
H%(J%):NEXT
 178Ø FORJ%=ØTO6:FORK%=ØTO12:PRINT#A%,N(
K%,J%):NEXT:NEXT
 179Ø CLOSE#A%:ENDPROC
 5ØØØ DEFPROCzi:*FX4,1
 5ØØ1 Z$=STRING$(255," "):I$=Z$:Q$=Z$
 5Ø1Ø *FX22Ø,Ø
 5Ø2Ø VDU23,255,&2C38;&Ø3E6;&E6Ø3;&382C;
23,254,Ø;&8ØFF;&FF8Ø;Ø;23,253,&C3ØØ;&FF;
Ø;Ø;23,Ø,&2ØØA;Ø;Ø;Ø;:ENDPROC
 5Ø3Ø DEFPROCze:PROCzw(32,"ERROR "+STR$(
ERR)):REPORT:PRINT''"Press ESCAPE to exi
t"''"Press RETURN to continue":A%=GET
 5Ø4Ø *FX22Ø,27
 5Ø5Ø *FX4
 5Ø6Ø IFA%=27VDU26,4:CLS:REPORT:PRINT" @
 ";ERL:END
 5Ø7Ø *FX22Ø,Ø
 5Ø8Ø *FX4,1
 5Ø9Ø ENDPROC
 51ØØ DEFPROCzo:LOCALX%,Y%:$&9ØØ=FNzc("*
 Command","",32):CLS:VDU14:X%=Ø:Y%=9:CAL
L&FFF7:VDU15:ENDPROC
 511Ø DEFPROCzw(Z%,Z$):LOCALX%,Y%:X%=127
1-(Z%+2)*32:Y%=324:VDU24,X%;Y%;1279;Y%+7
6;:GCOLØ,129:CLG:VDU26:GCOLØ,1:MOVEX%,Y%
:DRAWX%,Ø:DRAW1279,Ø:DRAW1279,Y%
 512Ø VDU24,X%+4;4;1275;324;:GCOLØ,128:C
LG:VDU28,38-Z%,21,38,2Ø:COLOURØ:COLOUR12
9
 513Ø PRINTTAB((Z%-LEN(Z$))/2,Ø)Z$;:VDU2
8,38-Z%,3Ø,38,22:COLOUR1:COLOUR128:CLS:E
NDPROC
 514Ø DEFFNzm(T$,Q$,D%):LOCALI$,I%,J%,K%
 515Ø VDU26:COLOUR128:CLS:VDU24,Ø;94Ø;12
59;1Ø15;:GCOLØ,129:CLG:VDU26,4:GCOLØ,1:M
OVEØ,96Ø:DRAWØ,16:DRAW1259,16:DRAW1259,9
6Ø
 516Ø COLOURØ:COLOUR129:PRINTTAB(2Ø-LEN(
T$)/2,1);T$;:COLOUR1:COLOUR128:VDU28,1,3
1,38,4:J%=-1
 517Ø REPEAT:K%=INSTR(Q$,"~"):IFK%=ØK%=L
EN(Q$)+1
 518Ø J%=J%+1:VDU31,Ø,J%*2,32,32,32:PRIN
TLEFT$(Q$,K%-1);:I$=I$+LEFT$(Q$,1):IFK%<
LEN(Q$)Q$=MID$(Q$,K%+1)ELSEQ$=""
 519Ø UNTILQ$="":K%=D%:IFK%>J%K%=K%MOD(J
%+1)
 52ØØ VDU31,Ø,K%*2,&FFFE;:REPEAT:I%=GET:
IFI%>96ANDI%<123I%=I%AND223
 521Ø VDU31,Ø,K%*2,32,32
 522Ø IFI%=ASC"*"PROCzo:VDU28,1,31,38,4
 523Ø IFI%=139ANDK%>ØK%=K%-1
 524Ø IFI%=138ANDK%<J%K%=K%+1
 525Ø IFINSTR(I$,CHR$(I%))>ØK%=INSTR(I$,
CHR$(I%))-1:I%=13
 526Ø VDU31,Ø,K%*2,&FFFE;:UNTILI%=13ORI%
=27:IFI%=13THEN=K%ELSE=-1
 527Ø DEFFNzr(D$,I$,W%,P%):$&9ØØ=I$+STRI
NG$(W%," "):PRINTTAB(Ø,Ø)LEFT$($&9ØØ,W%)
;
 528Ø REPEAT:VDU31,P%,1,253:A%=GET
 529Ø VDU31,P%,1,32:PROCzk:VDU31,P%,1,25
3
 53ØØ UNTILA%=13ORA%=27
 531Ø FORP%=&9ØØ+W%TO&9ØØSTEP-1
 532Ø IF?P%=32OR?P%=13?P%=13:ELSEP%=Ø
 533Ø NEXT:IFA%=27THEN=I$ELSE=$&9ØØ
 534Ø DEFPROCzk
 535Ø IFA%=13ORA%=27ENDPROC
 536Ø IFA%=136ANDP%>ØP%=P%-1:ENDPROC
 537Ø IFA%=137ANDP%<W%-1P%=P%+1:ENDPROC
 538Ø IF(A%>135ANDA%<14Ø)OR(A%<26)ENDPRO
C
 539Ø IFA%=127$(&9ØØ+P%)=$(&9Ø1+P%):PRIN
TTAB(Ø,Ø)$&9ØØ;" ":ENDPROC
 54ØØ IFA%=135$(&9ØØ+P%)=" "+$(&9ØØ+P%):
?(&9Ø1+W%)=13:PRINTTAB(Ø,Ø)$&9ØØ;:ENDPRO
C
 541Ø IFA%=126A%=253
 542Ø ?(&9ØØ+P%)=A%:VDU31,P%,Ø,A%:P%=P%+
1:IFP%>=W%P%=P%-1
 543Ø ENDPROC
 544Ø DEFFNzc(D$,I$,W%):PROCzw(W%,D$):=F
Nzr(D$,I$,W%,Ø)
 545Ø DEFFNzn(D$,V,L,H):LOCALV$,P%,S%,D%
,D
 546Ø PROCzw(24,D$):$&9ØØ=STR$(V):P%=Ø
 547Ø $&9ØØ=FNzr(D$,$&9ØØ,12,P%):CLS:PRI
NT$&9ØØ:S%=1:V=Ø:P%=Ø:D%=Ø:D=1
 548Ø IF?(&9ØØ+P%)=ASC"-"S%=-1:P%=1
 549Ø IF?(&9ØØ+P%)=13THENGOTO554Ø
 55ØØ IF?(&9ØØ+P%)=ASC"."ANDD%=ØD%=1:P%=
P%+1:GOTO549Ø
 551Ø IF?(&9ØØ+P%)<48OR?(&9ØØ+P%)>57PRIN
TTAB(Ø,3)"Non-numeric value";:GOTO547Ø
 552Ø IFD%=ØV=V*1Ø+?(&9ØØ+P%)-48:P%=P%+1
:GOTO549Ø
 553Ø D=D/1Ø:V=V+(?(&9ØØ+P%)-48)*D:P%=P%
+1:GOTO549Ø
 554Ø V=V*S%:IFV<L ORV>H PRINTTAB(Ø,3)"O
UT-OF-RANGE"'" ";L'" ";H:P%=Ø:GOTO547Ø
 555Ø =V
```

Sorting Data Files

If sorting disc files takes hours, find out how Jan Stuurman's program reduces this to minutes. And it can be used with the Filer database program.

Sorting data stored in data files on disc can be a very time-consuming operation unless the method used is carefully thought out, as users of the BEEBUG Filer database program may already have found to their cost. In this article I want to discuss the principles of fast file sorting, and show how these can be implemented to provide a vastly improved sort program for use with BEEBUG Filer.

PRINCIPLES OF GOOD FILE SORTING

The first point to recognise is that accessing discs to read and write records in a file is actually quite time-consuming compared with the Beeb's internal rate of working. Thus it makes sense to minimise this aspect by performing as much of the sorting as possible in main memory. This is what the program listed here does in contrast to Filer which performs all its sorting on disc.

Secondly, any sort can be speeded up by concentrating on the 'key' field alone rather than complete records. By 'key' we mean that part of each record (often just a few characters) which is to be the basis of the sort. The technique is to read each record in turn, extract from it the key, and save in memory just the key and a simple reference or pointer to the whole record on disc. This reference is often called a 'tag' and a sort which uses this idea is called a 'tag' sort.

Finally, we must employ an efficient sort algorithm to sort the key fields in memory. The re-ordered keys (with their pointers) can then be used to read each record from the original file, using direct access, and write it out to a new file in the new order. There is no single 'best' sort technique, as the efficiency of any sorting algorithm is a function of the initial degree and form of disorder of the file to be sorted. The Shell sort employed here is well known, and together with the so-called 'Quick' sort is generally considered to be one of the fastest sorts around.

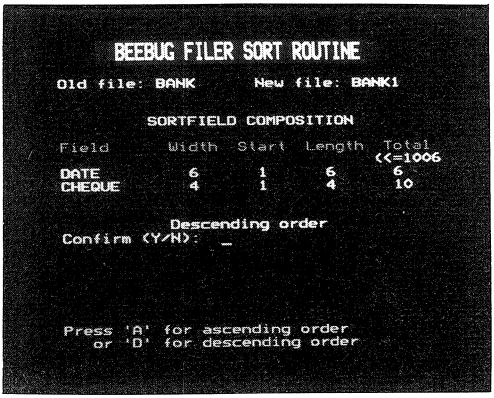
There are two limitations on the whole process; it must be possible to fit all the keys and pointers into available memory (though this is not insuperable, by splitting, sorting and merging longer files), and there must be sufficient disc space for both the original and re-ordered files, though these can be on different discs or surfaces. These principles are embodied in the Filer Sort program below.

THE FILER SORT PROGRAM

The program listed with this article, FILERS, is designed to sort any Filer database file. The routines could also be adapted to form a sort program that will equally well deal with other data files.

When typing the program in from the printed listing, omit all but essential spaces to maximise the amount of free memory. Basic I users should also replace OPENUP by OPENIN at lines 8Ø2Ø, 9Ø5Ø and 11Ø2Ø.

To make the most of the memory available, the program should be run with PAGE set to &14ØØ (as with the other Filer programs) unless running on a Master or Compact where PAGE is at &EØØ anyway, and it will always reserve as much memory as possible above HIMEM and below the mode 7 screen (&7CØØ).



```
            BEEBUG FILER SORT ROUTINE

  Old file: BANK        New file: BANK1

             SORTFIELD COMPOSITION

  Field       Width  Start  Length  Total
                                    <<=1006
  DATE          6      1      6      6
  CHEQUE        4      1      4      10


              Descending order
  Confirm (Y/N):  _




  Press 'A' for ascending order
     or 'D' for descending order
```

After both old and new filenames have been entered (the new file is "overwrite protected"), up to 6 data fields of the original file may be used to create a set of keys in memory ready to be sorted.

Enter the name of each key field in turn, and either the entire field (by pressing Return) or just part of the field (enter start position and length) can be used to build up a complete 'sort field' for the file. The total length of the sort field depends on the memory available (above HIMEM) and the number of records on file. For example, with PAGE=&1400 and 300 records, the maximum length of the sort field is 64. The program displays this maximum length, keeps a running total length, and prints an error message if the sort field becomes too large.

To some extent, the length of the sort field will affect the time taken to sort the file, so it makes sense to keep this as short as possible. You might allow up to 20 characters for a 'NAME' field, but probably the first six or so characters would be sufficient to sort the file by NAME alphabetically.

When sorting on a single field, the user may choose between an alphabetic or numeric sort. Numeric sorts on more than one field can only be accomplished by successively sorting the file for each field separately, starting with the field of lowest priority. The sorting algorithm used is stable when sorting in ascending order: it preserves the relative order of records with equal keys. Note: a numeric sort will not work correctly if the numbers have been written in scientific notation (e.g. 1.22E-3).

Finally, the user has to specify the order, either descending or ascending, in which the records should be sorted. Note that in this context, descending means starting at the highest (or smallest) and descending to the lowest (or largest) - this would be equivalent to a normal alphabetical ordering, for example.

The procedure, PROCsort, performs the actual sorting in three stages, each indicated in turn on the screen.

1. Read the old file from disc and, using the sort field information given by the user, create a tag file in memory. A three-figure (001-999) record pointer is appended to each entry in the tag file.

2. Sort the file in memory using the Shell Sort algorithm (see D.E.Knuth, The Art of Computer Programming, Volume 3, published by Addison-Wesley). This algorithm has an acceptable running time which does not depend very much on the initial ordering of the file.

3. Using the pointers that were affixed to the records of the tag file, records are read in from the old file (on disc) and written to the new file in the new order.

The program can sort BEEBUG Filer data files of up to 999 records. The following timings in seconds, using an Opus Challenger disc unit, will give some indication of the speeds that can be achieved. Total sort times are given in the last column.

No of records	Read file (secs)	Sort (secs)	Writing new file (secs)	Total time (mins)
100	10	16	48	1:14
200	20	38	100	2:38
400	43	83	217	5:43
800	83	216	519	13:38

PROGRAM NOTES

Lack of space precludes more than a brief description of the program this month. However, PROCfilenames deals initially with the file specifications, reading the file description record (FDR) from the old file and creating a new file of the right length and FDR. PROCsortfield then obtains the details of the sort keys to be used, storing the information in 3 arrays sfld%, strt% and len%.

The main procedure is PROCsort. This uses the details already collected to form a template (s$ at line 11030) which with the aid of the EVAL function extracts the relevant parts of each record as it is read into memory. The sort specification is also constructed as a string (t$ at line 11120) and EVALuated to compare keys. Finally, records are read from the old file in the revised order and written out to the new file.

Next month we will provide a detailed description of this useful program so that it can be readily modified and used to sort other data files that you may have stored on disc.

```
   1Ø REM PROGRAM FILERS
   2Ø REM VERSION B.2
   3Ø REM AUTHOR Jan Stuurman
   4Ø REM BEEBUG Dec 1986
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
  1ØØ MODE7:HIMEM=PAGE+&18ØØ
  11Ø ONERRORPROCerror:END
  12Ø PROCsetup
  13Ø PROCtitle
  14Ø PROCinput
  15Ø PROCsort
  16Ø PROCclose
  17Ø VDU26:END
  18Ø :
 1ØØØ DEFPROCsetup
 1Ø1Ø rec=1:FDR=256
 1Ø2Ø B%=HIMEM:maxf=12:maxs=6
 1Ø3Ø DIMrecord$(maxf),field$(maxf),widt
h%(maxf),os 22
 1Ø4Ø DIMsfld%(maxs),strt%(maxs),len%(ma
xs)
 1Ø5Ø ENDPROC
 1Ø6Ø :
 2ØØØ DEFPROCtitle
 2Ø1Ø FORI=ØTO1:PRINTTAB(3,I)CHR$141CHR$
129CHR$157CHR$131"BEEBUG FILER SORT ROUT
INE  "CHR$156:NEXT
 2Ø2Ø ENDPROC
 2Ø3Ø :
 3ØØØ DEFPROCinput
 3Ø1Ø REPEAT
 3Ø2Ø PROCwindow1:CLS:PROCwindow2
 3Ø3Ø PROCfilenames
 3Ø4Ø PROCsortfield
 3Ø5Ø IFsf=1PROCalfnum
 3Ø6Ø PROCascdec
 3Ø7Ø UNTILFNask("Confirm (Y/N): ")<3
 3Ø8Ø ENDPROC
 3Ø9Ø :
 4ØØØ DEFPROCfilenames
 4Ø1Ø REPEAT
 4Ø2Ø PRINTCHR$13Ø"Enter name of file to
 be sorted:"'CHR$131;
 4Ø3Ø INPUTTAB(15)""op$:PROCopen(op$)
 4Ø4Ø UNTILF1:PRINT
 4Ø5Ø REPEAT
 4Ø6Ø PRINTCHR$13Ø"Enter new name of sor
ted file  :"'CHR$131;
 4Ø7Ø INPUTTAB(15)""np$:PROCnopen(np$)
 4Ø8Ø UNTILF2:PRINT
 4Ø9Ø PROCwindow1:PRINTCHR$134"Old file:
"CHR$131;op$TAB(2Ø)CHR$134"New file:"CHR
$131;np$
 41ØØ ENDPROC
 411Ø :
 5ØØØ DEFPROCsortfield
 5Ø1Ø PRINTTAB(9,3)CHR$134"SORTFIELD COM
POSITION"TAB(Ø,5)CHR$133"Field"SPC6"Widt
h  Start  Length  Total"
 5Ø2Ø PRINTTAB(32,6)CHR$134"<=";maxl
 5Ø3Ø lf=Ø:sf=Ø
 5Ø4Ø REPEAT:PROCwindow2:REPEAT
 5Ø5Ø PRINTCHR$13Ø"Enter field name";:IF
sf>ØPRINT", or <RETURN>"'CHR$13Ø"to end
composition";
 5Ø6Ø PRINT":"CHR$131;:INPUT""sfld$
 5Ø7Ø UNTIL(sf>ØANDsfld$="")ORFNfwidth(s
fld$)
 5Ø8Ø IFsfld$=""GOTO525Ø ELSEsf=sf+1
 5Ø9Ø sfld%(sf)=R:strt%(sf)=1:len%(sf)=F
Nfwidth(sfld$)
 51ØØ PROCwindow1:PRINTTAB(Ø,sf+6)CHR$13
1;sfld$TAB(16-LEN(STR$(len%(sf))))CHR$13
4;len%(sf):PROCwindow2
 511Ø REPEAT
 512Ø PRINTCHR$13Ø"Enter start position
in field or press"'CHR$13Ø"<RETURN> for
entire field:"CHR$131;
 513Ø INPUT""strt$:strt%=VALstrt$
 514Ø UNTILstrt$=""OR(strt%>ØANDstrt%<=l
en%(sf))
 515Ø IFstrt$<>""strt%(sf)=strt%
 516Ø PROCwindow1:PRINTTAB(23-LEN(STR$(s
trt%(sf))),sf+6)CHR$131;strt%(sf):PROCwi
ndow2
 517Ø IFstrt$=""GOTO523Ø
 518Ø REPEAT
 519Ø PRINTCHR$13Ø"Enter length of field
block or press"'CHR$13Ø"<RETURN> for res
t of field  :"CHR$131;
 52ØØ INPUT""len$:len%=VALlen$
 521Ø UNTILlen$=""OR(len%>ØANDlen%+strt%
(sf)<=len%(sf)+1)
 522Ø IFlen$<>""len%(sf)=len% ELSElen%(s
f)=len%(sf)-strt%(sf)+1
 523Ø IFlf+len%(sf)>maxl VDU7:PRINTTAB(8
)CHR$129"SORTFIELD TOO LONG":GOTO518Ø EL
SElf=lf+len%(sf)
 524Ø PROCwindow1:PRINTTAB(3Ø-LEN(STR$(l
en%(sf))),sf+6)CHR$131;len%(sf)TAB(38-LE
N(STR$(lf)),sf+6)CHR$134;lf
 525Ø UNTILsf=maxs ORsfld$="" ORlf=maxl
 526Ø ENDPROC
 527Ø :
 6ØØØ DEFPROCalfnum
 6Ø1Ø PROCwindow2:PRINT'CHR$13Ø"Press 'A
' for alphabetic sort"'TAB(3)CHR$13Ø"or
'N' for numeric sort :";
 6Ø2Ø REPEAT *FX15 1
 6Ø3Ø A=INSTR("AaNn",GET$):UNTILA
 6Ø4Ø PROCwindow1:PRINTTAB(11,9)CHR$131;
 6Ø5Ø IFA>2t$="VAL":PRINT"Numeric sort"
ELSEt$="":PRINT"Alphabetic sort"
 6Ø6Ø ENDPROC
 6Ø7Ø :
 7ØØØ DEFPROCascdec
 7Ø1Ø PROCwindow2:PRINT'CHR$13Ø"Press 'A
' for ascending order"'TAB(3)CHR$13Ø"or
'D' for descending order :";
```

→ 47

PROGRAM CRUNCHER

Insufficient memory for your programs can be a problem even on the Master and Compact. Jan Stuurman's short utility offers one solution, by squeezing your Basic programs into the smallest possible space.

The Basic program listed below will allow you to squeeze just a little more from your machine by 'crunching' a program to occupy a smaller space. It does this by combining statements to form longer lines, reducing the number of line numbers and hence the memory space required. Quite useful savings can be made in this way, particularly with longer programs.

Having typed the program in, save it to tape or disc before running it. Once the code is assembled correctly, follow the instructions displayed by the program for saving the machine code, and you should be ready to test the cruncher.

To crunch a Basic program, first load it into memory, and then type *RUN CRUNCH <Return>. After a short delay, the prompt should re-appear, and your Basic program will be compacted.

Warning: do not crunch a program unless you have saved an uncrunched copy first, and always do any editing with an uncrunched version. In fact, it is advisable not to crunch any program until you have finished developing it.

The utility will crunch all program lines, regardless of whether they contain Basic or assembler instructions, although lines that have a GOTO pointing to them will not be crunched onto a previous line. Because the crunched program may have some line numbers missing, you may wish to renumber the whole program.

The crunch program is located at &7ØØØ. To relocate it (e.g. to &58ØØ on the Electron), alter the value of HIMEM in line 1ØØ to read HIMEM=&58ØØ.

