

PRICE £1.30

JUNE 1987

AUTO-SAVE
UTILITY

BEEBUG

VOLUME 6 ISSUE 2

FOR THE BBC MICRO

THE WORLD BY
NIGHT AND DAY

BEEBUG

Volume 6 Number 2
June 1987

FEATURES

| | |
|---|----|
| Acorn's 32 Bit RISC | 5 |
| The World by Night and Day | 8 |
| Exploring Assembler (Part 1) | 13 |
| An Intelligent Auto-Save Utility | 20 |
| Marks Analysis for Teachers | 23 |
| The Comms Spot | 37 |
| The Master Pages | |
| Master Enhancements for | |
| Wally Corbett's World | 39 |
| Work Disc Generator | 41 |
| Master Hints | 42 |
| Auto-Booting ADFS Menus | 43 |
| How to write an Adventure Game (Part 1) | 46 |
| First Course | |
| Plotting for Effect | 51 |
| Workshop | |
| Turtle Graphics | 56 |

REVIEWS

| | |
|-------------------------------|----|
| Softstrip | 17 |
| Hard Disc for All? | 26 |
| High Resolution Graphics Dump | 50 |
| Battle of Fleet Street | 54 |
| New Books for Viewers | 58 |

REGULAR ITEMS

| | |
|--------------------|-------|
| Editorial Jottings | 3 |
| News | 4 |
| Supplement | 29-36 |
| Points Arising | 36 |
| Postbag | 60 |
| Hints & Tips | 61 |

HINTS AND TIPS

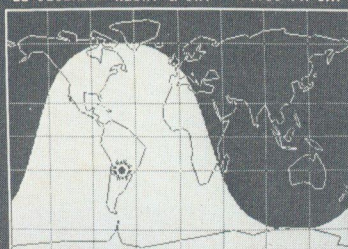
GENERAL

Editing Basic with Wordwise
Ibico LTR-1 and Pound Signs
Return Battering
Watford's Diagnostic Disc and
Shadow RAM Board
Printing with Tilde
Flashy REM Statements
Printed Output Only

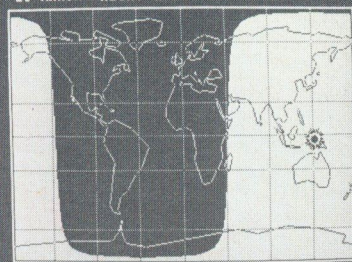
MASTER

ADT Toolkit Basic Load
Grouping ADFS Directories
Double Bar Function Keys
Where it's at
Speeding up Repeated *LOADs
CMOS RAM Display

22 DECEMBER THE WORLD BY NIGHT & DAY 4.00 PM GMT



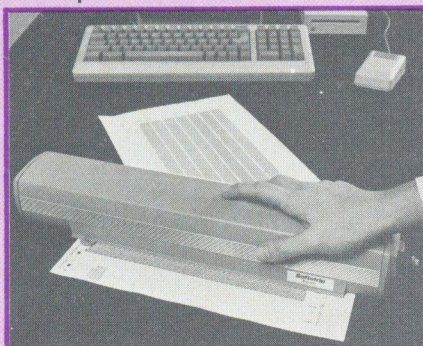
15 MARCH THE WORLD BY NIGHT & DAY AT: 3.00 AM GMT



Another Date and time or Quit ? (R/O)

Night and Day

Softstrip



Marks Analysis

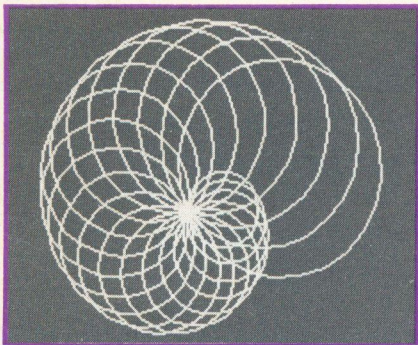
| | |
|-----------------------|---------|
| Total no. of items | 202 |
| Maximum possible mark | 80 |
| Spread of marks | 70 to 9 |
| Mean | 37.71 |
| Standard deviation | 12.09 |
| New mean ? | 40 |
| New deviation ? | 12 |

STANDARDISED MARKS in column E:

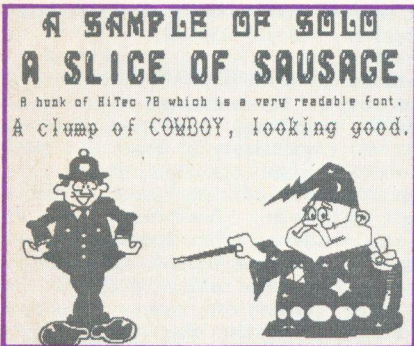
| | |
|---------------|-------|
| New Mean | 40.00 |
| New deviation | 12.00 |

A = Mark C = Frequency
B = Position D = Cumulative %

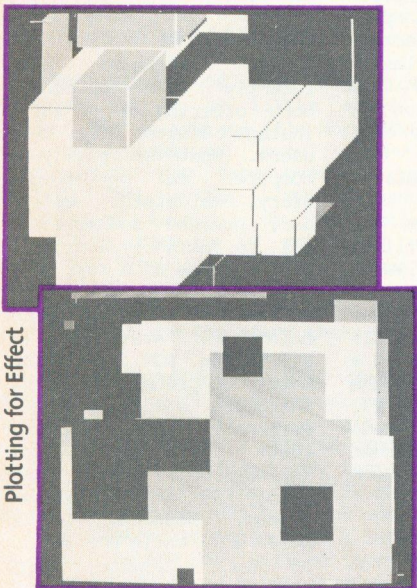
| A | B | C | D | E |
|----|---|---|----|----|
| 70 | | | | |
| 69 | 1 | 1 | 14 | 72 |
| 68 | | | | * |
| 67 | 2 | 1 | 24 | 69 |
| 66 | | | | * |
| 65 | | | | * |
| 64 | 3 | 1 | 40 | 66 |



Turtle Graphics



Battle of Fleet Street



Plotting for Effect

EDITORIAL JOTTINGS

ACORN'S RISC MACHINES

The future of micros looks increasingly likely to be bound up with RISC (Reduced Instruction Set Computer) technology, in which Acorn claim to have a two year lead. Acorn's next generation of micros, scheduled for release this year, will certainly be based on this concept. This month, Ian Burley (Features Editor of Micronet) has written a special in-depth appraisal of Acorn's RISC technology to set the scene for the new range.

This is clearly a highly important step for Acorn, and one which BEEBUG will be watching and reporting on with considerable interest. We believe that the new range is likely to be launched this June, and we shall be carrying a full and detailed report at the earliest opportunity. Exciting times seem to be ahead for Acorn and all Acorn users.

NEW DATABASE FROM BEEBUG

Because of the continuing success of Masterfile, the first database system for the BBC Micro, we are developing a totally new and advanced database on ROM to include many of the features requested by existing Masterfile users. This new database will continue alongside Masterfile II which will remain in its present form. The new ROM will be totally compatible with the BBC Micro, Master, Compact and RISC machines, and provide for the transfer of datafiles between the new system and many other applications such as Inter-Link range, Wordwise and View.

The new system has been designed for the user, not the programmer, with a choice of 40 or 80 column screen and the ability to use existing Masterfile datafiles. Particular features include user defined record layout and user defined menus, range checking of data, and comprehensive arithmetic procedures. The software can be readily customised by the user to suit any particular task, and all the steps involved can then be repeated automatically. Also included will be fast location of records through indexing, searching and sorting, a comprehensive report generator and many other features. Watch for our future product announcements later this year.

PROGRAM/REVIEW CLASSIFICATION

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

Computer System
Master (Basic IV)



Compact (Basic IV)



Compact (Basic VI)



Model B (Basic II)



Model B (Basic I)



Electron



Filing System
ADFS



DFS



Cassette



Tube Compatibility

Tube



Truly Superior

Superior Software, have launched another three games compilations, this time of Superiors' own games (previous collections included Acorn games). The Superior Collection Volumes 1 & 2 cost £9.95 (dual cassette), £11.95 (disc) and £14.95 (3.5" disc for the Compact). These include favourites such as Repton and Repton2, Star Striker, Karate Combat and Crazy Painter. Volume 3 (cassette or 3.5" disc only) is a similar collection for the Electron. Superior are on (0532) 459453.

LEGO Control

LOGOsoft have recently released a new extended version of their Master LOGO for the Master and BBC Micro. It is a control language for LEGO models that's easy to use, fast, and powerful. The Technical LOGO pack comprises a full LOGO interpreter, technical LOGO extension disc plus two toolkit discs, three different manuals and reference card - it costs £45. The extension alone is £20. Contact Chris Squire on 01-891 0989.

Arraymate

Arraymate is a utility EPROM which facilitates the manipulation of Basic arrays. Features include sorting, rotating, erasing, and passing arrays to procedures. The cost is £15.75 from Fosse Systems. Phone: (0666) 54477.

Logo Workshop

BLUG - the British Logo Users Group - are holding a practical Workshop called "Logo & Computers in

Education and Youth Work" at Southlands College in Wimbledon from 24th - 27th August. The cost is £59 non-residential and £89 residential. Apply to BLUG, PO Box 79, Walsall WS5 3RW.