```
   1Ø REM PROGRAM BUCMPCT
   2Ø REM VERSION B 1.Ø
   3Ø REM AUTHOR  Jan Stuurman
   4Ø REM BEEBUG  December 1986
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
  1ØØ MODE7:HIMEM=&7ØØØ:S%=HIMEM
  11Ø PROCassemble
  12Ø PRINT''"To SAVE code, type"''" *SA
VE CRUNCH ";~S%;" ";~P%;" ";~start;" ";~
S%
  13Ø END
  14Ø :
 1ØØØ DEFPROCassemble
 1Ø1Ø listlo=S%:listhi=S%+&1ØØ:buffer=S%
+&2ØØ
 1Ø2Ø ptr=&7Ø:oldptr=&7Ø:newptr=&72
 1Ø3Ø listptr=&74:bufptr=&75
 1Ø4Ø linenbr=&8Ø:linelength=&82
 1Ø5Ø temp=&76:endflag=&77
 1Ø6Ø FOR pass=Ø TO 3 STEP3
 1Ø7Ø P%=S%+&3ØØ
 1Ø8Ø [OPT pass
 1Ø9Ø .start
 11ØØ LDA #Ø:STA listptr
 111Ø JSR setuptable
 112Ø JSR backupline
 113Ø RTS
 114Ø .setuptable \ of line numbers
 115Ø LDA #1:STA ptr \ program loaded at
 PAGE
 116Ø LDA &18:STA ptr+1 \ ptr to start o
f program
 117Ø .nextline
 118Ø LDY #Ø
 119Ø LDA (ptr),Y:CMP #&FF:BEQ endofprog
 12ØØ STA linenbr+1 \ high byte first
 121Ø INY:LDA (ptr),Y:STA linenbr \ then
 low byte
 122Ø INY:LDA (ptr),Y:STA linelength
 123Ø .nextbyte
 124Ø INY:LDA (ptr),Y
 125Ø CMP #&8D:BNE noref
 126Ø JSR listref:JMP nextbyte
 127Ø .noref
 128Ø CMP #&ØD:BNE nextbyte
 129Ø .endofline
 13ØØ CLC:LDA ptr
 131Ø ADC linelength:STAptr:BCC nextline
 132Ø INC ptr+1:BCS nextline
 133Ø .endofprog
 134Ø RTS
 135Ø .listref \decode & list linenumber
 136Ø LDX listptr
 137Ø INY:LDA (ptr),Y:PHA
 138Ø AND #&3Ø:EOR #&1Ø:ASL A:ASL A:STA
temp
 139Ø INY:LDA (ptr),Y:AND #&3F
 14ØØ CLC:ADC temp:STA listlo,X
 141Ø PLA:AND #&Ø4:EOR #&Ø4
 1420 ASL A:ASL A:ASL A:ASL A:STA temp
 1430 INY:LDA (ptr),Y:AND #&3F
 1440 CLC:ADC temp:STA listhi,X
 1450 .updatelistptr
 1460 INC listptr:BNE ok
 1470 BRK \ 256 linenumbers listed
 1480 OPT FNEQUB(100)
 1490 OPT FNEQUS("List space")
 1500 OPT FNEQUB(0)
 1510 .ok
 1520 RTS
 1530 .backupline
 1540 LDA #1:STA oldptr:STA newptr
 1550 LDA &18:STA oldptr+1:STA newptr+1
 1560 .newbuffer
 1570 LDA #0:STA bufptr:STA endflag
 1580 .readline
 1590 LDY #0
 1600 LDA (oldptr),Y:BMI endbuffer \ end
 of program
 1610 STA linenbr+1
 1620 INY:LDA (oldptr),Y:STA linenbr
 1630 INY:LDA (oldptr),Y
 1640 SEC:SBC #2:STA linelength
 1650 LDX bufptr:BNE check
 1660 LDA linenbr+1:STA buffer,X:INX
 1670 LDA linenbr:STA buffer,X:INX:INX
 1680 JMP readbyte
 1690 .check
 1700 LDA endflag:BNE endbuffer \ .token
 in last line
 1710 JSR checklist:BNE endbuffer \ line
nbr in listlo/hi
 1720 INY:LDA (oldptr),Y:DEY
 1730 CMP #&DD:BEQ endbuffer \ line star
ts with DEF
 1740 CMP #&DC:BEQ endbuffer \ line star
ts with DATA
 1750 LDX bufptr:TXA
 1760 CLC:ADC linelength:BCS endbuffer \
 buffer full
 1770 LDA #&3A \ insert colon in buffer
 1780 .inbuffer
 1790 STA buffer,X:INX
 1800 .readbyte
 1810 INY:LDA (oldptr),Y:PHA
 1820 CMP #&2A:BNE notstar
 1830 CPY#3:BEQ star \line starts with *
 1840 DEX:LDA buffer,X:INX
 1850 CMP #&3A:BNE notstar \ * not prece
ded by :
 1860 .star
 1870 STA endflag \ OS command
 1880 .notstar
 1890 PLA:BPL nontoken \ byte < &80
 1900 JSR checktoken:JMP inbuffer
 1910 .nontoken
 1920 CMP #&0D:BNE inbuffer \ not end ol
d line
 1930 STX bufptr:INY:TYA \ end old line
 1940 CLC:ADC oldptr:STA oldptr
 1950 LDA oldptr+1:ADC #0:STA oldptr+1
 1960 JMP readline
 1970 .endbuffer
 1980 PHA:INX:STX buffer+2:DEX \ store n
ew line length
 1990 LDY #0
 2000 .storebyte
 2010 LDA buffer,Y:STA (newptr),Y
 2020 INY:DEX:BNE storebyte
 2030 LDA #&0D:STA (newptr),Y
 2040 INY:TYA
 2050 CLC:ADC newptr:STA newptr
 2060 LDA newptr+1:ADC #0:STA newptr+1
 2070 PLA
 2080 CMP #&FF:BEQ finished \ end of pro
gram
 2090 JMP newbuffer
 2100 .finished
 2110 LDY #0:STA (newptr),Y
 2120 RTS
 2130 .checklist
 2140 STX temp
 2150 LDX listptr:BEQ nomatch
 2160 DEX:LDA linenbr+1
 2170 .listloop
 2180 CMP listhi,X:BNE listnext
 2190 LDA linenbr
 2200 CMP listlo,X:BEQ match
 2210 LDA linenbr+1
 2220 .listnext
 2230 DEX:CPX #&FF:BNE listloop
 2240 .nomatch
 2250 LDX temp
 2260 LDA #0:RTS
 2270 .match
 2280 LDX temp
 2290 LDA #1:RTS
 2300 .checktoken
 2310 STX temp:LDX #7
 2320 .tokenloop
 2330 CMP token,X:BEQ setendflag
 2340 DEX:BNE tokenloop
 2350 CMP #&8D:BNE return
 2360 .codedlinenbr \ transfer to buffer
 2370 LDX temp:STA buffer,X:INX
 2380 INY:LDA(oldptr),Y:STA buffer,X:INX
 2390 INY:LDA(oldptr),Y:STA buffer,X:INX
 2400 INY:LDA(oldptr),Y
 2410 RTS
 2420 .setendflag
 2430 STA endflag
 2440 .return
 2450 LDX temp:RTS
 2460 .token \last statement on new line
 2470 OPT FNEQUD(&F4DCE700) \rem,data,if
 2480 OPT FNEQUD(&E0E1F8EE) \end,endproc
,return,on
 2490 ]
 2500 NEXT
```

→27

Acorn's ViewPlot Reviewed

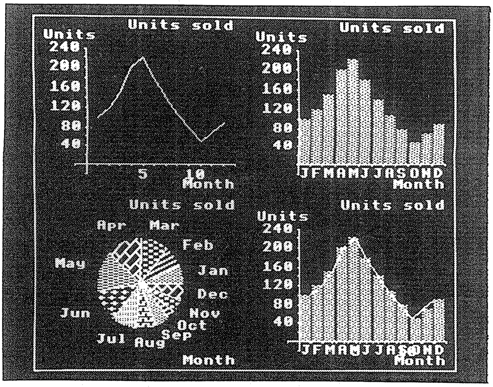
If graphical displays of ViewSheet or ViewStore data appeal to you, then read on. David Otley has been trying out Acorn's latest addition to the View family, ViewPlot.

Product : ViewPlot
Supplier : Acorn Computers Ltd.
Cambridge Technopark,
645 Newmarket Road,
Cambridge CB3 8PD.
Tel. (0223) 214411
Price : £25.30

The ViewPlot program is part of Acornsoft's View family, and allows various graphs and charts to be produced from data contained in a ViewSheet spreadsheet (or a ViewStore data file). It also allows charts to be constructed from data which is directly entered into the program.

The disc-based program is menu-driven with three main sections controlling data entry, chart presentation from a single set of data, and the production of graphs combining several sets of data. Direct data entry is quite straightforward and allows for up to 100 pairs of data values, which must then be saved to a disc file. A chart of the data can then be prepared by choosing the next menu option and following the prompts given. A choice of line graph, bar chart or pie chart is offered, although moving from one form of presentation to another requires all the menu prompts and the data file to be re-entered - an irritating feature once serious work has begun.

Once a graph is displayed, the colours of the backgound, axes and lines can be altered by using the function keys, and the most pleasing form of presentation quickly obtained. Unfortunately, although modes 0, 1, 2, 4 and 5 can be used on the B+ and Master (and if a 6502 second processor is used), only modes 4 and 5 are available on a standard BBC B, presumably due to the program space required. Further, if the mode is changed or the type of chart required is altered, time-consuming disc accesses are needed as parts of the program re-load and the data is re-accessed.



Data can be combined in various ways. Firstly, either one, two or four graphs and charts can be displayed on the screen at the same time by splitting the screen into windows. This allows, for example, the same data to be viewed in the three alternative modes of presentation, or for four different series of data to be seen together (see illustration), although this must be done in mode 4 on the model B.

Secondly, graphs of the same type (or different types, although this is not recommended!) can be overlaid on one another. In this case, the vertical axis is rescaled so that it includes the maximum range in all of the data sets, and provision for a new title is made. Unfortunately, the horizontal axis overwrites itself if different legends have been used on the original graphs. This facility is best used with line graphs, as bar charts rapidly become illegible when more than two or three are combined. The process is completely automatic with the user having no control over, for example, the spacing and widths of the bars. But again, the colours used in the display can be adjusted, and the patterns used to fill bars and pie segments selected (see illustration).

Once a screen display has been constructed and adjusted to give the most pleasing presentation, it may be saved to disc. Here one of the poorer aspects of the package becomes apparent. The screen dump automatically saves to the program disc in drive Ø under the name Image. The file thus has to be *COPY(ed) to another disc and renamed if several dumps are required. Similarly, data is saved on the program disc unless the rather tedious :1.filename notation is used. The manual makes no mention of this latter possibility and is otherwise rather thin on details of importance to the serious user. One would really expect a program of this sort to ask whether separate data and program discs were being used, and on which drives.

A printer dump is also supplied that allows a screen to be sent to the printer. Only one size of dump is allowed (approx 9" x 6", fitting neatly onto A4 paper) and the dump is for Epson FX or compatible printers. It worked adequately on my MX8Ø, but took over 8 minutes to dump a single screen.

However, the main reason for the program's existence is to extract data from Viewsheet spreadsheets and to plot graphs from it. This can be done in two ways. The first involves spooling part of the spreadsheet using the *prepare program supplied. This works easily enough, provided the data required is in two adjacent columns (not rows!), and a data file suitable for ViewPlot is the result. The second method involves the creation of a link file. The advantage of the link file is that it is permanently associated with the appropriate spreadsheet, so that whenever a change is made to the spreadsheet, the link file is automatically updated. In this way you can move quickly between the spreadsheet and the graphical plots. The disadvantage is that the manual is very cryptic as to how such files should be created, and will probably be quite incomprehensible to most users. I am familiar with the BBC micro and the ViewSheet spreadsheet, but it took me several hours of trial and error to get a working link file. It is quite inexcusable for such a vital feature to be so badly explained, and it will make the whole program useless to many purchasers.

The best feature of the program lies in its ability to construct a 'slide' show of linked screens of graphs that can be presented in sequence. The screen output is so much better than the printed output and, in any case, there seems to be no simple way of incorporating graphs into reports produced using View. The slide facility allows a whole series of graphs, either singly, or in twos or fours, to be presented in selected colours. The number in a sequence is limited only by the catalogue capacity, as one file is linked to call the next file. The screen display is quite quickly written, and the whole effect is impressively professional.

The program has to be compared with Interchart from Computer Concepts (see review in BEEBUG Vol.4 No.8). Viewplot suffers from being disc-based rather than ROM-based. It has less memory available for screen and data on a model B and is considerably less user-friendly. It is far easier to produce charts from Intersheet using Interchart than from ViewSheet using ViewPlot, and I cannot recommend the latter combination to new buyers. But if you already have ViewSheet or require its unique features (especially the screen windows) ViewPlot is the only way to get graphical output and, once set up, works passably well. Its strongest feature is its ability to produce a 'slide' show, and if you wish to present data in this form, it is well worth the price. But it is badly let down by its manual on the vital matter of constructing the necessary link files.

[BEEBUG members may also like to consider the above comments in relation to our own series on Business Graphics - part two in this issue.]

←25

PROGRAM CRUNCHER

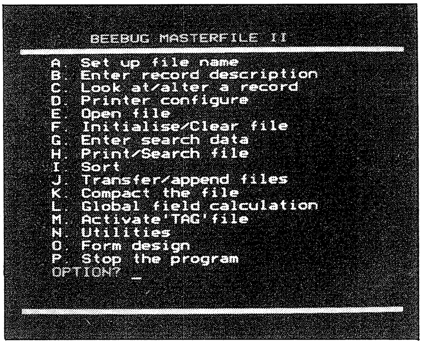
```
2510 ENDPROC
2520 :
2530 DEFFNEQUB(A%)
2540 ?P%=A%:P%=P%+1
2550 =pass
2560 :
2570 DEFFNEQUD(A%)
2580 !P%=A%:P%=P%+4
2590 =pass
2600 :
2610 DEFFNEQUS(A$)
2620 $P%=A$:P%=P%+LEN(A$)
2630 =pass
2640 :
2650 DEFPROCnum(I%)
2660 IFI%<&FFF PRINT;"Ø";
2670 PRINT;~I%;:ENDPROC
```

MASTERFILE II — General purpose database

Combines with Wordwise/View to produce Standard Letters etc.

MASTERFILE is a general purpose file management package allowing large amounts of information to be stored and processed. It is extremely powerful yet flexible and easy to use. Once set up, the information may be retrieved, sorted on any field, displayed, updated, printed etc. as required.

Separate versions of the program are available for cassette and disc users, the disc version using random access files and offering many extra features.



```
        BEEBUG MASTERFILE II

A. Set up file name
B. Enter record description
C. Look at/alter a record
D. Printer configure
E. Open file
F. Initialise/Clear file
G. Enter search data
H. Print/Search file
I. Sort
J. Transfer/append files
K. Compact the file
L. Global field calculation
M. Activate'TAG'file
N. Utilities
O. Form design
P. Stop the program
OPTION? _
```

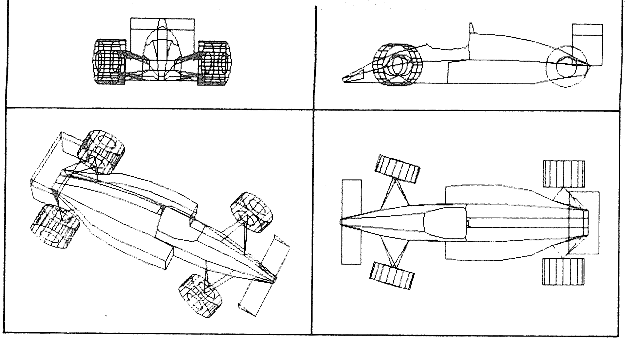
MASTERFILE II FACILITIES

- Extremely fast "tag" sorting on virtually any combination of fields. Any number of tag files may be associated with your data file.
- Flexible print layouts allowing additional text to be combined with the output to produce standard letters etc.
- Output from MASTERFILE II may be spooled directly to a file for later use with WORDWISE or VIEW.
- Flexible data entry.
- Fields may be defined as numeric, string, decimal or date and up to 255 characters in length.
- Spreadsheet type facilities allowing global changes to be made on string, numeric and decimal fields.
- Sub-files may be created from selected fields and records, from the master file, according to most criteria.
- Extra utilities to delete a field throughout a file, copy file descriptors and change dates.

ORDERING INFORMATION

	Price	Mem Price
Masterfile II	£22.00	£16.50
ADFS Masterfile II	£22.00	£16.50
Upgrade M.file II to ADFS M.file II	£10.00	£7.50
Upgrade M.file I (Disc) to M.file II	£12.50	£9.35
Upgrade M.file I (Disc) to ADFS M.file II	£12.50	£9.35

Please specify 40 or 80 tracks and standard or ADFS format.



BEEBUGSOFT UPGRADES

We offer a full service for members who wish to upgrade their Beebugsoft products from cassette to disc, disc to Rom etc., or to the latest version (e.g. Masterfile I to Masterfile II). To take advantage of this service please return your original program along with your order and remittance. Send all orders to the usual St. Albans address, marked 'UPGRADE' and add the usual postage charges.

Product	RRP	Members
● Design (cass) to Design (disc)	£15.00	£11.25
● Dumpmaster (cass) to Dumpmaster II (Rom)	£25.00	£18.75
● Dumpmaster (cass) to Dumpmaster (disc)	£9.00	£6.75
● Dumpmaster (disc) to Dumpmaster II (Rom)	£23.50	£17.60
● Exmon I (cass) to Exmon II (Rom)	£26.00	£19.50
● Exmon I (Rom) to Exmon II (Rom)	£18.50	£13.85
● Help (Rom) to Help II (Rom)	£18.50	£13.85
● Hershey Characters (cass) to Hershey Characters (disc)	£15.00	£11.25
● Masterfile I (cass) to Masterfile II (disc)	£16.00	£12.00
● Masterfile I (disc) to Masterfile II (disc)	£12.50	£9.35
● Masterfile I (disc) to ADFS Masterfile II	£12.50	£9.35
● Masterfile II (disc) to ADFS Masterfile II	£10.00	£7.50
● Paintbox I (cass) to Paintmaster (cass)	£7.00	£5.35
● Paintbox I (cass) to Paintmaster (disc)	£10.00	£7.50
● Paintbox I (disc) to Paintmaster (disc)	£9.00	£6.75
● Quickcalc (cass) to Quickcalc (disc)	£10.50	£7.85
● Sleuth (Rom) to Sleuth 1.06 (Rom)	£7.50	£7.50
● Spellcheck I (disc) to Spellcheck III (Rom + disc)	£26.50	£19.85
● Spellcheck II (Rom + disc) to Spellcheck III (Rom + disc)	£11.00	£11.00
● Sprites (cass) to Sprites (disc)	£9.00	£6.75
● Starter Pack (cass) to Starter Pack (disc)	£10.50	£7.85
● Teletext (cass) to Teletext (disc)	£9.00	£6.75
● Toolkit to Toolkit Plus	£25.50	£19.10
● Wordease (disc) to Wordease (Rom)	£18.50	£13.85

Personal Ads

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

We also accept members' business ads at the rate of 3Øp per word (inclusive of VAT) and these will be featured separately. Please send all ads (personal and business) to MEMBERS' ADS, BEEBUG, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX. The normal copy date for receipt of all ads will be the 15th of each month.

BBC B with Watford 32K RAM card, MK II ROM board (including 16K RAM) and a Watford DDFS complete with Wordwise Plus, Interword, Toolkit plus and six other ROMs fitted. Recently serviced and fitted with speech chips £35Ø. Will include a modem plus software for £4ØØ. Metal cased slimline disc drive 8Ø track 4ØØK £8Ø. Epson FX8Ø with Watford NLQ ROM £19Ø. Tel:(Ø5827) 5929 (St.Albans area).

Single disc drive, double sided 4Ø track (2ØØK). Powered from BBC. £6Ø plus carriage. Tel: Ø61-439 9665.

Wanted - Clean copy of book 'Making Music on the BBC Computer' by Ian Waugh. Tel:(Ø294) 53648.

Wanted - A good 8271 FDC. Also Unistat, Watford Alnoor Arabic ROM and OCCS Pascal 2.1 software packages. Please write, stating price, to George Scerri, 9 A/2 St.Mary Street, Ghaxaq, Malta G.C. Only reasonable offers will be considered.

ATPL Sidewise ROM/RAM board, £2Ø. Solidisk PC enclosure, £2Ø. ROMs: Accelerator £3Ø, Graphics (Computer Concepts) £14, Murom £14, Sleuth £14, ROM Manager (Watford) £1Ø, Discs: Paintbox II (8Øtrk) £8, 3-D Graphics (Glentop) 2 discs (8Øtrk) £14. All with manuals. P&p extra. Tel:(Ø722) 72219.

BBC B, colour monitor, recorder, Help ROM, joysticks, manual, mags, all for £25Ø. Tel:Ø1-446 3Ø33.

BBC model B linked to Torch Z8Ø (Torch BBC Basic) disc pack, twin 4ØØk 8Ø track drives. Basic II, DFS Ø.9, ATPL ROM board, View 1.4, other ROMs and software available. £5ØØ. Tel: Ø43 871 (Tewin) 7841.

BBC B latest specification, Solidisk DDFS, Sidewise ROM board, 16k sideways RAM, Vine tape to disc ROM, Beebugsoft Toolkit ROM, Speech upgrade, volume control, manuals. Cost over £63Ø - only £31Ø. Jennings, St.Albans (Ø727) 61835.

Mint condition double sided 4Ø/8Ø track Opus 58Ø2 disc drive £8Ø inc. p&p. New Floppy-Wise Master ROM, boxed with instructions £2Ø inc.p&p. Phone (Ø263) 78488.

BBC B, OS 1.2, plus Advanced User Guide £26Ø. Tel: Sheffield 682Ø1Ø or St.Helens 612153.

BBC B - ROM expansion board, DDFS, 65Ø2 2nd processor, 4Ø/8Ø twin drives, monitor stand, Hi-View, printer drivers, Watford NLQ, BCalc, Interchart, Masterfile II, Spellcheck II, manuals. £55Ø ono. Tel:Ø1-671 9233.

AMX Pagemaker Master 128, 8Ø track version £28, pristine condition, Island logic music system Master 128, 4Ø/8Ø track £15. The Artist by Wigmore House £35. G.Brown. Tel:Ø1-341 2187.

BBC B, DFS, Wordwise, BCPL, Solidisk 32K board, Canon dual disc drive, Kaga KP81Ø as new, lots of ROMs and other software and books - £1ØØØ ono. Tel: Sonny on Ø1-55Ø 5Ø82.

→ 33

←32

BBC B, ROM board, hard dust cover, Oxford Pascal, Opus DOS, Opus disc drive 4Ø/8Ø switchable, Shinwa CP8Ø printer and many extras. Original invoices available - complete price £575. Tel: Ø1-31Ø Ø414.

For Sale: Opus DDFS 3.45 £3Ø. Wordwise Plus £25. Wordease on 2 ROMs £18. Spellcheck III on ROM £18. Mini Office 4Ø/8Ø disc £4.5Ø. Few games cheap. Phone (Ø244) 315716 evenings.

4Ø track 1ØØK single sided disc drive £75 ono. Tel:(Ø628) 312Ø2.

Beebugsoft Toolkit Plus £25. Complete with box and manual. Owner upgrading to Master. Phone Bracknell (Ø344) 55772 after 6pm only.

Morley RAM disc, unused £15Ø. Various ROMs and manuls; Wordwise Plus £2Ø, Watford NLQ £1Ø, CJMicros NLQ (Epson) £2Ø, AMX mouse plus Pagemaker and Superart ROMs and software £75. Oberon Omni reader, unused £13Ø. Tel:(Ø4427) 57Ø2 evenings.

ATPL sidewise ROM board with 16K sideways RAM £25, Beebugsoft Sleuth ROM £2Ø. Phone Doncaster (Ø3Ø2) 744ØØ5.

Disc drive, Cumana CS1ØØ, 4Ø track, 1ØØK, own PSU £6Ø. Tel: Harefield 3Ø86.

Viglen 4Ø track disc drive 1ØØK for sale £1ØØ ono. Also BBC disc penpals wanted. Tel: East Grinstead (Ø342) 81Ø5Ø8.

Computer Concepts Speech ROM, with manual and Acorn 522Ø speech processor chip £25. R.J.Follett, 26 Arbor Lane, Winnersh, Berks. RG11 5JD.

Beebugsoft Sleuth 1.Ø6 as new £15. Tel:(Ø6Ø2) 231395.

Bitstick £275. Acorn 65Ø2 2nd processor with DNFS and HiBasic ROMs, £13Ø. Both boxed, as new. Tel: Paul, Derby (Ø332) 362798.

BBC B, Torch Z8Ø twin 4ØØK discs, joysticks, RAM/ROM board, sideways RAM, Music 5ØØ, about 4Ø books and over 1ØØ magazines £399. Phone Ø1-542 2747.

Oak universal BBC model B PC with Cumana twin double sided 4Ø/8Ø switchable disc drives, Phillips high resolution 12" monochrome monitor, CVX 16-2 battery backed ROM/RAM board, Wordwise Plus, Wordease, Spellcheck II and all manuals. Cost new over £1ØØØ, in very good condition £675 ono. L.Johnstone, Meikle Larg, Crocketford, By Dumfries, Scotland DG2 9ST.

Acornsoft Micro Prolog (unused) and some books £5Ø (save £2Ø). Tel:(Ø6Ø3) 811ØØ5.

Wordwise Plus ROM including all manuals and keystrip etc. £2Ø. Tel:Ø1-883 1415.

O and A level computer studies/science: detailed answers available (AEB board). For information, please send S.A.E. to: A. Beniston, 71 Princes Way, Fleetwood FY7 8DB.

BBC B, DFS Ø.9Ø, Basic II, OS 1.2, dual 4Ø track double sided disc drives, Watford sideways RAM, Phillips green monitor, Joysticks, cassette player, many books, magazines, software worth hundreds of pounds. £495. Tel:Ø51-428 313Ø.

Miscellaneous BBC software for sale. List from Carlin, St.Chad's Vicarage, Ragpath Lane, Stockton, Cleveland TS19 9JN.

BBC B, OS 1.2, Basic II including various games and books £185. Tel:(Ø227) 264986.

Fleet Street Editor £15, Masterfile II £9. Both are 4Ø track original discs and manuals in original packaging. Unwanted gifts. Peter Sotheran, Hime Farm, Yearby, Cleveland TS11 8HQ.

Flectron, Plus-1 interface, Cumana disc drive, Cumana disc interface, discs, T2CV ROM, View and Hopper ROM cartridges, many original software cassettes, all boxed as new and in excellent condition £35Ø the lot. Will split. If interested write to: TJN, 49 Blith-Meadow Drive, Sprowston, Norwich, Norfolk NR7 8PY.

Events

Higher Technology and Equipment in Education	Barbican Centre London	21-24 Jan
Electron and BBC Micro User Show	Old Horticultural Halls, Westminster	2Ø-22 Mar
Electron and BBC Micro User Show	Old Horticultural Halls, Westminster	8-1Ø May
Electron and BBC Micro User Show	Old Horticultural Halls, Westminster	13-15 Nov

Discounts Update

Waterloo Travel Service,
67A Cheadle Road,
Cheadle Hulme,
Cheshire.
Tel: Ø61-486 Ø912

The above company are offering 4% discount to members on all package holidays from any tour operator, any airport departure, and using their insurance. Please make sure you have your membership number handy.

High Scores

Supplier	Game	Score	Player
ACORNSOFT	Revs (Silverstone)	1:24.2	B.Hughes
ACORNSOFT	Revs (Brands Hatch)	1:25.5	B.Hughes
ACORNSOFT	Revs (Donington)	1:13.5	B.Hughes
ACORNSOFT	Revs (Oulton)	1:16.9	B.Hughes
ACORNSOFT	Revs (Snetterton)	1:Ø1.8	B.Hughes
BEEBUG MAG	Manhole	264	R.Koten
BEEBUG MAG	Scrumpy	45925	K.Rosser
BEEBUG MAG	Wee Shuggy	7643	R.Koten
MICRO POWER	Alien Swirl	8475Ø	C.Dunford
SOFTWARE INV	Blitzkrieg	1681ØØ	M.Reardon
SUPERIOR SOFT	Karate Combat	42333	M.Reardon
SUPERIOR SOFT	Mr.Wiz	2869Ø	R.Koten
SUPERIOR SOFT	Stryker's Run	14ØØØ	M.Williams
TYNESOFT	Jet Set Willy	26	R.Koten
ULTIMATE	Atic Atac	98%	R.Koten
ULTIMATE	Atic Atac	14956	R.Koten
ULTIMATE	Atic Atac	13.55 mins	R.Koten
ULTIMATE	Nightshade	12%	R.Koten
ULTIMATE	Nightshade	42ØØØ	R.Koten

Classified Ads

Oscilloscope teaching simulator, perfect for explaining CRO operation. Demonstration and real-time modes, extremely versatile. For sample and details send large SAE to: Osc-Demo, 63 Stanbrook Road, Solihull, West Midlands B9Ø.

Master Art for BBC Master 128. Take full advantage of the new graphic capabilities. Easily create stunning full screen pictures in mode 2 in 64 shades, using spray, fill, line and much more. Uses all 64K of SWR. Multi-screens in memory. Pull-down menus. Controlled by Mouse, Trackball or keyboard. Instant replay. Copy and paste between screens. Text in multi-size and 3D. 4Ø or 8Ø track, please specify. £25 inclusive, or SAE for specifications. No.3 Software, 3 Dairy Farm Court, Attleborough, Norfolk NR17 2BT.

MARKS AND STATISTICS For teachers and lecturers. Simple to use spreadsheet format. 33Ø students per file. Up to 3ØØ subjects. Sorts, analyses, normalises, calculates means, applies weighting factors. Prints histograms and lists. Full documentation supplied. Disc (4Ø or 8Ø track) £17. In-Form, 73 Woodfield Park, Colinton, Edinburgh EH13 ØRA.

Churchbase - disc database for BBC micro. Easy to use yet powerful and sophisticated. Very flexible print out. Full specification from: Carlin, St.Chad's Vicarage, Ragpath Lane, Stockton, Cleveland TS19 9JN.

UK Bulletin Boards
(continued)

MBBS Mitcham 24 Hours	Ø1 648 ØØ18 12ØØ/75 & 3ØØ/3ØØ
Metrotel 24 Hours	Ø1 941 4285 (London) 12ØØ/75 Viewdata
MG-NET Sun 17.ØØ-22.ØØ	Ø1 399 2136 3ØØ/3ØØ
Microlive 24 Hours	Ø1 579 2288 (London) 3ØØ/3ØØ
Microweb 24 Hours	Ø61 456 4157 (Manchester) 3ØØ/3ØØ
Mileways 18.ØØ-22.ØØ Mon-Fri 1Ø.ØØ-22.ØØ Sat/Sun	Ø533 6Ø8442 (Leicester) 3ØØ/3ØØ
MOBB 24 Hours	Ø61 736 8449 (Manchester) 12ØØ/75 & 3ØØ/3ØØ
Morecambe OBBS 24 Hours	Ø524 426133 (Morecambe) 3ØØ/3ØØ & 12ØØ/75
NBBS Aberdeen 24 Hours	Ø224 647158 (Aberdeen) 3ØØ/3ØØ
NBBS Cheshire 24 Hours	Ø936 77Ø25 (Cheshire) 12ØØ/75 & 3ØØ/3ØØ
NBBS East 24 Hours	Ø692 63Ø186 (Norfolk) 12ØØ/75 & 3ØØ/3ØØ
NBBS Essex 24 Hours	Ø277 228867 (Essex) 12ØØ/75 & 3ØØ/3ØØ
NBBS London 21.ØØ-Ø8.ØØ WE 24 Hrs	Ø1 883 529Ø (London) 12ØØ/75 & 3ØØ/3ØØ
NBBS Lutterworth 2Ø.ØØ-Ø8.ØØ WE 24 Hrs	Ø45 55 4798 (Lutterworth) 3ØØ/3ØØ
NBBS Midlands II 19.ØØ-23.ØØ WE 24hrs	Ø246 865843 (Chesterfield) 3ØØ/3ØØ & 12ØØ/75

→37

NOVACAD

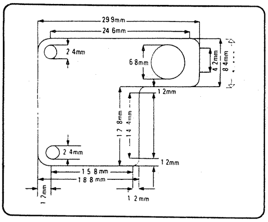
Computer Aided Draughting for the BBC Micro

> *"The Novacad system is a thoughtfully designed piece of software requiring a minimum of manual bashing so that real results are possible with just a few hours work."*

> *"Novacad is a professional package, well worth the £90 for the ROM and disc and is expandable to suit your own particular requirements. I would have no reservations if I heard Novacad called, "the best on the market"."* **A&B Computing August 1986**

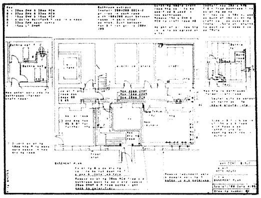
A sophisticated CAD package with unique facilities which are generally confined to the professional CAD packages costing thousands of pounds.

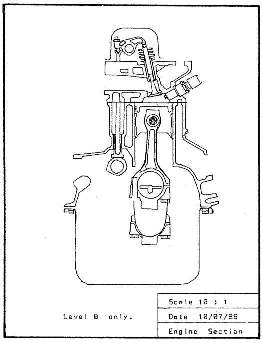
NOVACAD is being used in widely differing fields ranging from leading British industries for producing engineering drawings to Accountants' offices for producing flow charts. Architects, engineers, plumbers, kitchen planners and many other professions requiring easy to use facilities for producing professional quality output are making extensive use of NOVACAD's unique features. NOVACAD is proving to be a invaluable teaching aid in technical schools and colleges simply because of the ease with which it can be used.

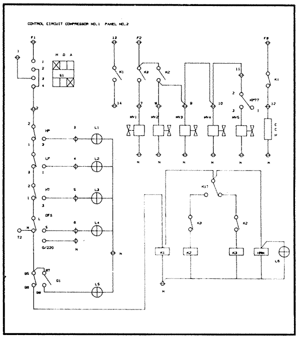


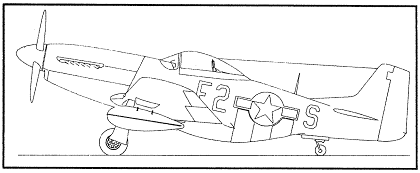
The major features of NOVACAD are:

- Simple and easy to use with logical command sequence.
- A large drawing area (approx 15 screens).
- The ability to create complex icons (upto 500 lines instead of the usual 15 to 20 pixels), store them in a library and recall them at different scales and orientations.
- A zoom facility allowing magnification of up to 20 times to insert finer details.
- Block copy of shapes on the screen for repetitive use.
- Text entry at any 90° orientation at any position on the screen.
- Create multi layer drawing with 8 levels available.
- Full compatibility with the BBC range with minimal hardware requirements yet having the capability of exploiting the potential of the extra shadow memory or the second processor's memory and speed.
- Printer routine for Epson 80 and 136 column printers.
- An option to output to any BBC compatible plotter of any size.









MINIMUM HARDWARE REQUIREMENTS

- **Any BBC range computer with Acorn DFS.**
- **40 Track single sided single drive.**
- **Epson compatible dot matrix printer.**

With basic BBC B, NOVACAD operates in mode 4 allowing use of any 2 colours. With shadow memory on the B+ or the Master, the user can work in mode 0 for high resolution or mode 1 for using any 4 logical colours in the drawings. 6502 second processor or the Turbo board will in addition to mode 0 and mode 4 operation increase the drawing speed.

NOVACAD is available in two versions

*BASIC version for cursor control from the keyboard.

*T/M version for cursor control from the keyboard, Tracker Ball or a Mouse.

OPTIONAL EXTRAS

Plotter driver generator is available as an optional extra to configure a plotter routine for any BBC compatible plotter. Pre-configured programs for the popular plotters like **Epson HI 80, Hitachi 672, Plotmate, Penman** and **HP GL.** This program allows the drawing to be scaled to the maximum plotter size thus enabling drawing of A2 and A3 sizes to be produced with equal ease. The zoom facility will enable inaccuracies of drawing on the screen to be eliminated.

ROUTCAD, an optional software package for our Graphics Workstation allows for production of shapes drawn using NOVACAD in 3D on suitable material and is ideal for a range of applications including template making and model building.

NOVACAD package includes a 16K ROM, Utility disc and a comprehensive manual.

NOVACAD £79 (d) | **NOVACAD T/M £99 (d)**
Stand Alone Mouse £34 (d) | **RB2 Tracker Ball (no software) £45 (c)**
Plotter Driver Generator £28 (d) | **RoutCad £28 (d)**

Prices do not include Carriage or VAT charges. Carriage charges (c) £1.50 (d) £1.00

TECHNOMATIC LTD.
13/17 Burnley Road, London NW10 1ED Tel. 01-208 1177
305 Edgware Road, London W2 ..... Tel. 01-723 0233

NBBS Marlow Sat-Thur 18.00-00.00 Ring and Ask	0628 46691 (Berkshire) 1200/75 & 300/300
NBBS NE 24 Hours	0224 641066 (Abrerdeen) 300/300 & 1200/75
NBBS Wallington 23.00-16.00	01 669 7249 (London) 1200/75 & 300/300
NKABBS 21.30-00.00	0795 842324 (Kent) 300/300 *
Norview 24 Hours	0604 20441 (Northampton) 1200/75 Viewdata
OBBS Bradford WD 18.00-00.00	0274 496783 (Bradford) 300/300
OBBS Manchester 24 Hours	061 427 1596 (Manchester) 300/300
OBBS 2 North Wales 24 Hours	0244 549336 (Wales) 300/300
Octopus 18.00-08.30	0272 421196 (Bristol) 300/300
Optel 24 Hours	01 794 0655 (London) 1200/75 Viewdata
OSI Lives! 24 Hours	01 429 3047 (London) 300/300
Owltel 24 Hours	01 927 5820 (London) 1200/75 Viewdata
PBBS-NI 22.00-00.00	0762 333872 (N.Ireland) 300/300
Pete's Place 24 Hours	0206 862354 (Essex) 300/300
PIP 24 Hours	0742 667983 (Sheffield) 300/300
Pot-Bug 20.00-23.00 Mon-Fri 14.00-23.00 Sat/Sun	0782 503254 (Stoke-on-Trent) 300/300 1200/75
PSBBS 18.00-23.00	0443 755298 (Wales) 300/300
React 24 Hours	0376 518818 (Essex) 300/300
RSGB 24 Hours	0707 57477 (Hertfordshire) 1200/75 Viewdata
SABBS 24 Hours	0698 884804 (Larkhall) 300/300
Sanctuary 24 Hours	0784 38110 (Surrey) 300/300
SBBS The Irishman 21.00-09.00	0247 455162 (N.Ireland) 300/300
SBBS Watford 21.00-08.00	0923 676644 (Hertfordshire) 300/300
Southern BBS 24 Hours	0243 511077 300/300
Stoke ITEC 24 Hours	0782 265078 (Stoke-on-Trent) 1200/75 Viewdata
Swafax 24 Hours	0622 850440 (Kent) 1200/75 Viewdata
System Aid 24 Hours	01 571 0026 (London) 1200/75 Viewdata
TBBS Blandford 24 Hours	0258 54494 (Dorset) 300/300
TBBS London 24 Hours	01 348 9400 (London) 300/300

MASTER SERIES

48/64K Printer Buffer Utility

Use Tim Powys-Lybbe's excellent printer buffer utility to free your Master or Compact during any form of printing.

How annoying to have to sit through long printing sessions, while the 64k of RAM in your Master remains idle. The accompanying program allows the whole of this area to be used as a buffer. If the assembled code is stored in one of the sideways RAM banks, it gives you up to 48k of buffer, or if put into ROM, you get up to 64k, providing that the RAM is not otherwise engaged.

To reap all these benefits, just type in the listing and save it away. Then run the program, and press f8 when the prompt appears. This will save the machine code to disc (filename Buffer). Then press f9 to load the machine code into sideways bank no. 4. To load it from disc on future occasions, use:

```
    *SRLOAD Buffer 8000 n (n=bank no.)
```

Once loaded in, perform Ctrl-Break to initialise the ROM image, then *HELP will give you the 4 new command names. They are:

*BUFFON	To turn it on
*BUFFOFF	To turn it off
*TESTBUFF	Test if on/off & full/empty
*CLEARBUFF	Instantly clear contents

Type *BUFFON (or even *BUF.), and print from Wordwise, View, Basic, the Editor, or even execute screen dumps without having to wait for the printer. Pressing Escape, and even performing a BACKUP will not affect the printing. But if you want to abort printout, just use *CLEARBUFF (or *CLE.).

```
10 REM Program Master Buffer
20 REM Version B 3.7
30 REM Author  Tim Powys-Lybbe
40 REM Beebug  December 1986
50 REM Program subject to copyright
60 :
  100 OScomd=&F2:ThisROM=&F4:ROMsel=&FE3
Ø:SidewaysVecs=&FFØØ:OSRDCH=&FFEØ
  110 OSASCI=&FFE3:OSNEWL=&FFE7:OSWRCH=&
FFEE:OSWORD=&FFF1:OSBYTE=&FFF4:CLI=&FFF7
  120 ZP=&7Ø:InsertVec=&22A:RemoveVec=&2
2C:CountVec=&22E:PrinterBuffer=&88Ø
  130 BankStart=&8ØØØ:BankTop=&CØØØ:z=Pr
interBuffer
  140 Xstore=z+8:Ystore=z+9:Astore=z+1Ø:
EmptyFlag=z+12:FullFlag=z+13
  150 MinRAM=z+14:MaxRAM=z+15:InputRAM=z
+16:OutputRAM=z+17
  160 TextMin=z+18:TextMax=z+2Ø
  170 InsertExit=z+22:RemoveExit=z+24:Co
untExit=z+26
  180 BREAKIntercept=z+28:Transfer=z+41
  190 TextInput=Transfer+1Ø:TextOutput=T
ransfer+15:TransferRAM=Transfer+1
  200 DIMQ%3ØØØ:FORPass=ØTO1
  210 O%=Q%:P%=&8ØØØ
  220 [OPTPass*2+4
  230 .Start BRK:BRK:BRK
  240 .Service JMPServiceHandle
  250 .ROMtype EQUB&82
  260 .CopyRightOffset EQUB(Copyright-St
art)
  270 .BinaryNo EQUB1
  280 .Title EQUS"Master Buffer"
  290 .Copyright BRK:EQUS"(C) T.Powys-Ly
bbe & Beebug 1986":BRK
  300 .ServiceHandle PHP:CMP#2:BEQInitia
lise
  310 CMP#4:BNEService1:JMPBuffer
  320 .Service1 CMP#9:BEQHelp:PLP:RTS
  330 .End PLA:TAY:PLA:TAX:PLA:PLP:RTS
  340 .EndZ PLA:TAY:PLA:TAX:PLA:LDA#Ø:PL
P:RTS
  350 .Initialise
  360 PHA:LDA#Ø:STA&DFØ,X:PLA:PLP:RTS
  370 .ZonkOnFlag:BCSZonk1:RTS
  380 .Zonk1 LDX(BREAKIntercept+Zonk2-Zo
nkOnFlag):LDA#Ø:STA&DFØ,X:RTS
  390 .Zonk2 EQUB&FF
  400 .TransferRoutine:LDY#&FF:STYThisRO
M:STYROMsel:BCCRead
  410 STATextInput:BCSReturn
  420 .Read LDATextOutput
  430 .Return STXThisROM:STXROMsel:RTS
  440 .TransferRoutineEnd
  450 .Help:PHA:TXA:PHA:TYA:PHA:DEY
  460 .Help1 INY:LDA(OScomd),Y:CMP#ASC"
":BEQHelp1
  470 CMP#&D:BNEHelp5:JSROSNEWL:LDX#&FF
  480 .Help2 INX:LDATitle,X:BEQHelp3:JSR
OSASCI:JMPHelp2
  490 .Help3 LDX#&FF
  500 .Help4 INX:LDAHelpWords,X:BEQHelp5
:JSROSASCI:JMPHelp4
  510 .Help5 JMPEnd
  520 .HelpWords EQUB13:EQUS"  Buffon":E
QUB13
  530 EQUS"  Buffoff":EQUB13:EQUS"  Clea
rbuff":EQUB13:EQUS"  Testbuff":EQUW&D
  540 .Buffer:PHA:TXA:PHA:TYA:PHA:DEY:ST
Yz+2:LDX#Ø:STXz+1:DEX
  550 .Buffer1 LDYz+2
  560 .Buffer2 INX:LDAHelpWords+3,X:CMP#
&D:BEQSortItOut
```