A'Maze'ing Generator

XOR Designer is a Maze Generator from Logotron. It has been designed for players of XOR their maze game, and allows the design of new mazes and character icons. For further details contact Wendy Frazer of Logotron on (0223) 323656.

Your Obedient Servant

Avon Data Systems - a company specialising in business software have released a comprehensive package entitled "The Hotel Servant" for the hotel trade. It runs on the range of BBC micros and will handle all the reception work of a hotel with up to 80 rooms. The package retails at £175 inclusive of VAT from Avon Data Systems, 184 Badminton Rd, Downend, Bristol BS16 6NP.

Llangaffo 1871 Census Data

Llangefni School are compiling data from the 1871 census for the village of Llangaffo in Anglesey. Data will be available in Enform or ViewStore format or as a straight ASCII file. The cost, including a potted history of the village, will be £7.99. Cheques should be made payable to I.T.I.S and sent to Llangefni School, Room 43, Llangefni, Gwynedd.

Grand Prix Construction Set

This latest release from Superior Software features an icon driven track

designer, and you can add extra hazards such as chicanes and hump-back bridges. The game features dual cockpit view, race against a friend or challenge the computer. It comes complete with 18 Grand Prix tracks from round the world and costs £14.95 on 5.25" and £14.95 on 3.5" disc (prices inc VAT). More details from (0532) 459453.

Mrs Sheffield's Planner

Beeb-Planner is a menu-driven, project-planning package that runs on a Beeb with sideways RAM, or a Master. It uses "Critical Path Analysis" and is ideal for industry, business or anyone who needs to plan a task or project. The cost is £39.95. Further details from Mrs E.J. Sheffield, 8 Langdon Close, Camberley, Surrey GU15 1AQ.

BEEB Sidekicks

Two packages have recently been announced both offering a PC style instant desktop for Beeb users. Whatever task your computer is performing, diary, calculator, calendar, notepad and more can all be instantly available at the press of a key.

Sideset from Maze Technology is available at £45.94 (on ROM), while the more comprehensive Genie (hardware & software) from Permanent Memory Systems costs £79.35 (both prices include VAT). Maze Technology is at 11 Braemar Avenue, Neasden, London NW10 0DY, while PMS are at 38 Mount Cameron Drive North, St Leonards, East Kilbride G74 2ES.

ACORN'S 32 BIT RISC

Ian Burley explores Acorn's RISC chip set, and considers the potential for Acorn's next generation of micros.

A short while ago, viewers to BBC's Micro-Live programme were treated to the first public appearance of a prototype of one of Acorn's next generation of micros. These are based on an entirely new type of processor called a RISC processor. RISC stands for Reduced Instruction Set Computer. Essentially the idea is to use a central processor chip which has been pruned down to respond to a much smaller set of instructions than on conventional state-of-the-art processors, and to use special techniques such as "pipelining" to ensure that it operates at great speed. The philosophy behind the RISC processor is to ensure that almost all instructions take place in just a single clock cycle.

The Beeb's 6502 can execute certain instructions in as little as 2 clock cycles, but on average it's more like 3 or 4, and sometimes more. Some 280 instructions can take over 10 machine cycles. One way the RISC processor is capable of keeping up its efficient processing rate is by employing what's known as "pipelining". This means that while the processor is executing one instruction it is also decoding the next, and fetching the one after that.

In reality the Acorn RISC Machine, or ARM, as Acorn call their new processor, is incredibly fast. It is a full 32 bit processor and contains what is known as a Barrel Shifter providing very fast logical shift operations. The old 6502 can shift an 8 bit operand left or right one bit at a time - taking anything from 2 to 7 clock cycles. The ARM can utilise its Barrel Shifter to shift 32 bits either way up to 32 bits at a time, all in a single cycle!

A RISC processor can do a massive amount of work in just one clock cycle. By contrast, an optimised piece of 6502 machine code can execute at an average of just over 3 cycles per instruction. A Beeb runs its 6502 at 2MHz, but let's give the Beeb a head start in the comparison. Let's take a Master Turbo running at 4MHz. The Turbo card, for argument's sake can run its code at about 1 million instructions per second (MIPS). Not bad. That's already about 10 times faster than say a Sinclair Spectrum. But the ARM is already up to 8 times faster than that in terms of MIPS (peak), as the current production ARM chips use a clock rate of 8MHz.

So we're already looking at something that can execute machine code instructions 8 times faster than a Master Turbo, or 16 times faster than a standard Beeb, and when you start to compare BBC Basic Benchmarks this advantage doubles and even trebles according to the application program. Optimised ARM assembler can reach peaks of performance nearly 100 times faster than the standard Beeb's 6502. We're talking about minicomputer power, on one chip.