```
  57Ø .Buffer3 LDAHelpWords+3,X:AND#&DF:
STAz+3:INY:LDA(OScomd),Y:CMP#ASC" ":BEQB
uffer3:CMP#ASC".":BEQSortItOut
  58Ø AND#&DF:CMPz+3:BNEBuffer4:CMP#&D:B
NEBuffer2:BEQSortItOut
  59Ø .Buffer4 CMP#&D:BEQBuffer6:INX:LDA
HelpWords+3,X:BNEBuffer4
  6ØØ .Buffer5 JMPEnd
  61Ø .Buffer6 INCz+1:INX:INX:BNEBuffer1
  62Ø .SortItOut:LDXz+1:TXA:ASL A:TAX
  63Ø LDAJumpTable,X:STAz+1:LDAJumpTable
+1,X:STAz+2:JMP(z+1)
  64Ø .JumpTable EQUWBufferOn:EQUWBuffer
Off:EQUWClearBuffer:EQUWShowBuffer
  65Ø .BufferOn:LDYThisROM:STYz:LDA&DFØ,
Y:BEQTestForRAM:JMPEndZ
  66Ø .TestForRAM LDA#&44:JSROSBYTE:STXz
+1:BNETestForROMs:LDA#Ø:JMPMessageEndZ
  67Ø .TestForROMs LDAZP:PHA:LDAZP+1:PHA
  68Ø LDA#&AA:LDX#Ø:LDY#&FF:JSROSBYTE:ST
XZP:STYZP+1:LDY#3:LDA#1:STAAstore
  69Ø .TestFor1 INY:LDA(ZP),Y:ASLAstore:
CPY#8:BEQRAMBanks:AND#&CØ:BEQTestFor1
  7ØØ LDAAstore:LSRA:EORz+1:STAz+1:JMPTe
stFor1
  71Ø .RAMBanks PLA:STAZP+1:PLA:STAZP:LD
X#Ø:STXXstore:LDAz+1:BNERAMBanks1
  72Ø LDA#1:JMPMessageEndZ
  73Ø .RAMBanks1 LSRz+1:BCSRAMBanks3:LDA
Xstore:BNERAMBanks5
  74Ø .RAMBanks2 INX:CPX#4:BNERAMBanks1:
BEQRAMBanks5
  75Ø .RAMBanks3 LDAXstore:BNERAMBanks4:
TXA:CLC:ADC#4:STAMinRAM:STAMaxRAM
  76Ø INCXstore:BNERAMBanks2
  77Ø .RAMBanks4 INCMaxRAM:BNERAMBanks2
  78Ø .RAMBanks5
  79Ø .MoveToPrintBuffer:LDX#Ø
  8ØØ .MoveTo1 LDAZonkOnFlag,X:STABREAKI
ntercept,X:INX
  81Ø CPX#(TransferRoutineEnd-ZonkOnFlag
):BNEMoveTo1
  82Ø LDX#(Zonk2-ZonkOnFlag-1):LDAThisRO
M:STABREAKIntercept+1,X
  83Ø LDA#&F7:LDX#&4C:LDY#Ø:JSROSBYTE
  84Ø LDA#&F8:LDX#BREAKIntercept MOD256:
LDY#Ø:JSROSBYTE
  85Ø LDA#&F9:LDX#BREAKIntercept DIV256:
LDY#Ø:JSROSBYTE
  86Ø .SetPointers
  87Ø LDA#BankStart MOD256:STATextMin:LD
A#BankStart DIV256:STATextMin+1
  88Ø LDA#BankTop MOD256:STATextMax:LDA#
BankTop DIV256:STATextMax+1
  89Ø JSRInitTextPtrs
  9ØØ .SetIndirects:LDA#InsertVec MOD256
:STAAstore:LDX#SidewaysVecs DIV256:STXXs
tore:LSRA:CLC:ADCAstore:STAAstore:TAY:ST
YYstore:LDAZP:PHA:LDAZP+1:PHA
  91Ø LDX#Ø:LDY#&FF:LDA#&A8:JSROSBYTE:ST
XZP:STYZP+1:LDX#Ø:LDYYstore
  92Ø .SetIndirects1 LDARoutineAddresses
,X:STA(ZP),Y:LDAInsertVec,X
  93Ø STAInsertExit,X:INX:INY:LDARoutine
Addresses,X:STA(ZP),Y:LDAInsertVec,X
  94Ø STAInsertExit,X:INX:INY:LDAThisROM
:STA(ZP),Y:INY:CPX#6:BNESetIndirects1
  95Ø PLA:STAZP+1:PLA:STAZP
  96Ø .SetVectors:SEI:LDX#Ø
  97Ø .SetVectors1 LDAAstore:STAInsertVe
c,X:LDAXstore:STAInsertVec+1,X
  98Ø INCAstore:INCAstore:INCAstore:INX:
INX:CPX#6:BNESetVectors1
  99Ø LDA#1:JSRMark:JMPEndZ
 1ØØØ .RoutineAddresses EQUWInsertRoutin
e:EQUWRemoveRoutine:EQUWCountRoutine
 1Ø1Ø .BufferOff:LDYThisROM:LDA&DFØ,Y:BE
QBufferOff3:LDAEmptyFlag:BNEBufferOff1:L
DA#7:JMPMessageEndZ
 1Ø2Ø .BufferOff1 LDX#5:SEI
 1Ø3Ø .BufferOff2 LDAInsertExit,X:STAIns
ertVec,X:DEX:BPLBufferOff2:LDA#Ø:JSRMark
:LDA#&F7:LDX#Ø:LDY#Ø:JSROSBYTE
 1Ø4Ø .BufferOff3 JMPEndZ
 1Ø5Ø .ClearBuffer:JSRInitTextPtrs:JMPEn
dZ
 1Ø6Ø .InsertRoutine:PHP:SEI:CPX#3:BEQIn
sert1:PLP:JMP(InsertExit)
 1Ø7Ø .Insert1 STAAstore:PHA:TXA:PHA:TYA
:PHA:LDAFullFlag:BEQInsert2
 1Ø8Ø PLA:TAY:PLA:TAX:PLA:PLP:SEC:RTS
 1Ø9Ø .Insert2 LDAInputRAM:STATransferRA
M:LDAAstore:SEC:LDXThisROM:JSRTransfer
 11ØØ INCTextInput:BNEInsert3:INCTextInp
ut+1
 111Ø .Insert3 LDATextInput+1:CMPTextMax
+1:BNEInsert4:LDATextInput:CMPTextMax
 112Ø BNEInsert4:LDATextMin:STATextInput
:LDATextMin+1:STATextInput+1
 113Ø LDAInputRAM:CMPMaxRAM:BEQInsert3B:
INCInputRAM:BNEInsert4
 114Ø .Insert3B LDAMinRAM:STAInputRAM
 115Ø .Insert4 JSRInputEqualsOutput:BCCI
nsert5:LDA#1:STAFullFlag
 116Ø .Insert5 LDA#Ø:STAEmptyFlag
 117Ø .Insert6 PLA:TAY:PLA:TAX:PLA:PLP:C
LC:RTS
 118Ø .RemoveRoutine:PHP:SEI:CPX#3:BEQRe
move1:PLP:JMP(RemoveExit)
 119Ø .Remove1 PHA:LDAEmptyFlag:BEQRemov
e2:PLA:PLP:SEC:RTS
 12ØØ .Remove2 TXA:PHA:TYA:PHA:BVCRemove
4:JSRReadRAM
 121Ø .Remove3 PLA:PLA:TAX:PLA:LDYAstore
:TYA:PLP:CLC:RTS
 122Ø .Remove4 JSRReadRAM:INCTextOutput:
BNERemove5:INCTextOutput+1
 123Ø .Remove5 LDATextOutput+1:CMPTextMa
x+1:BNERemove6:LDATextOutput:CMPTextMax
```

```
 124Ø BNERemove6:LDATextMin:STATextOutpu
t:LDATextMin+1:STATextOutput+1
 125Ø LDAOutputRAM:CMPMaxRAM:BEQRemove5A
:INCOutputRAM:BNERemove6
 126Ø .Remove5A LDAMinRAM:STAOutputRAM
 127Ø .Remove6 JSRInputEqualsOutput:BCCR
emove7:LDA#1:STAEmptyFlag
 128Ø .Remove7 LDA#Ø:STAFullFlag:JMPRemo
ve3
 129Ø .CountRoutine:PHP:CPX#3:BEQCount1:
PLP:JMP(CountExit)
 13ØØ .Count1 PLP:BVCCount2:PHP:LDA&FF:B
PLCount1A:PLP:RTS
 131Ø .Count1A JSRInitTextPtrs:PLP:RTS
 132Ø .Count2 PHP:BCSCount4
 133Ø LDAFullFlag:BNECount3:JSRInputLess
ThanOutput:BCCCount3
 134Ø JSRInputRAMMinusOutputRAM:PLP:RTS
 135Ø .Count3 JSROutputRAMMinusInputRAM:
PLP:RTS
 136Ø .Count4 LDAFullFlag:BNECount6:JSRI
nputLessThanOutput:BCCCount6
 137Ø LDAFullFlag:BNECount6:JSRInputLess
ThanOutput:BCCCount6
 138Ø JSRInputRAMMinusOutputRAM
 139Ø .Count5 TXA:PHA:TYA:PHA:JSRRAMsize
:PLA:STAYstore:PLA:STAXstore:TXA:SEC:SBC
Xstore:TAX:TYA:SBCYstore:TAY:PLP:RTS
 14ØØ .Count6 JSROutputRAMMinusInputRAM:
JMPCount5
 141Ø .ShowBuffer
 142Ø LDA#2:JSRMessages:LDXThisROM:LDA&D
FØ,X:BNEShow1:LDA#4:JMPMessageEndZ
 143Ø .Show1 LDA#3:JSRMessages:LDAEmptyF
lag:BEQShow2:LDA#5:JMPMessageEndZ
 144Ø .Show2 LDAFullFlag:BEQShow3:LDA#6:
JMPMessageEndZ
 145Ø .Show3 JMPEndZ
 146Ø .Mark:LDYThisROM:STA&DFØ,Y:RTS
 147Ø .InitTextPtrs
 148Ø LDAMinRAM:STAInputRAM:STAOutputRAM
:LDATextMin:STATextOutput:STATextInput
 149Ø LDATextMin+1:STATextOutput+1:STATe
xtInput+1
 15ØØ LDA#1:STAEmptyFlag:LDA#Ø:STAFullFl
ag:RTS
 151Ø .InputEqualsOutput
 152Ø LDAInputRAM:CMPOutputRAM:BNEIEO1
 153Ø LDATextInput+1:CMPTextOutput+1:BNE
IEO1:LDATextInput:CMPTextOutput:BNEIEO1:
RTS
 154Ø .IEO1 CLC:RTS
 155Ø .InputLessThanOutput:LDAInputRAM:C
MPOutputRAM:BEQILTO1:RTS
 156Ø .ILTO1 LDATextInput+1:CMPTextOutpu
t+1:BEQILTO2:RTS
 157Ø .ILTO2 LDATextInput:CMPTextOutput:
RTS
 158Ø .InputRAMMinusOutputRAM:LDA#Ø:STAX
store:STAYstore:STAAstore:TAX:TAY
 159Ø LDAInputRAM:SEC:SBCOutputRAM:BEQIR
MOR1:STAAstore:JSRPartRAMsize
 16ØØ .IRMOR1 TXA:CLC:ADCTextInput:TAX:T
YA:ADCTextInput+1:TAY
 161Ø TXA:SEC:SBCTextOutput:TAX:TYA:SBCT
extOutput+1:TAY:RTS
 162Ø .OutputRAMMinusInputRAM
 163Ø LDA#Ø:STAXstore:STAYstore:STAAstor
e:TAX:TAY:LDAMaxRAM:CLC:ADCOutputRAM
 164Ø SEC:SBCMinRAM:SBCInputRAM:BEQOrmir
1:STAAstore:JSRPartRAMsize
 165Ø .Ormir1 TXA:CLC:ADCTextMax:TAX:TYA
:ADCTextMax+1:TAY:TXA:CLC:ADCTextInput:T
AX:TYA:ADCTextInput+1:TAY
 166Ø TXA:SEC:SBCTextOutput:TAX:TYA:SBCT
extOutput+1:TAY:TXA:SEC:SBCTextMin:TAX:T
YA:SBCTextMin+1:TAY:RTS
 167Ø .ReadRAM:LDAOutputRAM:STATransferR
AM:CLC:LDXThisROM:JSRTransfer:STAAstore:
RTS
 168Ø .RAMsize:LDA#Ø:TAX:TAY:LDAMaxRAM:S
EC:SBCMinRAM:STAAstore:INCAstore
 169Ø .PartRAMsize
 17ØØ LDATextMax:SEC:SBCTextMin:STAXstor
e:LDATextMax+1:SBCTextMin+1:STAYstore
 171Ø .PartRAMsize1 TXA:CLC:ADCXstore:TA
X:TYA:ADCYstore:TAY
 172Ø DECAstore:BNEPartRAMsize1:RTS
 173Ø .Messages:ASLA:TAX:LDAZP:PHA:LDAZP
+1:PHA:LDAMessageTable,X:STAZP:LDAMessag
eTable+1,X:STAZP+1:LDY#Ø
 174Ø .Messages1 LDA(ZP),Y:BEQMessages2:
JSROSASCI:INY:BNEMessages1
 175Ø .Messages2 PLA:STAZP+1:PLA:STAZP:R
TS
 176Ø .MessageTable EQUWNone:EQUWNoSpare
:EQUWMaster:EQUWOn:EQUWOff:EQUWBare:EQUW
Covered:EQUWCannot
 177Ø .MessageEndZ JSRMessages:JMPEndZ
 178Ø .None EQUS"No RAM found!":EQUB13:B
RK
 179Ø .NoSpare EQUS"No spare RAM":EQUB13
:BRK
 18ØØ .Master EQUS"Buffer ":BRK
 181Ø .On EQUS"ON":EQUB13:BRK
 182Ø .Off EQUS"OFF":EQUB13:BRK
 183Ø .Bare EQUS"EMPTY":EQUB13:BRK
 184Ø .Covered EQUS"FULL":EQUB13:BRK
 185Ø .Cannot EQUS"Can't: not empty!":EQ
UB13:BRK
 186Ø ]:NEXT:*K.9*SRLOAD Buffer 8ØØØ 4|M
 187Ø *K.8OSCLI("S.Buffer "+STR$~Q%+" "+
STR$~O%+" Ø FFFF8ØØØ")|M
```

MASTER SERIES

VDU Graphics made simple

The Master provides a greatly enhanced range of graphics features, if only you can follow the syntax. Thomas Nunns shows how to package up the graphics commands for ease of use.

When Acorn announced that the Master contained a new Basic and that many of the commands from the BBC's Graphics Extension ROM were included, I hoped very much that some of the more common VDU calls would be anglicised. It was not to be, only the old MOVE and DRAW (and questionably PLOT) can be 'read'. What a pity that such an excellent and readable language as BBC Basic uses such an unintelligible string of graphics commands. As an example:

```
    MOVE 780,520:VDU25,157,680;400;
```

must be used to draw a filled circle.

But we can force the Master to understand English with the aid of a few procedures. The accompanying program contains a set of procedures to draw triangles, rectangles, parallelograms and circles. The procedures are listed from line 1000 onwards, and they are accompanied by examples which put them through their paces.

The parameters used for each procedure are fairly self-explanatory, with x1,y1, x2,y2 etc. for the coordinates of successive points on the figure. The final parameter of each (except for the parallelogram, which is always filled) determines whether the object will be filled or not. If it is set to TRUE (or to -1) the object will be filled. Setting it to FALSE (or to 0) will draw the outline only. For example, the following call to PROCcircle:

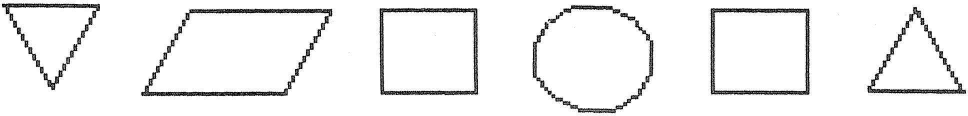
```
    PROCcircle(500,600,200,TRUE)
```

will draw a filled circle of radius 200 units, centred at the point 500,600.

SOURCES

The VDU commands are treated in detail in the Master Reference Manual part 1. But for Master and Compact users who are new to (and therefore probably mystified by) the advanced VDU commands, the best advice is to try and track down a manual for the Graphics Extension ROM (Acornsoft 1985 ISBN 0 907876 28 5). All of the GXR's commands are available unchanged in the Master's Basic, except for the sprites, and the GXR Manual not only has much clearer descriptions than the Master Reference Manual but it also includes many useful and pertinent examples.

```
   10 REM Program Simplifying VDU 25
   20 REM Version B 2.5
   30 REM Author  Thomas Nunns
   40 REM Beebug  December 1986
   50 REM Program subject to copyright
   60 :
  100 MODE129:VDU23,1|
  110 PROCtriangle(32,800,120,520,520,72
0,TRUE)
  120 PROCrectangle(580,720,400,260,TRUE
)
  130 PROCparallelogram(32,400,320,520,5
60,240)
  140 PROCcircle(980,520,156,TRUE)
  150 VDU23,1,1|:END
  160 :
 1000 DEF PROCtriangle(x1,y1,x2,y2,x3,y3
,fill)
 1010 MOVE x1,y1
 1020 DRAW x2,y2
 1030 IF fill VDU25,85,x3;y3; ELSE DRAW
x3,y3:DRAW x1,y1
 1040 ENDPROC
 1050 :
 1060 DEF PROCrectangle(x,y,width,height
,fill)
 1070 MOVE x,y
 1080 IF fill VDU25,101,x+width;y+height
; ELSE PLOT1,width,0:PLOT1,0,height:PLOT
1,-width,0:PLOT1,0,-height
 1090 ENDPROC
 1100 :
 1110 DEFPROCparallelogram(x1,y1,x2,y2,x
3,y3)
 1120 MOVE x1,y1
 1130 DRAW x2,y2
 1140 VDU25,117,x3;y3;
 1150 ENDPROC
 1160 :
 1170 DEF PROCcircle(x,y,radius,fill)
 1180 MOVE x,y
 1190 IF fill code=157 ELSE code=149
 1200 VDU25,code,x+radius;y;
 1210 ENDPROC
```



MASTER SERIES

Master Hints

Another selection of hints and other information for the Master and Compact rounded up by David Graham.

Checking Available RAM Banks

The following routine from Thomas Nunns prints which of the Master's internal RAM banks are switched in. It can be incorporated into any program as a check before loading ROM images etc.

```
1Ø A%=&44:R%=USR&FFF4 MOD&1ØØØ DIV&1ØØ
2Ø FOR C%=Ø TO 3:PRINT"Bank ";C%+4;
3Ø D%=2^C%:IF(R% AND D%)<>D% PRINT" pag
ed out" ELSE PRINT" available"
4Ø NEXT
```

Displaying and Saving Function Keys

The operating system supplies a star command, *SHOW, to display the contents of any function key, but it cannot be called with a wildcard to display the whole set. The following can be used in an EXEC file called "showkeys" to neatly display them all.

```
***************************
*| DISPLAY FUNCTION KEYS |*
***************************
P.':FOR A=Ø TO 9:PRINT"Key ";A;SPC2;:OS
CLI("SHOW"+STR$(A)):NEXT
```

The following, again stored in an EXEC file, perhaps called "savekeys", will save the keys to a file called "fkeys" in such a way that typing "*fkeys" at any time, will load them back in without disturbing any resident program or data etc. You may wish to delete the header and footer of the "fkeys" file which cause error messages as it loads in, but which otherwise do no harm.

```
**********************
*| SAVE FUNCTION KEYS *
*| to file: fkeys     *
**********************
*SPOOL fkeys
FOR A=1 TO 9:PRINT"*KEY";A;SPC2;:OSCLI(
"SHOW"+STR$(A)):NEXT
*SPOOL
```

ADFS Shortcuts

Listed below are a number of shorthand ADFS commands for cataloguing various directories.

COMMAND	EFFECT
*.-DISC-1	Cat DFS disc in drive 1
*-DISC-.1	ditto
*.:1	Cat root directory in drive 1
.$.P	Cat first sub directory of root directory starting with "P" on current drive
.:1.$.P	As above, but drive 1
*LCAT	Cat library directory

ADFS Wildcard Wipe

Perhaps the most annoying omission from the ADFS command set is the DFS equivalent of *WIPE which allows wildcards in specifying filenames to be deleted (like the ADFS *DESTROY), but which provides a YES/NO option before each file is deleted. This makes it an easy matter to clean up your discs, while retaining only those files which are really necessary. The following implementation, presented as an EXEC file goes some way to bridging the gap. When it is called, it requests a "Root name" for a group of files to be deleted, and then repeatedly asks for extensions to the root. It deletes these as they are entered, and both catalogues the current directory, and prints the free space remaining on the disc. It is halted by pressing Escape, and may be re-run by pressing fØ. It is, as you can see, self documenting.

```
MODE 128
*****************************
*|  *WIPE WILDCARD          *
*|  Enter Root name <ret>   *
*|  Then extension  <ret>   *
*|  Escape to Exit          *
*|  To repeat, Press fØ     *
*****************************
*KEYØ INPUT "Root name "A$:REPEAT:P.A$;
:INPUT " ext "B$:OSCLI("DELETE "+A$+B$)
:OSCLI("CAT"):OSCLI("FREE"):UNTIL Ø|M
*FX138,Ø,128
```

For further details of EXEC files see "Executive Power" in the previous issue of BEEBUG (Vol.5 No.6).

MASTER SERIES

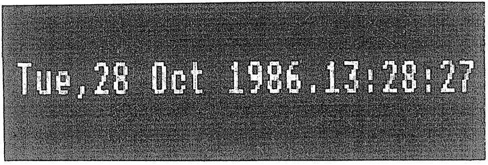
Master Clock Manager

Whatever your needs, David Graham's timely piece shows how to adjust the Master's built-in clock/calendar at the touch of a key.

The Master's clock calendar needs adjustment from time-to-time: either because it has drifted a few seconds, or in order to follow the vagaries of British Summer Time, or you may even have had to change the back-up battery. Either way, you should find this stand-alone utility a lot easier to use than the one on the Welcome disc.

It provides an uninterrupted digital clock/calendar display, and allows any parameter to be individually adjusted. The left and right cursor keys move a pointer along the base of the display to select the parameter to be updated. The up and down keys are then used to adjust the selected parameter. There is full error trapping, and the program keeps a check on the range of all parameters.

```
   10 REM Program Clock Manager
   20 REM Version B 3.C
   30 REM Author  David Graham
   40 REM Beebug  December 1986
   50 REM Program subject to copyright
   60 :
  100 ON ERROR GOTO 1540
  110 MODE7:PROCsetup
  120 REPEAT:REPEAT
  130 PROCclock
  140 I%=INKEY(Ø)
  150 UNTIL NOT I%
  160 IF I%=136 OR I%=137 PROCparam
  170 IF I%=138 OR I%=139 PROCchange
  180 UNTIL FALSE
  190 :
 1000 DEFPROCsetup
 1010 DIM array$(13,3),offset(1Ø)
 1020 DIM upper(1Ø),lower(1Ø)
 1030 N=1:VDU23,1|:*FX4,1
 1040 FOR A=1 TO 7
 1050 READ array$(A,1)
 1060 NEXT
 1070 FOR A=1 TO 12
 1080 READ array$(A,3)
 1090 NEXT
 1100 FOR A=1 TO 8
 1110 READ offset(A),upper(A),lower(A)
 1120 NEXT
 1130 PRINTTAB(6,12)"^"
 1140 ENDPROC
 1150 :
 1160 DEFPROCclock
 1170 FOR A=1 TO 2
 1180 PRINTTAB(4,9+A)CHR$134;CHR$141;TIM
E$
 1190 NEXT:ENDPROC
 1200 :
 1210 DEFPROCparam
 1220 N=N-(I%=136)*(N<>1)+(I%=137)*(N<>7
)
 1230 PRINTTAB(6,12)SPC(offset(N))"^";SP
C(24-offset(N))
 1240 ENDPROC
 1250 :
 1260 DEFPROCchange
 1270 inc=2*(I%=138)+1
 1280 IF N=2 OR N>3 THEN PROCdigits
 1290 IF N=1 OR N=3 THEN PROCtext
 1300 IF flag THEN TIME$=LEFT$(TIME$,off
set(N))+new$+RIGHT$(TIME$,25-offset(N+1)
)
 1310 ENDPROC
 1320 :
 1330 DEFPROCdigits
 1340 flag=TRUE
 1350 Z=offset(N)+1-2*(N=4)
 1360 new$=STR$(VAL(MID$(TIME$,Z,2))+inc
)
 1370 IF VAL(new$)>upper(N) OR VAL(new$)
<lower(N) THEN flag=FALSE
```


Tue,28 Oct 1986.13:28:27

```
 1380 IF LEN(new$)=1 THEN new$="Ø"+new$
 1390 IF N=4 THEN new$="19"+new$
 1400 ENDPROC
 1410 :
 1420 DEFPROCtext
 1430 old$=MID$(TIME$,(1-7*(N=3)),3)
 1440 B=Ø:flag=TRUE
 1450 REPEAT:B=B+1
 1460 UNTIL array$(B,N)=old$
 1470 new$=array$((B+inc),N)
 1480 ENDPROC
 1490 :
 1500 DATA Mon,Tue,Wed,Thu,Fri,Sat,Sun
 1510 DATA Jan,Feb,Mar,Apr,May,Jun,Jul,A
ug,Sep,Oct,Nov,Dec
 1520 DATA Ø,Ø,Ø,4,31,1,7,Ø,Ø,11,99,Ø,16
,23,Ø,19,59,Ø,22,59,Ø,25,Ø,Ø
 1530 :
 1540 ON ERROR OFF:VDU23,1,1|:*FX4
 1550 PRINT:REPORT:PRINT" at line ";ERL
 1560 END
```

1st course

Character Formation Part 2

David Graham continues to experiment with the Beeb's character set.

Last month we took a look at the way in which the information about any character is stored in the Beeb, and how that information can be read by the user and manipulated. This month we will put the "Read Character" routine, from the last issue, to further use, where it will form the basis of a set of procedures to generate double height and rotated text. These routines may easily be imported into your own programs to enhance screen displays.

The rotated character routine is particularly useful for labelling the vertical axes of charts and graphs, while the enlarged character procedure can be used to embellish program title screens and menus.

DOUBLE HEIGHT TEXT

Program 2 of last month's article contained a procedure to read the dot pattern of the Beeb's character set. This was listed from line 1ØØØ onwards, and was accompanied by routines which called this procedure, and then defined a number of characters derived from it, so as to produce upside-down and (in listing 3) inverted characters. Generating double height text using the same "Read Character" procedure is quite straightforward.

Essentially it involves defining two new characters: one for the top half of the double height character, and the other for the bottom. This is achieved by doubling up the definition. When the "Read Character" procedure is called, it emerges with the array B(n) containing the eight pieces of data for the row definitions of the specified character, in such a way that B(Ø) contains the value for the top row, B(1) the next, and so on. All we need to do is to define two new characters (251 and 252) as follows:

```
VDU23,251,B(Ø),B(Ø),B(1),B(1),B(2),B(2
   ),B(3),B(3)
VDU23,252,B(4),B(4),B(5),B(5),B(6),B(6
   ),B(7),B(7)
```

Notice how the doubling up has been performed, with each row value used to define two adjacent rows of the new characters.

Listing 1

```
  135 big$=CHR$(251)+CHR$(1Ø)+CHR$(8)+CH
R$(252)+CHR$(11)
  212 VDU23,251,B(Ø),B(Ø),B(1),B(1),B(2)
,B(2),B(3),B(3)
  214 VDU23,252,B(4),B(4),B(5),B(5),B(6)
,B(6),B(7),B(7)
  29Ø PRINT'TAB(15)big$
```

These lines have been incorporated into Listing 1. If you append this to last month's Program 2, and run the resulting program, you will see characters entered from the keyboard appearing in double height mode, as well as being inverted and reversed out. Line 135 of Listing 1 contains an important statement. It defines a text string big$ which is used to print each double height character. As you can see it actually incorporates a string of five individual characters. If we just went ahead and printed the two halves of our double height character, they would be printed side by side, which would be quite useless.

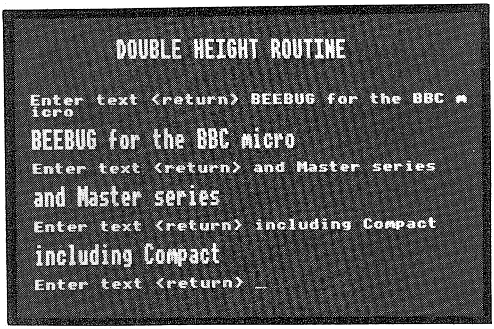
The string big$ fixes it all for us. It is defined as the top half of the double height pair, followed by character 1Ø, which sends the text cursor down one line, then character 8, which sends it back one position. It then prints the bottom half (in the correct position now), and then moves up one line (character 11), ready to print the next double height sequence. So every time we print big$, a complete double height character will be correctly printed. Which one it is, depends on the definitions of characters 251 and 252.

To make our double height routine more useful, we can parcel it up as a procedure in such a way that we can send whole strings of text to it in a single operation. This is achieved in Listing 2, which is a stand-alone program. It contains the "Read Character" procedure at line 1ØØØ, though this has been slightly modified from last month's version to make it more portable, and its name is now in lower case.

Listing 2

```
  1ØØ REM DOUBLE HEIGHT PROCEDURE
  11Ø REM WITH EXAMPLE v4
  12Ø :
  13Ø DIM B(8),D%1Ø
  14Ø big$=CHR$(251)+CHR$(1Ø)+CHR$(8)+CH
R$(252)+CHR$(11)
  15Ø :
  16Ø MODE4
  17Ø PRINTTAB(8,3);
  18Ø PROCdouble("DOUBLE HEIGHT ROUTINE"
)
  19Ø PRINT'
  2ØØ :
  21Ø REPEAT
  22Ø INPUT'''"Enter text <return> "inpu
t$
  23Ø PRINT
  24Ø PROCdouble(input$)
  25Ø UNTIL FALSE
  26Ø :
 1ØØØ DEF PROCchread(I)
 1Ø1Ø LOCAL A:?D%=I
 1Ø2Ø X%=D%MOD256:Y%=D%DIV256:A%=&A
 1Ø3Ø CALL &FFF1
 1Ø4Ø FOR A=Ø TO 7
 1Ø5Ø B(A)=D%?(A+1)
 1Ø6Ø NEXT
 1Ø7Ø ENDPROC
 1Ø8Ø :
 1Ø9Ø DEFPROCdouble(text$)
 11ØØ LOCAL A
 111Ø IF LEN(text$)=Ø THEN ENDPROC
 112Ø FOR A=1 TO LEN(text$)
 113Ø PROCchread(ASC(MID$(text$,A,1)))
 114Ø VDU23,251,B(Ø),B(Ø),B(1),B(1),B(2)
,B(2),B(3),B(3)
 115Ø VDU23,252,B(4),B(4),B(5),B(5),B(6)
,B(6),B(7),B(7)
 116Ø PRINTbig$;
 117Ø NEXT
 118Ø ENDPROC
```

The double height procedure is defined from line 1Ø9Ø. It has a parameter (text$) which contains the text which is to be printed in double height. It first checks that this variable is not empty (line 111Ø), and then enters a FOR-NEXT loop running from 1 to the length of the text, which takes one character at a time from the text string, passes it to PROCchread (line 113Ø), and creates the two halves of the character. The pair are printed using big$ (line 116Ø), and then the process is repeated with the next character from text$, and so on, until the whole of the contents of text$ have been printed in double height.



```
DOUBLE HEIGHT ROUTINE

Enter text <return> BEEBUG for the BBC m
icro
BEEBUG for the BBC micro
Enter text <return> and Master series
and Master series
Enter text <return> including Compact
including Compact
Enter text <return> _
```

The procedure is accompanied by an example of its use. Line 18Ø calls PROCdouble and causes the words "DOUBLE HEIGHT ROUTINE" to be printed in double height at the top of the screen. Notice how the print position is set using TAB in line 17Ø, immediately before calling PROCdouble. The program then accepts input from the user (line 22Ø) and passes this to PROCdouble with predictable results. You may alter line 16Ø to any graphics mode that you wish (i.e. modes Ø, 1, 2, 4 or 5), and may import the two procedures (lines 1ØØØ - 118Ø) into your own programs, but you will also need lines 13Ø and 14Ø.

ROTATED TEXT

PROCchread can also be used as the basis for a procedure to rotate characters through 9Ø degrees. This technique is used in plotting packages to allow the labelling of axes. If we are going to rotate a character we must perform a subtle reorganisation of its bit pattern. Again, we can use PROCchread to read the data from the bit pattern into the array B(n). But we can no longer rely on shuffling the order of the 8 row definitions. This can be used for standing the character on its head, or doubling its height. But to rotate it, we need to take the bit definitions of each column of the 8x8 graphics grid, and convert them into the rows of the new character.

Because the definitions are packed into just eight bytes, we need to unpack them in order to get at the column values. This involves a spot of binary arithmetic, since each pixel of the character's 8x8 grid is represented by a binary digit. Listing 3 contains a complete rotation procedure (lines 1Ø9Ø - 124Ø). It makes use of PROCchread (lines 1ØØØ - 1Ø7Ø), and is accompanied by an example. When the program is run, it asks for a string of text, followed by <Return>. This is then printed sideways from the bottom of the screen; and up to 1Ø strings may be printed in this way before the program terminates. This program is somewhat more complex than usual for this series, and rather too convoluted to describe in full. It is nevertheless easy to incorporate into any host program.

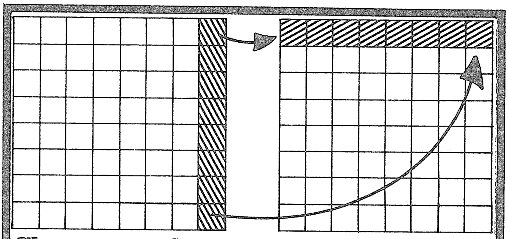
Listing 3

```
  1ØØ REM SIDEWAYS PRINT PROCEDURE
  11Ø REM WITH EXAMPLE v5
  12Ø :
  13Ø DIM B(8),D%1Ø
  14Ø side$=CHR$(246)+CHR$(11)+CHR$(8)
  15Ø MODE4
  16Ø PRINTTAB(8,1)"SIDEWAYS PRINT ROUTI
NE"
  17Ø pos=Ø
  18Ø REPEAT:pos=pos+1
  19Ø INPUT TAB(2,pos+4)"Enter text <ret
urn> "word$
  2ØØ PRINTTAB(pos,29);
  21Ø PROCside(word$)
  22Ø UNTIL pos>9
  23Ø PRINTTAB(Ø,31);
  24Ø END
  25Ø :
 1ØØØ DEF PROCchread(I)
 1Ø1Ø LOCAL A:?D%=I
 1Ø2Ø X%=D%MOD256:Y%=D%DIV256:A%=&A
 1Ø3Ø CALL &FFF1
 1Ø4Ø FOR A=Ø TO 7
 1Ø5Ø B(A)=D%?(A+1)
 1Ø6Ø NEXT
 1Ø7Ø ENDPROC
 1Ø8Ø :
 1Ø9Ø DEFPROCside(text$)
 11ØØ LOCAL A,I,N,value
 111Ø IF LEN(text$)=Ø THEN ENDPROC
 112Ø FOR A=1 TO LEN(text$)
 113Ø PROCchread(ASC(MID$(text$,A,1)))
 114Ø VDU23,246
 115Ø FOR N=Ø TO 7
 116Ø value=Ø
 117Ø FOR I=Ø TO 7
 118Ø value=value-2^I*((B(7-I) AND 2^N)>
Ø)
 119Ø NEXT
 12ØØ VDU value
 121Ø NEXT
 122Ø PRINTside$;
 123Ø NEXT
 124Ø ENDPROC
```

If you are going to use the procedure in another program, you will need lines 1ØØØ - 124Ø, together with lines 13Ø and 14Ø. Simply set the required print position with a TAB statement (as in line 2ØØ), and then call PROCside with the required text string as a parameter. This may be a variable, as in the example (line 21Ø), or a literal string as follows:

PROCside("Vertical Axis") etc.

Character 246 is used to hold the rotated image, and you may notice that the VDU23 definition has been split up and placed in a FOR-NEXT loop for convenience. As with the Double Height program, a string is set up in line 14Ø to print the character, and then move the cursor to the next print position, immediately above the position of the last character.



Character rotation by reading columns into rows.

It is hoped that these articles have given you a taste for character redefining, and that you may be tempted to experiment along similar lines, and maybe even try some new manipulations of your own. For example, you may like to have a go at double height reversed out, or for the foolhardy, what about double height rotated or double width ?

This month's magazine cassette/disc contains both programs from this article, plus an extra routine to print double height double width text - particularly useful for mode Ø applications.

```
 7020 REPEAT *FX15 1
 7030 A=INSTR("AaDd",GET$):UNTILA
 7040 PROCwindow1:PRINTTAB(11,sf+9)CHR$1
31;
 7050 IFA>2o$="<":PRINT"Descending order
" ELSEo$=">":PRINT"Ascending order"
 7060 ENDPROC
 7070 :
 8000 DEFPROCopen(p$):LOCALI
 8010 IFp$=""VDU7:PRINTTAB(10)CHR$129"NO
 FILE GIVEN":F1=Ø:ENDPROC
 8020 F1=OPENIN(p$)
 8030 IFF1=ØVDU7:PRINTTAB(1Ø)CHR$129"NO
SUCH FILE":ENDPROC
 8040 PTR#F1=Ø:INPUT#F1,rec,recn,recs,f:
maxl=(&7CØØ-B%)DIV(rec-1)-4
 8050 FORI=1TOf:INPUT#F1,field$,width%(I
):field$(I)=FNstrip(field$,"."):NEXT
 8060 CLOSE#F1:ENDPROC
 8070 :
 9000 DEFPROCnopen(p$):LOCALI
 9010 IFp$=""VDU7:PRINTTAB(1Ø)CHR$129"NO
 FILE GIVEN":F2=Ø:ENDPROC
 9020 F2=OPENIN(p$):CLOSE#F2
 9030 IFF2<>ØVDU7:PRINTTAB(1Ø)CHR$129"FI
LE EXISTS":IFFNask("Overwrite existing f
ile? (Y/N) : ")>2 F2=Ø:ENDPROC
 9040 PROCoscli("SAVE "+p$+" Ø+"+STR$~(F
DR+recn*recs))
 9050 F2=OPENUP(p$):PTR#F2=Ø:PRINT#F2,re
c,recn,recs,f
 9060 FORI=1TOf:PRINT#F2,LEFT$(field$(I)
+STRING$(12,"."),12),width%(I):NEXT
 9070 CLOSE#F2:ENDPROC
 9080 :
10000 DEFPROCclose:VDU7
10010 PRINTTAB(2)CHR$134"Files closed -
";rec-1;" records in use"
10020 CLOSE#Ø
10030 ENDPROC
10040 :
11000 DEFPROCsort
11010 LOCALI%,J%,K%,L%,U%,T$:L%=lf+4:PRO
Cwindow2:PRINTCHR$133CHR$136;
11020 F1=OPENIN(op$):F2=OPENUP(np$)
11030 s$="":FORI%=1TOsf:s$=s$+"MID$(reco
rd$("+STR$sfld%(I%)+"),"+STR$strt%(I%)+"
,"+STR$len%(I%)+")+":NEXT:s$=LEFT$(s$,LE
Ns$-1)
11040 PRINTTAB(13,Ø)"READING"
11050 FORI%=1TOrec-1:PROCread(I%)
11060 $(B%+L%*(I%-1))=EVALs$+RIGHT$("ØØ"
+STR$I%,3):NEXT
11070 t$=t$+"T$"+o$+t$+"$(B%+L%*(I%-1))"
11080 PRINTTAB(13,Ø)"SORTING"
11090 K%=1:REPEAT K%=3*K%+1:UNTIL3*K%+1>
=rec-1
11100 REPEAT K%=K%DIV3
11110 FORJ%=K%+1TOrec-1:T$=$(B%+L%*(J%-1
)):I%=J%-K%
11120 REPEAT U%=EVALt$:IFNOTU% $(B%+L%*(
I%+K%-1))=$(B%+L%*(I%-1)):I%=I%-K%
11130 UNTILU%ORI%<1:$(B%+L%*(I%+K%-1))=T
$
11140 NEXT:UNTILK%=1
11150 PRINTTAB(13,Ø)"WRITING"
11160 FORI%=1TOrec-1:J%=VALRIGHT$($(B%+L
%*(I%-1)),3)
11170 PROCread(J%):PROCwrite(I%)
11180 NEXT:CLS:ENDPROC
11190 :
13000 DEFFNask(msg$)
13010 LOCALA:PRINTCHR$134msg$;CHR$131;
13020 REPEAT:A=INSTR("YyNn",GET$):UNTILA
:IFA>2THENPRINT"N"ELSEPRINT"Y"
13030 =A
13040 :
14000 DEFPROCwindow1
14010 VDU28,Ø,21,39,3:ENDPROC
15000 DEFPROCwindow2
15010 VDU28,Ø,24,39,21,12:ENDPROC
15020 :
16000 DEFPROCread(n):LOCALI
16010 PTR#F1=FDR+recs*(n-1)
16020 FORI=1TOf:INPUT#F1,record$(I):NEXT
16030 ENDPROC
16040 :
17000 DEFPROCwrite(n):LOCALI
17010 PTR#F2=FDR+recs*(n-1)
17020 FORI=1TOf:PRINT#F2,record$(I):NEXT
17030 ENDPROC
17040 :
18000 DEFFNstrip(p$,c$):LOCALI:I=LENp$+1
18010 REPEAT:I=I-1:UNTILMID$(p$,I,1)<>c$
ORI=Ø
18020 =LEFT$(p$,I)
18030 :
19000 DEFFNfwidth(p$):IFp$=""=Ø
19010 R=Ø:REPEAT:R=R+1:UNTILp$=field$(R)
OR R=f
19020 IFR=f ANDp$<>field$(R)VDU7:PRINTTA
B(1Ø)CHR$129"NO SUCH FIELD"':=Ø ELSE =wi
dth%(R)
19030 :
20000 DEFPROCoscli($os)
20010 LOCALX%,Y%
20020 X%=os:Y%=os DIV256:CALL&FFF7
20030 ENDPROC
20040 :
21000 DEFPROCerror
21010 IFERR=17PRINT'TAB(12)CHR$129"SORTI
NG ABORTED":GOTO21Ø4Ø
21020 ONERROROFF
21030 REPORT:PRINT" at ";ERL
21040 PROCclose:VDU26
21050 ENDPROC
```

HINTS and tips HINTS and tips HINTS and tips HINTS and tips HINTS and ti

Speech Translation

Superior Software's "Speech!" program uses two commands: *SAY which does its best to pronounce a string written normally, and *SPEAK which utters a string made up of the set phonemes. In fact *SAY translates the 'English' string into phonemes and puts them in a buffer starting at &AØØ (normally the cassette input buffer) before using *SPEAK to deliver them.

To see how Speech! translates English into phonemes, and so learn a little about the phonemes themselves, this buffer can be inspected after a *SAY command is issued. Type in:

```
    *SAY HELLO
    PRINT $&AØØ
```

This will give '/HEHLOW' - the phoneme version of 'hello'.

Alan Street

Disabling DFS's

Acorn's 177Ø DFS and ADFS may be 'permanently' disabled - without restoration with Ctrl-Break - by the following method. Poke any number with the top bit set (e.g. &8Ø) into the relevant location in the ROM private workspace pointer table. The correct location can be found by adding the ROM socket number (Ø-15) to &DFØ. The ROM socket number can be found with the *ROMS command. For example, if the DFS is in socket 13 use:

```
    ?&DFD=&8Ø
```

Then press Break. This will disable the ROM until the machine is switched off and on, or the ROM is specifically re-enabled by poking a number with the top bit reset (e.g. zero) into the location, and pressing Break again.

It is worth noting the socket number of the DFS, as once the DFS is disabled, the *ROMS command won't work!

J.F.Button

Cursor Turn-Off

Although the cursor can be simply turned off using:

```
    VDU23,1,Ø;Ø;Ø;Ø;
```

this effect is cancelled by joining and separating the text and graphics cursors (VDU4 and 5). To turn off the cursor 'permanently' - until a change of mode - use the two *FX commands:

```
    *FX151,Ø,11
    *FX151,1,Ø
```

These FX commands access the registers of the 6845 display controller chip to remove the cursor.

C.N.Brickman

Star Command Workspace

When writing your own star commands, you often need a little zero page workspace. Although the 32 bytes from &7Ø to &8F are allocated to the user's applications, these should be avoided if the routine is to be used by other programs which might themselves need this space. Instead there are eight bytes from &A8 to &AF specifically reserved as star command workspace.

Arthur Neasden

BEEBUG Editor Extensions

The excellent Full Screen Editor program in BEEBUG Vol.4 No.7 can be extended with the following short additions:

To work with Shift enabled - i.e. a key with Shift producing lower case - add :*FX2Ø2,16Ø to the end of line 32ØØØ.

For faster cursor movement add :*FX12,4 to the end of line 3219Ø.

To produce an easy-to-see block editing cursor add :VDU23;1Ø,Ø;Ø;Ø; to line 3222Ø.

To restore the line cursor and clear the screen on Escape amend line 3236Ø to read:

```
 VDU12,23;1Ø,18;Ø;Ø;Ø;:*FX4
```

D.A.Stevens

Vertical Printing

Printing a string with each letter the correct way up but beneath the previous one is useful for labelling the Y-axis of a graph, amongst other applications. However, it is tedious (and slow) to produce the desired effect with a PRINT TAB command for each letter even if a loop is used. Instead, define a text window only one character wide, but several lines deep, and PRINT the string as normal. The Beeb does all the hard work by filling the window with the string of characters.

G.Hamilton

BEEBUG 29 FEB 1986 A.I.

POSTBAG

BEEBUG 29 FEB 1986 A.I. POSTBAG

DFS Gets Confused

I am sure that I am not the only user to have the following problems when using ViewStore. I have confirmed that the first is due to my use of the Ø.9 DFS, but wonder if this is also true in the second case.

1.The index utility will not produce a valid index. All seems to work correctly but the eventual index shows no records in the datafile.

2.The select utility works correctly to printer, but when using file output and sorting the resultant file, the whole contents of side 2 of the disc are backed up on side Ø, causing a loss of data.

Would it be too much to ask that software senses its operating environment and either warns the operator or aborts operation?

S.Waller

Both problems are due to the Ø.9 DFS which according to Acorn "gets confused" when files are open on more than one side of a DFS format disc. The DFS has "a neat trick of writing catalogues to the wrong disc surface which is usually disastrous"! The problem can be solved by upgrading from DFS Ø.9 to DFS (or DNFS) 1.2 or later versions. Apparently ViewStore users are warned of this need! ViewSpell users are also likely to be similarly affected.

Typing Problems

There is a severe limitation in the use of the *TYPE command to re-create a graphics screen from a *SPOOLed file generated by a graphics program (see *SPOOL Graphics, BEEBUG Vol.5 No.1). Characters 1Ø or 13 generated by the original program will be correctly stored in the *SPOOL file, but incorrectly executed by the *TYPE command: 1Ø will be ignored and 13 will be replaced by 1Ø and 13.

The following procedure will simulate the *TYPE command correctly:

```
DEFPROCTYPE(A$):LOCALA%,I%:
A%=OPENIN A$:FOR I%=1 TO EX
T#A%:VDU BGET#A%:NEXT:CLOSE
#A%:ENDPROC
```

Sebastian Lazareno

Richard Lambley, the author of the original article, comments: Since *TYPE is a function of the DFS its effects may vary from one DFS to another. The Basic procedure (given above) certainly does provide a general solution, though at some cost in speed.

Although *TYPE may seem to create a problem, I have not found it a limitation in practice. Where a problem really does arise is with the Master series. *TYPE now traps all characters with bit 7 set (e.g. Teletext codes). A new command *PRINT looks rather like the old *TYPE, but routes everything to the VDU drivers. So for ASCII 1Ø you get Linefeed and for 13 Return - and nothing extra.

Mastering ROM Software

In BEEBUG Vol.4 No.8 you previewed the Master 128, indicating that the use of sideways ROM software was out. But in more recent editions I get the impression that there are ways in which Wordwise and other similar ROMs can reside and operate in these machines. As a fully committed Wordwise Plus, Spellcheck III and Inter-sheet user, I am interested to know if I can still use this software in a new Beeb.

John Wardley

As Mr Wardley implies, the early information that we gave from Acorn regarding the use of sideways ROMs on the Master proved subsequently to be inaccurate. Sideways ROMs can be used on the Master, either fitted internally in a vacant ROM socket (two 16/32K, plus one 16K), or by using a ROM cartridge (16 /32K). We have a Master in use in the magazine office that normally has Wordwise Plus fitted internally with Acorn's ViewSpell and our own Spellcheck III fitted in a ROM cartridge, and all work well.

Of course, the fact that a ROM can be fitted to the Master does not in itself ensure that the ROM software is compatible with the Master, and this should be checked with the ROM supplier before purchase.

BEEBUG WORKSHOP Data Driven Routines

This month, Surac shows how to economise on your programming efforts by using so-called 'data-driven' routines.

Many programs often need the same sort of code time and again. It is tedious to keep re-writing it and so, perhaps, you build up a library of standard procedures and functions, *EXECing them into your programs as you need them. Sometimes, though, that doesn't work too well, particularly if a routine has a lot of special data associated with it.

A good example is a menu. All menus are much the same - they display a list of choices, accept and check a selection, and then send the choice off somewhere else to drive the main program. However, each menu has different choices. You end up rewriting the code, and playing around with TAB settings to get the effect you want. Several menus in a program use a lot of memory, too.

Professional programmers would not dream of taking that approach - they are too busy to re-invent wheels. Instead, they would use a standard routine and feed it with a list of data to make it do the job they wanted.

This technique of writing a single program which is fed with data to do different jobs is potentially very powerful. I've suggested a menu, but that's not the only option. How about a routine to validate a string against a number of possibilities? This approach is especially useful if you have disc(s), but that's far from essential.

To illustrate the approach, let's look at a general-purpose menu routine. As a minimum, it should take its information from DATA statements and be capable of dealing with any screen mode. You wouldn't always need all the bells and whistles of the result but it shows all the key points.

```
10000 DEF FNMenu
10010 LOCAL cols%,maxcol%,maxlines%,
        mode,nochoices%,OK,Opt%,rows%,
        title%,width,I%,J%,N%,X%
10020 RESTORE 20000
10030 CLS
10040 A%=135:!&70=USR&FFF4:mode=?&72
10050 width=40
10060 IF mode=0 OR mode=3 THEN width=80
10070 IF mode=2 OR mode=5 THEN width=20
10080 maxlines%=32
10090 IF mode=3 OR mode=6 OR mode=7
         THEN maxlines%=25
10100 READ title$
10110 title$=LEFT$(title$,width)
10120  PRINT  TAB((width-LEN(title$))/2,1)
       title$
10130 maxcols%=FNMin(width DIV 20,3)
10140 READ nochoices%
10150 nochoices%=FNMin(nochoices%,
            maxcols%*(maxlines%-6))
10160 cols%=(nochoices%-1) DIV
            (maxlines%-6)+1
10170 rows%=INT(nochoices%/cols%+.8)
10180 FOR I%=0 TO cols%-1
10190   X%=width/2-cols%*10+I%*20
10200   FOR J%=0 TO rows%-1
10210     N%=I%*rows%+J%+1
10220     IF N%<=nochoices%
            THEN PROCPChoice
10230     NEXT
10240   NEXT
10250 REPEAT
10260   PRINT TAB(12,maxlines%-2) SPC6
10270    PRINT  TAB(0,maxlines%-2)  "Which
Item? ";
10280   INPUT ""Opt%
10290   PRINT TAB(0,maxlines%-1)
            SPC(width-1);
10300   OK=Opt%<=nochoices% AND Opt%>0
10310   IF NOT OK THEN VDU 7:PRINT
          TAB(0,maxlines%-1) "1 - ";
          nochoices%;" ONLY!";
```

```
10320   UNTIL OK
10330 =Opt%
10990 :
11000 DEF PROCPChoice
11010 READ choice$
11020 PRINT TAB(X%,J%+3) RIGHT$("    "+
          STR$(N%)+". ",4)+
          LEFT$(choice$,15)
11030 ENDPROC
11990 :
12000 DEF FNMin(a,b)
12010 IF b<a THEN =b ELSE =a
19990 :
20000 DATA "THIS IS A SAMPLE TITLE"
20010 DATA 39:REM ** No. of choices
20020 DATA "Choice 0","Choice 1","Choice 2
","Choice 3","Choice 4","Choice 5","Choice
 6","Choice 7","Choice 8","Choice 9"
20030 DATA ...and all other menu lines
```

The routine is a function returning a number, corresponding to the choice, which the rest of the program can use.

At the start, it RESTOREs the DATA pointer to the correct line - there may be several functions for different menus in a single program. It then reads the mode using OSBYTE 135 to establish the maximum number of lines and columns on the screen. This is essential if the routine must handle any mode, but could otherwise be simplified.

It then starts building the screen. It reads the title, limits its size to the maximum display width, and prints it at the top of the screen (lines 10100-10120). The routine is set up to print the menu choices in columns 20 characters wide. Since this allows three columns in modes 0 and 3, two in 1, 4, 6 and 7, and just one in modes 2 and 5, line 10130 works out how many columns of choices the function can actually display.

The number of choices to be displayed is then read from the DATA statement. To play safe, line 10150 limits the number of choices to how many can be displayed. It allows 3 clear lines at the top and bottom of the screen. Thus, the 80x25 display of mode 3 has a maximum of 57 menu choices [3*(25-6)]. NOTE - although the function can accept this sort of number, easily-used menus rarely present more than 6-9 choices at a time.

Lines 10160 and 10170 work out how many columns and lines of choices to display; a single column will do for a few. The loop at lines 10180-10240 positions each choice and, via PROCPChoice, displays it on the screen with an identity number. Line 11020 aligns the numbers, and limits each choice to the maximum field size. The latter approach means that if a menu choice has been defined with too many characters, only the first 15 will appear, but it keeps the screen tidy.

Finally, lines 10250-10320 prompt for input, and check for a valid selection. They make sure that only numbers in the range displayed on the screen are accepted.

The main limitation of this routine is that, because of the DATA statements, it wastes a lot of memory. A separate function, with its own set of DATA, is needed for every menu in a program. You could use just one function, and pass the location of its DATA in a parameter:

```
10000 DEF FNMenu(dataline)
  :
10020 RESTORE dataline
  :
```

That would work, but it would be very dangerous. Why? If you renumber the program, the location of the DATA will change but the references in the calls to the function would not alter. The result would be total confusion. You would also still need precious memory for your DATA statements.

A much better way, but it needs discs, is to hold the data for each menu in a disc file, and let the function OPEN the file, INPUT the data, and then CLOSE the file afterwards. The following changes would be needed:

```
10000 DEF FNMenu(file$)
10010 LOCAL cols%,maxcol%,maxlines%,
        mode,nochoices%,OK,Opt%,rows%,
        title%,width,F%,I%,J%,N%,X%
10020 F%=OPENIN(file$)
10100 INPUT#F%,title$
10140 INPUT#F%,nochoices%
10325 CLOSE#F%
11010 INPUT#F%,choice$
```

You would also need a separate program to create the data files, but I leave that exercise to you.

GAMES REVIEWS

Title : Cluedo
Supplier : Leisure Genius
Price : £12.95 (Tape)
Reviewer : Alan Webster
Rating : ****

Cluedo is the third 'board' game released for the BBC micro by Leisure Genius, following hot on the heels of Monopoly and Scrabble. Cluedo is a who-dunnit? game in which up to six players (with any number of these being controlled by the computer) try to solve a murder that has taken place at 'Tudor Close'.

The game is a faithful eight-colour mode 2 copy of the original. As in the board game, each player takes on the identity of one of the suspects. It is the task of each player to piece together the clues to the crime. This involves moving from room to room, and quizzing your opponents. The winner is the first to deduce the three components of the crime:

Title : Chart Challenge
Supplier : Outlook Enterprises
Price : Tape or Disc £17.95
Rating : ****
Reviewer : Alan Webster

Chart Challenge is a pop quiz containing over 6Ø,ØØØ questions covering every artiste who has had a hit record in the U.K. between 1952 and 1985, as well as personalities, lyrics and albums.

Players can select one of four formats for Chart Challenge which can be a one or two player game. The timed quiz is probably the better of these versions, with any time up to 6Ø minutes allowed for each player.

Players can select any period between 1952 and 1985 on which to answer questions, but must select a span of at least five years.

There are a few minor niggles with the program (especially where abbreviations

Title : Stryker's Run
Supplier : Superior Software
Rating : ****
Reviewer : Daniel Gaster
Price : Tape £9.95, Disc £11.95

Yet another quality release in time for Christmas by Superior Software, this time in association with Acornsoft. Your mission, as Commander John Stryker, is to fight your way through enemy lines to HQ. In reality, this means travelling from left to right across a scrolling landscape avoiding enemy soldiers, landmines and bombs from enemy helicopters and planes.

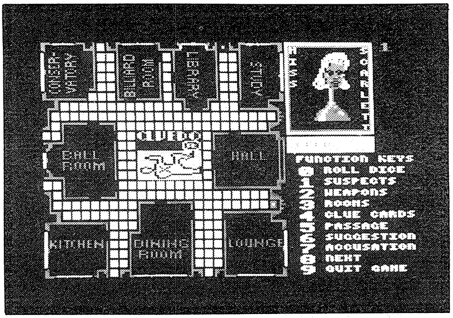
You can jump, duck, run, fire a pistol and throw grenades. You can also commandeer abandoned aircraft and helicopters to speed you on your way, but watch out for enemy aircraft. The graphics are the best part of the game; all the figures are drawn cartoon-style and even the larger aircraft are smoothly animated. The

Title : Mikie
Supplier : Imagine Software
Price : Tape £8.95 Tape only.
Rating : ***
Reviewer : Mike Williams

If there were a prize for the most unlikely game of 1986, then Mikie would surely romp away with an Oscar. Set in an American high school, you play the high school Romeo who is out to impress his girlfriend. You do this by collecting hearts scattered round the classroom, locker room, canteen etc while avoiding the teacher, janitor or whoever in authority stands in your way. Each heart collected builds into a message for Mikie's girlfriend. Thump your class-mates (a process called Hip-Zap) for extra points and to steal their hearts.

You can stun your pursuers by throwing chickens or basketballs at them, while the murderer, the weapon and the place of the dirty deed.

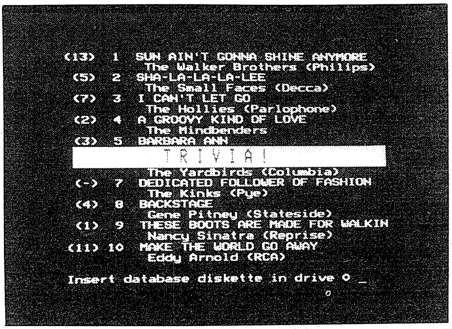
This is a good, colourful implementation of the old favourite by Waddingtons, but is really best suited to single player use. Half a dozen people cannot gather around a computer in the same way as they could a board game spread out on the table. The computer version is ideal for those unable to field a full team. Full marks to Leisure Genius, though, for the implementation.



are concerned), but with a 6Ø,ØØØ question database you would expect a few anomalies, and I don't think these detract too much from the general enjoyment of the game.

The quiz may seem expensive at £17.95, but is supplied on three 8Ø track discs, five 4Ø track discs or five cassettes, containing a total of 72ØK of data.

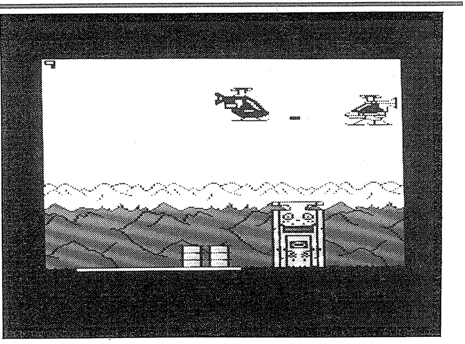
Overall, I enjoyed the game, and was able to polish up on my knowledge of the pop world. This is certainly another game to be recommended for Christmas.



```
(13)  1  SUN AIN'T GONNA SHINE ANYMORE
           The Walker Brothers (Philips)
 (5)  2  SHA-LA-LA-LA-LEE
           The Small Faces (Decca)
 (7)  3  I CAN'T LET GO
           The Hollies (Parlophone)
 (2)  4  A GROOVY KIND OF LOVE
           The Mindbenders
 (3)  5  BARBARA ANN
              T R I V I A !
           The Yardbirds (Columbia)
 (-)  7  DEDICATED FOLLOWER OF FASHION
           The Kinks (Pye)
 (4)  8  BACKSTAGE
           Gene Pitney (Stateside)
 (1)  9  THESE BOOTS ARE MADE FOR WALKIN
           Nancy Sinatra (Reprise)
(11) 10  MAKE THE WORLD GO AWAY
           Eddy Arnold (RCA)

Insert database diskette in drive 0 _
```

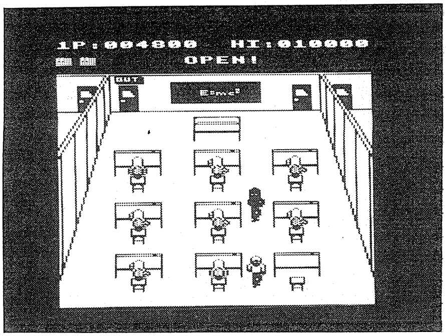
background of ruined buildings and other features are very well drawn, though they play no direct part in the game.

I felt that Stryker's Run was slighly disappointing in terms of originality but, as always with Superior, the actual game has been very well written. The package includes an enhanced Master version with some superb graphics, though the game is essentially the same. Overall, perhaps not Superior's best release this year, but still well above the standard of most other games.



when 'Teach' gets really mad, he starts throwing his false teeth about! In addition, as the instructions say, "Kissing is automatic"! Harrassed by the dancing girls (obviously normal in American schools) you may wonder if Mikie's girlfriend is really all worthwhile.

Well, Mikie may please some games fanatics, desperate for something different, but if you're looking for a really good game this Christmas then look elsewhere.



ROM ROUND-UP

Pssst! Wanna a ROM for Christmas? Geoff Bains has been trying out the latest utility ROMs for Beeb and Master.

Title : BROM+
Supplier : Clares Micro Supplies,
98 Middlewich Road,
Northwich, Cheshire.
Tel. (0606) 48511
Price : ROM £34.50, Disc £30.00

Brom Plus is a 16K ROM packed with over 35 commands. All the commands can be prefixed with a Z, in accordance with BEEBUG's system, to avoid conflict with commands in other ROMs. The commands are in three sections. The first covers Basic toolkit commands.

The expected 'Bad program' cure, search and replace (FIND and CHANGE), line renumber, move, and copy are all there. Also included is a formatted lister that indents all lines, spaces keywords, and splits multi-statement lines.

Brom Plus has both variable and cross reference listers to produce a listing of all real, integer, or string variables or array, procedure, or function names.

The most radical aspect of Brom Plus is its Basic editor. This is a mode 7 editor with more than a passing resemblance to Wordwise. Program lines can be freely edited in either insert or overwrite mode. New lines can be created and complete lines deleted. An unusual but useful feature is the change case facility (Ctrl-S, just like Wordwise). The Basic editor is not as powerful as others around but it is fast and very easy to learn.

The second section of Brom Plus is the disc utility section. This has the usual format and verify commands and a fairly standard disc sector editor. The editor uses mode 7 which means that only a small part of a sector can be displayed at a time. The data can be edited as hex numbers or ASCII characters, but there is no cursor in the ASCII section, so it can be unclear what you're editing.

A search routine for disc data is also provided, and this enters the sector editor when the string is found. Multiple file copy and delete commands offer a menu system for choosing which files are moved or removed.

An unusual feature is the two commands *DGET and *DPUT, to transfer chunks of data between memory and specified sectors of the disc. These can be particularly useful when rescuing 'lost' files, and the excellent Brom Plus manual contains a whole section on rescuing files with the Brom Plus commands.

Brom Plus also has an automatic menu system similar to that published in BEEBUG (Vol.4 Nos.2 and 5). Only files in the '&' directory are listed, and information describing the program is stored separately in a special file and created with the *UPDATE command.

The final section of Brom Plus is the general section. This has a function key lister, memory editor (similar to the disc sector editor) and a memory search command. ROM listing and enable/disable commands are also provided, as is a most useful *CASE command. This restricts input to upper, lower or mixed case, regardless of the Caps and Shift Lock keys.

There are many utility ROMs around. Brom Plus offers little unavailable elsewhere, but if you require some or all of these features, then Brom Plus is well thought out, well implemented and well worth it.

Title : Advanced Disk Investigator
Supplier : Advanced Computer Products,
6 Ava House, High Street,
Chobham, Surrey GU24 8LZ.
Tel. (0276) 76545
Price : £28.75

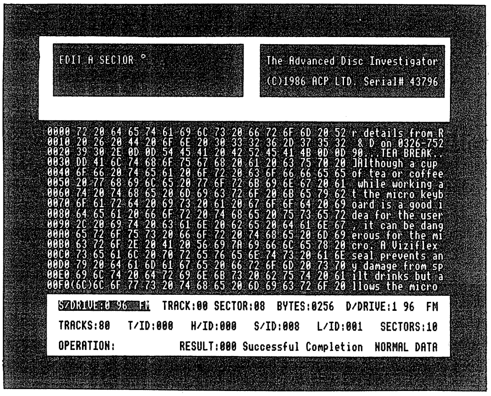
The Advanced Disc Investigator is a disc sector editor - just one of many utilities included in other ROMs. However, the ADI sector editor is the one to end all others. ADI is a 16K language ROM, It is compatible with the model B with an

8271 disc controller, a Master or B+ with an 177Ø or even a model B with most types of 177Ø upgrades.

ADI has just one command (*ADI) and one screen. This is a mode Ø screen with a picture of a disc in a sleeve. The disc 'label' is the command menu and the 'sleeve' displays status information about the disc drives and all of a disc sector in both hex and ASCII.

The commands are selected either with their initial letter or with a cursor menu system. This is the best of both worlds and a feature well worth imitating.

To cope with the various disc formats ADI can directly use all the parameters affecting the disc controller chip - number of tracks, density, track, sector, head and length IDs and so forth. ADI can also detect what a disc is already using, and defaults accordingly.



```
EDIT A SECTOR °                The Advanced Disc Investigator
                               (C)1986 ACP LTD. Serial# 43796

0000 72 20 64 65 74 61 69 6C 73 20 66 72 6F 6D 20 52 r details from R
0010 20 26 20 44 20 6F 6E 20 30 33 32 36 2D 37 35 32  & D on 0326-752
0020 39 30 2E 0D 0D 54 45 41 20 42 52 45 41 4B 0D 0D 90...TEA BREAK..
0030 DD 41 6C 74 68 6F 75 67 68 20 61 20 63 75 70 20 ]Although a cup
0040 6F 66 20 74 65 61 20 6F 72 20 63 6F 66 66 65 65 of tea or coffee
0050 20 77 68 69 6C 65 20 77 6F 72 6B 69 6E 67 20 61  while working a
0060 74 20 74 68 65 20 6D 69 63 72 6F 20 6B 65 79 62 t the micro keyb
0070 6F 61 72 64 20 69 73 20 61 20 67 6F 6F 64 20 69 oard is a good i
0080 64 65 61 20 66 6F 72 20 74 68 65 20 75 73 65 72 dea for the user
0090 2C 20 69 74 20 63 61 6E 20 62 65 20 64 61 6E 67 , it can be dang
00A0 65 72 6F 75 73 20 66 6F 72 20 74 68 65 20 6D 69 erous for the mi
00B0 63 72 6F 2E 20 41 20 56 69 7A 69 66 6C 65 78 20 cro. A Viziflex
00C0 73 65 61 6C 20 70 72 65 76 65 6E 74 73 20 61 6E seal prevents an
00D0 79 20 64 61 6D 61 67 65 20 66 72 6F 6D 20 73 70 y damage from sp
00E0 69 6C 74 20 64 72 69 6E 6B 73 20 62 75 74 20 61 ilt drinks but a
00F0(6C)6C 6F 77 73 20 74 68 65 20 6D 69 63 72 6F 20 llows the micro

S/DRIVE:0 96  FM  TRACK:00 SECTOR:08  BYTES:0256  D/DRIVE:1 96  FM
TRACKS:80   T/ID:000   H/ID:000   S/ID:008   L/ID:001   SECTORS:10
OPERATION:            RESULT:000 Successful Completion  NORMAL DATA
```

Two sets of this status information are displayed - for a source and destination disc. All the parameters can be changed with the cursor keys, although, strangely, the up key is used to decrease a value and the down key to increase it!

Once both discs are set up to your satisfaction some or all of the tracks can be formatted, verified, copied, edited, read or written to/from memory, printed out or 'unformatted'. The most impressive of these functions are the copier and editor. ADI can, with a little manipulation, copy just about any disc. I successfully used it to copy a Wordwise file onto a disc set up to be read by an IBM PC!

The editor is also excellent. This is the best disc sector editor I've seen. It is both fast and flexible. It does little more than other editors but it is pleasant to use - a definite rarity.

The ADI is a powerful package. There is little you'd want to do to a disc that this software can't help with. For serious disc users this is a must.

Title : Floppywise Master
Supplier : Software Services,
65 South Mossley Hill Road,
Allerton, Liverpool.
Tel. 051-427 7894
Price : £30.95

Floppywise Master is, as you'd expect, a ROM of disc utilities for the Master 128. It is an upgrade to the original Floppywise (reviewed BEEBUG Vol.4 No.1).

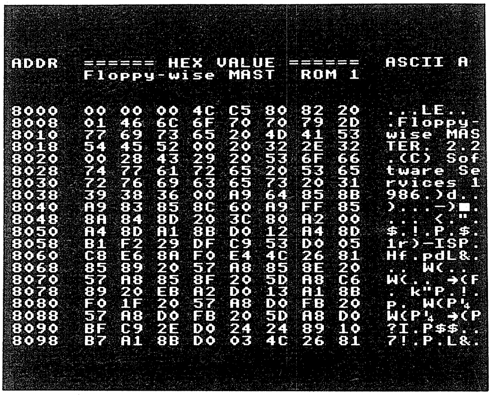
The ROM has (yet another) disc sector editor and main memory editor and a ROM inspector. These operate in mode 7 and are fast, if not original. Like BROM Plus, Floppywise Master has disc and memory search routines which enter the relevant editor on each find. It has multiple file copy and delete routines too, and even an automatic menu system.

*FCOMPARE compares a disc file with a section of memory. *RESTORE places data directly at a specified location on the disc. Most original is the *AUTOSAVE command. This will automatically save a copy of a program under development every four minutes without troubling the user.

Useful commands for Master owners are *AFORM and *AVERIFY. These format and verify a disc in ADFS mode. Three formats are available: 4Ø, 8Ø and 16Ø which treats both sides of a disc as one large disc. *FILES displays a hierarchical list of all the files on an ADFS disc - very useful when lost in a directory maze.

Routines are provided to enable or disable ROMs and load in ROM images from disc - rather pointless on the Master which already has the *SRLOAD command. However, more useful are the *RMOVE command to move sideways programs from one slot to another, *RSEARCH to search a sideways slot, *RCLEAR to clear a slot and *CROM to copy a ROM to disc.

Floppywise Master also has a selection of Basic toolkit commands which sit uneasily amongst the rest. *FIND is a standard Basic program string search. *CLEAR zeros all variables, *BAD recovers a 'Bad program', *BMOVE moves down a program intended for the cassette filing system. *FUNC lists the function key definitions, *ASCII prints out all the ASCII codes and characters and *TOKENS lists the Basic keyword tokens.



```
ADDR   ====== HEX VALUE ======   ASCII A
       Floppy-wise MAST  ROM 1

8000   00 00 00 4C C5 80 82 20   ...LE..
8008   01 46 6C 6F 70 70 79 2D   .Floppy-
8010   77 69 73 65 20 4D 41 53   wise MAS
8018   54 45 52 00 20 32 2E 32   TER. 2.2
8020   00 28 43 29 20 53 6F 66   .(C) Sof
8028   74 77 61 72 65 20 53 65   tware Se
8030   72 76 69 63 65 73 20 31   rvices 1
8038   39 38 36 00 A9 64 85 8B   986.)d..
8040   A9 83 85 8C 60 A9 FF 85   )...—)■.
8048   8A 84 8D 20 3C 80 A2 00   ... <.".
8050   A4 8D A1 8B D0 12 A4 8D   $.!.P.$.
8058   B1 F2 29 DF C9 53 D0 05   1r)—ISP.
8060   C8 E6 8A F0 E4 4C 26 81   Hf.pdL&.
8068   85 89 20 57 A8 85 8E 20   .. W(..
8070   57 A8 85 8F 20 5D A8 C6   W(.. →(F
8078   89 20 EB A2 D0 13 A1 8B   . k"P.!.
8080   F0 1F 20 57 A8 D0 FB 20   p. W(P¼
8088   57 A8 D0 FB 20 5D A8 D0   W(P¼ →(P
8090   BF C9 2E D0 24 24 89 10   ?I.P$$..
8098   B7 A1 8B D0 03 4C 26 81   7!.P.L&.
```

Floppywise Master provides a useful array of commands but little that is original. What's more, Floppywise Master seems to be designed by a committee. It has none of the coherent design of, say, a Clares' ROM like Brom Plus or Ramrod which provide essentially the same material.

Title : Advanced 1770 DFS
Supplier : Advanced Computer Products,
6 Ava House, High Street,
Chobham, Surrey GU24 8LZ.
Tel. (0276) 76545
Price : £34.50

Only ACP has the cheek to call all its products 'advanced'. However, the label does usually fit. The A1770DFS is available for Master or B+ owners or model B owners with 1770 upgrades, and provides all the functions of Acorn's 1770 DFS and lots more besides.

The A1770DFS is fully compatible with Acorn's 1770 DFS, and is as compatible with old DFS discs as that is. However, the A1770DFS can format and use double density (MFM) discs, 'large format' discs (both sides of a disc treated as one), and 'large catalogue' (62 files) discs.

These extras are handled with extensions to the existing commands (such as *DRIVE Ø 8Ø MFM), and clever automatic detection by the DFS.

An extension to the OPT command allows the formatting skew to be changed to suit your particular drives and to alter the disc options (double/single density or automatic and read an 8Ø track drive as 8Ø track, 4Ø track, or automatically).

That alone is enough to warrant the 'advanced' title but the A177ØDFS can also use sideways RAM banks as a fast RAM disc, configured as drive number 8 (though this can be changed to any drive number). All the other commands can then be used as normal on this RAM disc.

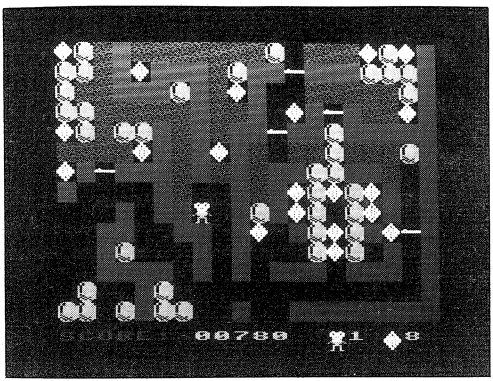
As many sideways RAM slots as you want can be used as a RAM disc and they need not be in adjacent slots. The *SRRAMS command sets it all up. This system is much preferable to Watford's Silicon disc system that requires either the DFS or Silicon filing system to be in use at any time. Treating the RAM as just another disc makes it much more useful.

ACP have also implemented on this ROM a number of useful commands already available to 177Ø Master users, and they have, quite sensibly, used the same command names as Acorn. Thus *APPEND will append entries to an EXEC file, and *SPOOLON will do the same with a spooled file. *CREATE can be used to generate an empty file of any specified length, while *MAP will tell you how the used sectors of a disc are distributed.

ACP's A177ØDFS is obviously intended to replace Acorn's 177Ø DFS, and for those model B and B+ users who are considering an upgrade but have not already purchased the standard 177Ø DFS, it looks like a good buy since it appears to contain all of the commands that Acorn provide in their DFS, plus a few more. The RAMdisc must be seen as an extremely useful added bonus - all thrown in for free - or almost.

PITFALL PETE

For this, the Christmas issue of BEEBUG, games writer Jonathan Temple has really gone to town. Not only is Pitfall Pete a superb game, but you can create and edit the game screens to provide endless fun.



'Pitfall Pete' is a 'Repton' style game in which you must collect all the diamonds on each screen, being careful not to trap yourself or the diamonds when you dislodge the many boulders.

As such, the game is high on strategy, and the five screens require a high degree of planning and forethought before you can complete them. The game also contains an easy-to-use screen designer, allowing you or others to try and conquer screens of your own design.

When you run the program you will be greeted by a menu. Pressing 'P' will allow you to play the game, and 'Q' will quit the program - the other options will be explained later.

The keys used to control Pete are the usual 'Z', 'X', '*' and '?' for left, right, up and down movement respectively. In addition, the game can be paused by pressing 'P', and continued with 'C'. The sound effects can be turned on and off with 'S' and 'Q'. Pressing 'R' will allow you to re-start a screen, which is useful if you become trapped (a common occurrence when you first start to play!), although you do lose a life. Pressing Escape will terminate the current game and return you to the menu.

If you should complete a screen, then a 3-digit 'pass code' for the next screen is revealed. If you wish to return to this screen, without the bother of playing the previous ones, selecting option 'C' from the menu ('Code entry') will allow you to enter the code and start at the relevant screen (Illegal codes are ignored).

EDITING AND DESIGNING SCREENS

Choosing option 'E' from the menu will allow you to edit the current screen (i.e. the one last played). Up to twenty screens can be designed and edited, although you will need to know the pass code for each one and select it using option C. When editing the screen, the cursor keys will move the cursor across the playing area. Features are inserted using the function keys - pressing f0 will insert a blank, f1 earth, f2 a wall, f3 a boulder, f4 a diamond and f5 a trapdoor. When you have finished editing press 'F' or use Escape.

Screens may contain as many diamonds as you like (they are counted automatically by the program), but remember, a screen without any diamonds cannot be completed!

Pete usually starts a screen from the top-left corner, but this can be changed by pressing f6. The default colour scheme used can also be changed, by using the keys 'D', 'R' and 'E' (diamonds, rocks and earth) to select the feature to have its colour changed, and then pressing the numbers 1 to 7 for the new colour. The feature currently being changed is displayed beneath the playing area. An additional feature is that Ctrl-f0 can be used to clear the screen to 'earth'.

Once you have edited a screen, or designed a new one, you will probably wish to save it. This can be done by using option 'S'. You will then be asked for the filename, the first screen to be saved and the last - allowing several screens to be saved at once.

LOADING

Loading previously saved screens is performed using option 'L'. Again, you will be asked for a filename, and the screen at which the stored screens are to be loaded. Be careful not to overshoot the screen storage area (by loading two screens starting from screen twenty, for instance) as this cannot be checked for!

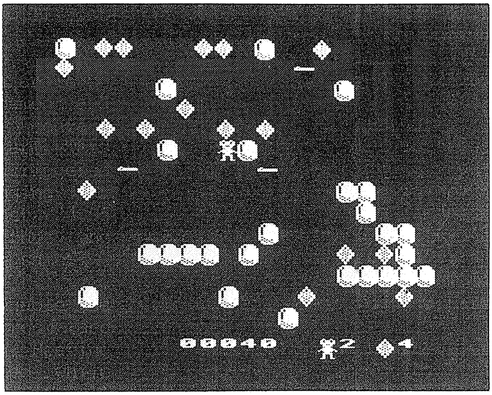
ENTERING THE PROGRAM

The program should be fairly straightforward to enter, although the data for the five screens (lines 359Ø-369Ø) should be entered carefully, or you may find yourself trying to complete an impossible screen! However, it is EXTREMELY important that you save the program, preferably twice, before trying to run it. It contains some assembler (lines 334Ø-336Ø; just enter these as you would normal Basic) and uses the indirection operators ? and ! frequently, so certain typing mistakes could prove fatal.

There is a lot of typing involved in entering the data for the five screens despite the extremely compact format used. Any or all of these lines (359Ø-367Ø) may be omitted, depending on how many screens you wish to have 'ready-designed'. However, line 369Ø MUST be included. When typing the program in, it may also be a good idea at first to omit line 338Ø, which disables the Escape key.

Since the program runs in mode 5, no awkward downloading or resetting of PAGE should be necessary - the program will run quite happily at &19ØØ (or &EØØ).

```
   1Ø REM PROGRAM PITFALL PETE
   2Ø REM VERSION B1.4
   3Ø REM AUTHOR  J. Temple
   4Ø REM BEEBUG  December 1986
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
  1ØØ ON ERROR GOTO 24Ø
  11Ø MODE 5:HIMEM=&41ØØ
  12Ø PROCinit
  13Ø PROCsetup
  14Ø REPEAT
  15Ø C%=FNmenu
  16Ø IF C%=1 PROCfile(Ø)
  17Ø IF C%=2 PROCedit
  18Ø IF C%=3 PROCfile(1)
  19Ø IF C%=4 PROCgame
  2ØØ IF C%=5 PROCcode
  21Ø UNTIL C%=6
  22Ø MODE7:GOTO 27Ø
  23Ø :
  24Ø IF ERR>127 CLS:PRINT''"FILING ERRO
R:"':REPORT:G%=INKEY(2ØØ):GOTO 14Ø
  25Ø MODE 7:PRINT'':REPORT
  26Ø PRINT" at line ";ERL
  27Ø *FX 4
  28Ø *FX 229
  29Ø END
  3ØØ :
 1ØØØ DEFPROCgame
 1Ø1Ø Z%=3:S%=Ø:P%=SS:SF=Ø
 1Ø2Ø REPEAT
 1Ø3Ø PROCscreen
 1Ø4Ø REPEAT
 1Ø5Ø PROCkeys:PROCman
 1Ø6Ø FOR N=1 TO 15Ø:NEXT
 1Ø7Ø UNTIL E%
```



```
 1Ø8Ø IF E%=1 Z%=Z%-1:SOUND 16,2,1ØØ,2:T
IME=Ø:REPEAT UNTIL TIME>1ØØ
 1Ø9Ø IF E%=2 PROCnext
 11ØØ UNTIL Z%=Ø OR E%=3
 111Ø VDU 28,4,13,15,11,12,26
 112Ø PRINTTAB(5,12);"GAME  OVER";
 113Ø TIME=Ø:REPEAT UNTIL TIME>2ØØ
 114Ø ENDPROC
 115Ø :
 116Ø DEFPROCkeys
 117Ø IF INKEY-17 THEN *FX21Ø,1
 118Ø IF INKEY-82 THEN *FX21Ø
 119Ø IF INKEY-56 REPEAT UNTIL INKEY-83
OR INKEY-113
 12ØØ IF INKEY-52 E%=1
 121Ø IF INKEY-113 E%=3
 122Ø ENDPROC
 123Ø :
 124Ø DEFPROCman
 125Ø V%=X%:W%=Y%:L%=Ø
 126Ø IF INKEY-98 IF X%>Ø PROCpush(-1):G
OTO13ØØ
```

```
 127Ø IF INKEY-67 IF X%<19 PROCpush(1):G
OTO13ØØ
 128Ø IF INKEY-73 IF Y%>Ø VDU31,X%,Y%-2:
CALL R%:L%=C%:IFC%<226 OR C%=228 Y%=Y%-2
:GOTO13ØØ
 129Ø IF INKEY-1Ø5 IF Y%<26 VDU31,X%,Y%+
2:CALL R%:L%=C%:IFC%<226 OR C%=228 Y%=Y%
+2
 13ØØ IF X%<>V% OR Y%<>W% PRINTTAB(V%,W%
);F$(Ø);TAB(X%,Y%);F$(6):IFL%=225 SOUND
16,-5,5,1
 131Ø IF D% PROCfall
 132Ø IF L%=228 SOUND17,1,1ØØ,1:Q%=Q%+1:
PROCscore(1Ø):IF Q%=D E%=2
 133Ø VDU31,X%,Y%-2:CALL R%
 134Ø IF C%=226 IF Y%>1 PROCfall:D%=TRUE
:J%=X%:K%=Y%
 135Ø ENDPROC
 136Ø :
 137Ø DEFPROCfall
 138Ø IF D%=Ø ENDPROC
 139Ø IF J%<>X% OR K%<>Y% D%=Ø:PROCdrop(
J%,K%-2)
 14ØØ ENDPROC
 141Ø :
 142Ø DEFPROCpush(D%)
 143Ø U%=X%+D%:VDU31,U%,Y%:CALL R%:L%=C%
 144Ø IF C%<226 OR C%=228 X%=U%:ENDPROC
 145Ø IF C%=226 IF U%>Ø IF U%<19 VDU 31,
U%+D%,Y%:CALL R%:IF C%<225 SOUND16,-12,6
,1:X%=U%:PRINTTAB(U%,Y%);F$(Ø);TAB(U%+D%
,Y%);F$(3);:PROCdrop(U%+D%,Y%)
 146Ø ENDPROC
 147Ø :
 148Ø DEFPROCdrop(A%,B%)
 149Ø VDU 31,A%,B%:CALL R%
 15ØØ IF C%<>226 OR B%+2=28 ENDPROC
 151Ø VDU 31,A%,B%+2:CALL R%
 152Ø IF C%>224 IF C%<231 ENDPROC
 153Ø IFC%=232 E%=1:ENDPROC
 154Ø VDU 31,A%,B%
 155Ø N%=Ø:REPEAT N%=N%+1
 156Ø IF VPOS=Ø C%=Ø ELSE VDU 11,11:CALL
 R%
 157Ø UNTIL C%<>226
 158Ø REPEAT
 159Ø PRINTTAB(A%,B%-(N%-1)*2) F$(Ø)
 16ØØ PRINTTAB(A%,B%+2) F$(3)
 161Ø FOR N=Ø TO N% DIV3
 162Ø SOUND Ø,2,15Ø,1:NEXT
 163Ø B%=B%+2:VDU 31,A%,B%+2:CALL R%
 164Ø UNTIL B%=26 OR C%=232 OR (C%>224 A
ND C%<231)
 165Ø IFC%=232 E%=1:ENDPROC
 166Ø IFC%=228 PROCslide(A%,B%)
 167Ø ENDPROC
 168Ø :
 169Ø DEFPROCslide(A%,B%)
 17ØØ LOCAL H%
```

```
 171Ø IF A%>Ø VDU31,A%-1,B%:CALL R%:IFC%
<225 H%=-1
 172Ø IF A%<19 VDU31,A%+1,B%:CALL R%:IFC
%<225 H%=1
 173Ø IF H% PRINTTAB(A%,B%);F$(Ø);TAB(A%
+H%,B%);F$(3);:PROCdrop(A%+H%,B%):PROCdr
op(A%,B%-2)
 174Ø ENDPROC
 175Ø :
 176Ø DEFPROCscreen
 177Ø VDU 23;1,Ø;Ø;Ø;Ø;
 178Ø S=P%:PROCmine
 179Ø COLOUR 2:PRINTTAB(Ø,29);"SCORE:";T
AB(14,29);F$(6);TAB(17,29);F$(4);
 18ØØ Q%=Ø:D%=Ø:E%=Ø:PROCscore(Ø)
 181Ø VDU 23;1,4Ø;Ø;Ø;Ø;
 182Ø ENDPROC
 183Ø :
 184Ø DEFPROCscore(N%)
 185Ø S%=S%+N%:COLOUR 3
 186Ø PRINTTAB(7,29);LEFT$("ØØØØØ",5-LEN
(STR$(S%)))+STR$(S%);TAB(15,29);Z%-1;TAB
(18,29);Q%;
 187Ø ENDPROC
 188Ø :
 189Ø DEFPROCnext
 19ØØ SF=SF+1:P%=P%+1:IFP%=21 P%=1
 191Ø VDU 28,4,15,15,11,12,26,17,2
 192Ø PRINTTAB(5,12);"NEXT  CODE";
 193Ø COLOUR 1:PRINTTAB(8,14);P$(P%);
 194Ø FOR N%=1 TO (3ØØ+SF*1ØØ) DIV25
 195Ø PROCscore(25):SOUND Ø,-1Ø,5,1
 196Ø FOR N=1 TO 5Ø:NEXT,
 197Ø RESTORE 2Ø4Ø:N=81:*FX 15
 198Ø FOR L%=1 TO 1Ø:READ A,D:N=N+A
 199Ø SOUND 1,-1Ø,N,D:SOUND 2,-5,N+48,D
 2ØØØ NEXT:REPEAT UNTIL ADVAL(-7)=15
 2Ø1Ø TIME=Ø:REPEAT UNTIL TIME>2ØØ
 2Ø2Ø ENDPROC
 2Ø3Ø :
 2Ø4Ø DATA Ø,4,8,4,8,4,4,4,8,8,-12,8,4,8
,-12,8,8,8,-16,8
 2Ø5Ø :
 2Ø6Ø DEFPROCfile(F%)
 2Ø7Ø B%=Ø:REPEAT VDU12,17,1,B%
 2Ø8Ø PRINT'''"ENTER FILENAME:"'
 2Ø9Ø COLOUR 3:INPUT ""F$:B%=7
 21ØØ UNTIL F$>""
 211Ø C$="LOAD":IF F% C$="SAVE"
 212Ø X%=FNinput(C$+" FROM SCREEN")
 213Ø IF F% Y%=FNinput("TO SCREEN"):IF Y
%<X% Y%=X%
 214Ø L$=STR$~(FNaddr(X%))
 215Ø C$=C$+" "+F$+" "+L$
 216Ø IF F% C$=C$+" "+STR$~(FNaddr(Y%)+2
89)
 217Ø $&7ØØ=C$:X%=Ø:Y%=&7:CALL &FFF7
 218Ø ENDPROC
 219Ø :
```

```
 2200 DEFPROCcode
 2210 CLS:PRINT''''"ENTER A CODE:"'
 2220 INPUT">--> "T%
 2230 N%=0:REPEAT N%=N%+1
 2240 UNTIL P%(N%)=T% OR N%=20
 2250 COLOUR 2
 2260 IF P%(N%)=T% SS=N%:PRINT'''"SCREEN
";SS;" CHOSEN":GOTO2280
 2270 PRINTTAB(4)'"ILLEGAL CODE"
 2280 TIME=0:REPEAT UNTIL TIME>200
 2290 ENDPROC
 2300 :
 2310 DEFPROCclear(S)
 2320 A%=FNaddr(S)
 2330 FOR N%=0 TO 276 STEP 4
 2340 A%!N%=&1010101:NEXT:A%?285=1
 2350 ENDPROC
 2360 :
 2370 DEFPROCedit
 2380 CLS:C%=282:S=SS:*FX4,1
 2390 REPEAT PROCmine
 2400 VDU 23;10,6;0;0;0;
 2410 A%=FNaddr(S):PROCcolours
 2420 D%=0:VDU 31,10,15
 2430 REPEAT G%=GET
 2440 IF G%=139 IFVPOS>1 VDU11,11
 2450 IF G%=138 IFVPOS<27 VDU10,10
 2460 IF G%=136 IF POS>0  VDU8
 2470 IF G%=137 IF POS<19 VDU9
 2480 IF G%=144 PROCclear(S):D%=1
 2490 IF G%=27  D%=2
 2500 N%=(G% OR16)-48
 2510 IF N%>0 IF N%<8 A%?C%=N%:PROCcolou
rs
 2520 G%=G% AND 223
 2530 IF G%=70 D%=2
 2540 IF G%=82 C%=282:PROCcolours
 2550 IF G%>67 AND G%<70 C%=212+G%:PROCc
olours
 2560 IF G%>1 IF G%<8 VDU11:PRINT F$(G%-
2);CHR$8;:A%?(POS+(VPOS-1)*10)=G%-2
 2570 IF G%=8 X=POS:Y=VPOS-1:VDU 31,A%?2
83,A%?284:PRINT F$(A%?285):A%?(A%?283+(A
%?284)*10)=A%?285:A%?285=A%?(X+Y*10):A%?
(X+Y*10)=1:A%?283=X:A%?284=Y:VDU 31,X,Y:
PRINT F$(6);CHR$8;
 2580 UNTIL D%:UNTIL D%=2:*FX 4
 2590 VDU 23;10,32;0;0;0;
 2600 ENDPROC
 2610 :
 2620 DEFPROCcolours
 2630 X=POS:Y=VPOS
 2640 VDU 19,1,A%?280;0;19,2,A%?281;0;19
,3,A%?282;0;
 2650 COLOUR C%-279
 2660 PRINTTAB(0,29);"CHANGING ";MID$("D
IAMONDSEARTH    ROCKS   ",(C%-280)*8+1,8)
;TAB(X,Y);
 2670 ENDPROC
 2680 :
 2690 DEFPROCmine
 2700 VDU 23;10,32;0;0;0;
 2710 A%=FNaddr(S):D=0
 2720 VDU4,12,19,1,A%?280;0;19,2,A%?281;
0;19,3,A%?282;0;
 2730 FOR V%=0 TO 13
 2740 FOR W%=0 TO 19:IF?A%=4 D=D+1
 2750 PRINT F$(?A%);:VDU 11:A%=A%+1
 2760 NEXT:PRINT:NEXT
 2770 X%=A%?3:Y%=A%?4
 2780 PRINTTAB(X%,Y%) F$(6)
 2790 ENDPROC
 2800 :
 2810 DEFFNaddr(S)=M%+S*290-290
 2820 :
 2830 DEFFNinput(P$)
 2840 COLOUR 1:PRINT'''P$+":";
 2850 Y%=VPOS:B%=0:COLOUR 3:REPEAT
 2860 PRINTTAB(LEN(P$)+2,Y%);SPC(20);TAB
(LEN(P$)+2,Y%);
 2870 VDU B%:INPUT ""S:B%=7
 2880 UNTIL S>0 AND S<21 AND INT(S)=S
 2890 =S
 2900 :
 2910 DEFFNmenu
 2920 VDU 4,12,20,17,1
 2930 FOR N%=0 TO 30
 2940 PRINTTAB(0,N%)STRING$(20,CHR$225)
 2950 NEXT:VDU 28,2,26,17,3,12,26
 2960 PRINTTAB(4,5);"PITFALL PETE";
 2970 COLOUR 2:PRINTTAB(4,7);STRING$(12,
CHR$228);TAB(4,8);STRING$(12,CHR$229);
 2980 RESTORE 3100
 2990 FOR N%=1 TO 6:READ L$,A$
 3000 COLOUR 1:PRINTTAB(4,9+N%*2);L$;
 3010 COLOUR 3:PRINT A$:NEXT:COLOUR 1
 3020 PRINTTAB(4,24);"CHOICE ?";
 3030 *FX21
 3040 REPEAT G$=GET$
 3050 C%=INSTR(" LlEeSsPpCcQq",G$)DIV2
 3060 IFG$=CHR$27 C%=2
 3070 UNTIL C%
 3080 =C%
 3090 :
 3100 DATA L,OAD SCREENS,E,DIT SCREEN,S,
AVE SCREENS,P,LAY GAME,C,ODE ENTRY,Q,UIT
 PROGRAM
 3110 :
 3120 DEFPROCsetup
 3130 VDU23,225,183,-4,91,239,186,111,24
5,95
 3140 VDU23,226,60,94,94,191,191,191,191
,191
 3150 VDU23,227,191,-1,191,223,171,87,10
6,60
 3160 VDU23,228,16,16,40,40,84,108,214,1
70
 3170 VDU23,229,214,108,84,40,40,16,16;
 3180 VDU23,230,-1;-1;-1;-1;
 3190 VDU23,231;0;0;64,191,-1
 32ØØ VDU23,232,36,9Ø,9Ø,126,36,6Ø,24,6Ø
 321Ø VDU23,233,126,189,24,6Ø,36,36,36,1
Ø2
 322Ø RESTORE 329Ø:FOR A%=Ø TO 6
 323Ø READ N%:FOR C%=1 TO N%:READ V%
 324Ø F$(A%)=F$(A%)+CHR$(V%):NEXT,
 325Ø ENVELOPE 1,133,8,4,8,3,1,1,126,Ø,Ø
,-1Ø,126,Ø
 326Ø ENVELOPE 2,1,Ø,Ø,Ø,Ø,Ø,Ø,9Ø,-1,-2,
-3,97,97
 327Ø ENDPROC
 328Ø :
 329Ø DATA 4,32,1Ø,8,32,6,17,2,225,1Ø,8,
225,6,17,2,23Ø,1Ø,8,23Ø,6,17,3,226,1Ø,8,
227,6,17,1,228,1Ø,8,229,6,17,1,231,1Ø,8,
32,8,17,1,232,1Ø,8,17,3,233
 33ØØ :
 331Ø DEFPROCinit
 332Ø DIM F$(6),P%(2Ø)
 333Ø M%=&41ØØ:R%=&7Ø
 334Ø P%=R%:[OPT 2
 335Ø LDA #135:JSR &FFF4
 336Ø TXA:ORA #96:STA &4ØC:RTS:]
 337Ø *FX 225,2
 338Ø *FX 229,1
 339Ø RESTORE 359Ø:PRINT'"PLEASE WAIT"
 34ØØ L%=1:REPEAT
 341Ø READ F%:IF F%=-1 GOTO35ØØ
 342Ø A%=FNaddr(L%):A%?282=F%
 343Ø READ A%?283,A%?284:A%?285=1
 344Ø FOR N%=1 TO 24:READ A$
 345Ø FOR Q%=1 TO 7+4*(N%=24) STEP 2
 346Ø H%=EVAL("&"+MID$(A$,Q%,2))
 347Ø ?A%=H% DIV36:A%?1=(H%-?A%*36)DIV6
 348Ø A%?2=(H%-?A%*36-A%?1*6)
 349Ø A%=A%+3:NEXT,:L%=L%+1
 35ØØ UNTIL F%=-1
 351Ø FOR B%=L% TO 2Ø:PROCclear(B%)
 352Ø A%?28Ø=3:A%?281=1:A%?282=6
 353Ø A%?283=1:A%?284=2:A%?285=1:NEXT
 354Ø R=RND(-1):FOR N%=1 TO 2Ø
 355Ø P%(N%)=N%*RND(5Ø)+1ØØ:NEXT
 356Ø VDU 23;1Ø,32;Ø;Ø;Ø;:SS=1
 357Ø ENDPROC
 358Ø :
 359Ø DATA 6,16,2,13A92E99,Ø44F329E,5556
56C2,31552B37,2B2B5B5Ø,5Ø566256,5Ø2C3234
,62586278,55554813,6CØCØ25Ø,5Ø685Ø59,4AA
5323E,2B732CØF,7F555556,563Ø4F3E,4F2D817
F,4A59C232,2B2B321Ø,63557A56,68485D81,4F
2E2B97,192B9856,56565456,ØE5B
 36ØØ :
 361Ø DATA 1,12,24,A34F2B2D,Ø73F9F74,9E5
15956,815B325C,3EØ12B38,7F2B313Ø,A14FAØ7
5,795Ø685C,564F2E5Ø,9DØ77338,A156315Ø,5D
55ØE2B,312C3487,9C4F9D55,3363164A,385B55
61,5E7CC19C,3F622B87,ØD4F565C,3156325Ø,7
57B2B2B,2B2B342E,612C9856,565E
 362Ø :
 363Ø DATA 6,17,2,81565656,564F2D3E,9F2B
7F74,2B2B5751,2D753133,32A4A58Ø,374F617A
,7C2B2B2C,732D9F56,5531577F,385C994F,5Ø9
72B3E,7AØ77432,682B5232,63312B2B,2C61342
B,4F5Ø4F73,3D562B74,8145562B,2B4F31A9,A5
7A6774,655Ø4F2B,2B2B2C4A,2B3B
 364Ø :
 365Ø DATA 6,17,12,815D8681,7A8733A4,815
8AB33,A3519F74,2B2B6374,75642B55,5Ø61347
3,3D2C7A32,555E735Ø,AØ2B2B2C,AC329E58,4A
4F5656,562B32AA,329D8181,6F7A5581,5281A5
82,392DA42E,8187972B,316F8281,85327451,6
42E822B,5D2D3856,4F2B56A9,2B37
 366Ø :
 367Ø DATA 6,11,2,5C623973,337A542B,7FAC
2B5B,2E585779,2B2C7C2C,56395D74,335D329D
,38642B5C,97584F3E,56507B74,514F572B,336
1322B,2B2B565C,5955589E,56972B2E,4F564F2
B,2E2C3873,39Ø15CAC,629F7382,517EØF82,64
2E862B,2A2B2C56,5656565Ø,ØE5F
 368Ø :
 369Ø DATA -1
```

POINTS ARISING POINTS ARISING POINTS ARISING POINTS

BEEBUG FILER ACCOUNTS (BEEBUG Vol.5 No.5)

Inexplicably, line 7Ø6Ø contains redundant information that will generate an error when this line is executed. Delete the 'd1,d2,' so that the line reads:

```
   7Ø6Ø FOR I=1 TO recp-1:PROCread(I,F1,1):PROCcheck(VAL(record$(3,1)),I):NEXT I
```

This applies to both magazine and cassete/disc versions.

Furthermore, some readers have encountered problems resulting from inherent inaccuracies in the Beeb's arithmetic routines. The principle indicator is a persistent '99' in the pence column of the statement balance. One solution, which appears to avoid the problems, has been suggested by Richard Brimson. This involves changing the '1ØØ*' multiplying factor in lines 352Ø, 46ØØ (twice), 476Ø and 782Ø to '1ØØØ*' followed by a '/1Ø' at the end of the particular instruction. Why a multiply by 1ØØØ followed by a divide by 1Ø should avoid the problem is unclear, but it appears to work and is recommended to any readers who encountered the problems described above. The problem will be investigated further and any additional information will appear in the next issue.

BEEBUG MAGAZINE is produced by BEEBUG Ltd.
Editor: Mike Williams
Technical Assistant: Alan Webster
Production Assistant:
Yolanda Turuelo
Secretary: Debbie Sinfield
Managing Editor: Lee Calcraft
Additional thanks are due to Sheridan Williams, Adrian Calcraft, Geoff Bains, John Yale and Tim Powys-Lybbe.

Editorial Address

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

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.

HINTS

There are prizes of £5 and £10 for the best hints each month, plus one of £15 for a hint or tip deemed to be exceptionally good.

Please send all editorial material to the editorial address above. If you require a reply it is essential to quote your membership number and enclose an SAE.

BEEBUG MEMBERSHIP

Send all applications for membership, membership renewals, membership queries and orders for back issues to the address below. All membership fees, including overseas, should be 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 £23.50 Middle East
£26.00 Americas & Africa. £28.00 Elsewhere

BACK ISSUES
(Members only)

Volume	Single issues	Volume sets (10 issues)
1	40p	£2.50
2	50p	£3.50
3	70p	£5.50
4	£1	£8.50
5	£1.30	–

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. Please note that the BEEBUG Reference Card and BEEBUG supplements are not supplied with back issues.

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

Hotline for All non-technical queries
i.e. orders, membership subscription, membership queries etc.

St. Albans (0727) 40303
Manned Mon-Fri 9am-4.30pm

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

Technical Queries
St. Albans (0727) 60263
Mon-Fri 10am-4pm

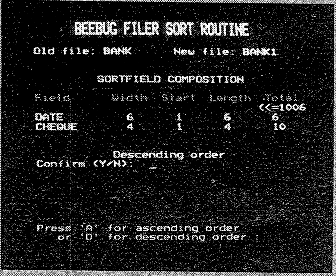
If you require members discount it is essential to quote your membership number and claim the discount when ordering.

Magazine Cassette/Disc

DECEMBER 1986
CASSETTE/DISC CONTENTS

PROGRAM CRUNCHER — a useful utility for squeezing the last drop of memory space from your machine.
BUILD YOUR OWN SIDEWAYS RAM — complete supporting program.
THE MASTER SERIES
48K/64K PRINTER BUFFER UTILITY — to make the most of the Master's sideways RAM.
VDU GRAPHICS MADE SIMPLE — complete set of useful drawing procedures.
EXEC FILES — from our Master hints.
MASTER CLOCK MANAGER — easy-to-use screen-based editor for time and date adjustment.
BUSINESS GRAPHICS — parts one and two combined to provide flexible editing of business data displays.
SORTING DATA FILES — complete sort program specially designed to work with the BEEBUG Filer database system.
FILE SECURITY — just the job to lock your files away securely at night.
BEEBUG WORKSHOP — all the procedures for data-driven routines in a complete demonstration.
FIRST COURSE — three complete examples of character formation including quad height characters.
PITFALL PETE — not only a superb game for our Christmas edition, but with a full games screen editor to provide you and your friends with a never-ending challenge.

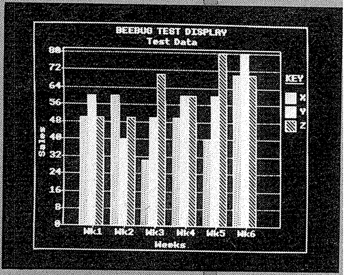
EXTRA FEATURES THIS MONTH
MAGSCAN — data for this issue of BEEBUG (Vol. 5 No. 7).



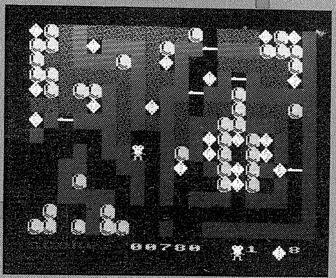
```
BEEBUG FILER SORT ROUTINE
Old file: BANK       New file: BANK1

        SORTFIELD COMPOSITION
Field        Width  Start  Length  Total
                                 <<=1006
DATE           6      1      6      6
CHEQUE         4      1      4      10

          Descending order
Confirm (Y/N): _



Press 'A' for ascending order
   or 'D' for descending order :
```

Sorting Data Files

Business Graphics



Pitfall Pete



All this for £3.00 (cass) £4.75 (disc) +50p p&p.
Back issues (disc since Vol. 3 No. 1, cass since Vol. 1 No. 10) available at the same prices.

Subscription rates	DISC UK	CASS UK	DISC O'seas	CASS O'seas
6 months (5 issues)	£25.50	£17	£30	£20
12 months (10 issues)	£50	£33	£56	£39

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

Cassette subscriptions can be commuted to disc subscription on receipt of £1.70 per issue of the subscription left to run.

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

DYNAMIC DISCS

FROM BEEBUG

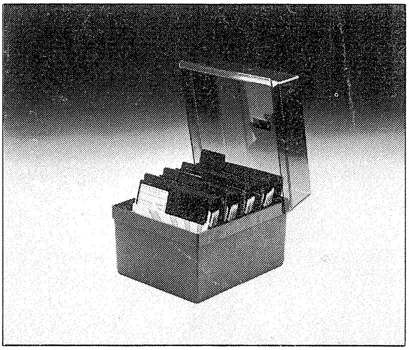
Our range of top quality 5¼" blank discs for the BBC Micro and Master 128 are now available at unbeatable prices.

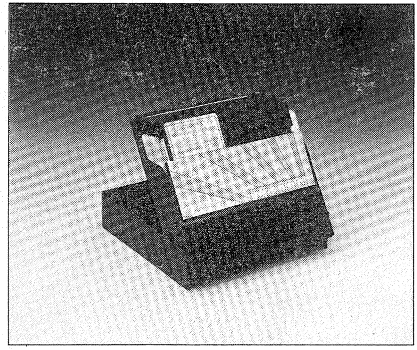
These discs are from the UK's leading disc manufacturer, individually tested and guaranteed for life. They are specially produced and boxed for Beebug. Don't settle for anything less than the best.

Beebug members may claim a 5% discount on the price. If you are not yet a member please phone for an application form.

48 TPI DOUBLE DENSITY

		PRICE	MEMBERS
10	S/S	£9.90	£9.40
25	S/S	£23.00	£21.85
50	S/S	£48.00	£45.60
10	D/S	£10.42	£9.90
25	D/S	£26.47	£25.15
50	D/S	£50.00	£47.50





96 TPI QUAD DENSITY

		PRICE	MEMBERS
10	S/S	£10.42	£9.90
25	S/S	£26.47	£25.15
50	S/S	£50.00	£47.50
10	D/S	£10.90	£10.35
25	D/S	£29.00	£27.55
50	D/S	£53.00	£50.30

5% MEMBERS DISCOUNT

- **Prices include FREE Storage Box**
- **Official Orders welcome**
- **ALL PRICES INCLUDE V.A.T.**
- UK Postage 10's — £1.00; 25's & 50's — £3.75

✂ — **ORDER FORM** — — — — — — — —

Please send me______Qty of ________________________ @ £_______ (Plus Postage £_______)

I enclose a cheque for £________

or Please debit my Access/Visa No. □□□□ □□□□ □□□□ □□□□

Name: ____________________________**Membership No:**____________________

Address: ____________________________________

____

________________________ **Post Code:**____________________

NOTE: To claim members prices it is essential to quote membership no. If not yet a member add **£12.90** membership fee, to claim members discount.

Send to: Beebug Mail Order, Dolphin Place, Holywell Hill, St. Albans, Herts AL1 1EX
Telephone Orders welcome. Tel: 0727 40303

ISSN – 0263 – 7561

Produced by Head Office Design (0782) 286375