Ian Burley is Features Editor of Micronet 800

ACORN'S RISC CHIP

Ever since the Atom, Acorn has had a love affair with the 6502, and Acorn's Roger Wilson, claimed to be the world's foremost expert on the 6502, has performed miracles with his implementations of Basic using this processor. Wilson has been a major figure in the design of the ARM, and it is he who wrote the ARM's Basic interpreter.

About two and a half years ago Acorn decided to go for RISC technology, and that meant 32 bits. The processor was to be designed completely from scratch, in house, with Californian based chip manufacturers VTI providing the manufacturing know-how and first production samples. The Acorn RISC processor has just 44 basic instructions, and all instructions are a maximum of one 32 bit word in length. There are sixteen 32 bit user definable registers and the processor was designed from the start with multi-user applications in mind, having a "privileged supervisor mode" in addition to its "user mode".

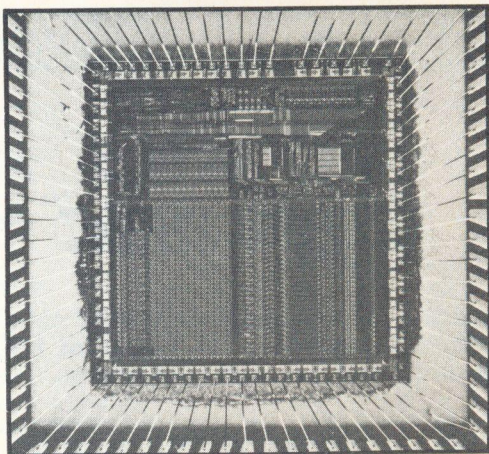
The ARM address bus is 26 bits wide making a maximum memory address range of 64 megabytes (or 16 mega-words). Interestingly the ARM is completely "hard wired"; in other words its design logic is fixed and can only be altered by a fairly drastic redesign. Most modern 16 and 32 bit microprocessors use a "microcode" system, which is akin to a machine code program running inside the processor itself, determining the processor's function. If the designers find a bug, they just alter the processor's microcode ROM. This is all very well when you're designing a chip, but it makes the silicon in that chip very complex and hence more expensive to manufacture.

The ARM's second important advantage is its physical simplicity, and although the chip is housed in an imposing square packaged 84 pin chip carrier, the chip itself is modest in size and uses very conservative (read inexpensive) silicon technology. Even in small production quantities, Acorn are claiming that the ARM costs just a 1/5th of that of its main rivals, the Intel 80386 and the Motorola MC68020 - both conventional processors.

The ARM is a VERY clever, VERY efficient piece of silicon. Acorn claim, conservatively, that the 8MHz ARM can match the benchmark speed of a 16.6MHz MC68020 - the 32 bit version of the 68000 as found in the Atari ST and Commodore Amiga, etc. Motorola are only too aware of RISC's pluses, and they have recently announced their own RISC design, the 78000.

SUPPORTING THE ARM

When the BBC Micro was first unveiled back in 1981 it was hailed as the most advanced budget priced microcomputer of its time. The Beeb was much admired for its graphics and sound, as well as its then blistering speed. Today the Commodore Amiga has the winner's laurels when it comes to sound and graphics, but Acorn will be out to regain that prize. It is now obvious that the ARM CPU will be used in Acorn's replacement of the old BBC Micro. However, Acorn decided that they would follow the same philosophy regarding the support chips necessary to accompany the ARM, and design them in house too. There are three such devices at present, and the signs are that there will be more. There is a Memory Controller (MEMC), Video Controller (VIDC) and an Input/Output



Close-up of Acorn RISC Chip

Controller (IOC). Probably the most interesting of these is the VIDC, as it has quite astonishing graphics capabilities, and to top that there was enough room on the chip to include a 16 channel full waveform sound system.

Each graphics pixel can have up to 8 bits determining colour from a palette of 4096 colours. That means up to 256 out of 4096 colours displayed on-screen at one time. Resolutions are amazing too - the VIDC can deal with hundreds of kilobytes of video RAM, meaning pixel resolutions of over 1000x1000 in several colours! Exactly how much of this potential video power will be put to use in any machine which Acorn releases has not been made public at this stage. It will depend on the, as yet unseen, machine operating system, but Acorn have decided that above all, the system will be completely "soft" - programmable, not overly limited by the hardware itself.

The MEMC memory controller keeps tabs on all the support chips, the ARM processor itself and the memory - be it RAM/ROM or virtual memory (hard disc storage treated as "slow" RAM). The MEMC chip can deal with up to 4 megabytes of "real" RAM, providing memory refresh etc. All critical system timing signals like the processor clock are provided by the MEMC.

The IOC is a sort of super VIA (Versatile Interface Adaptor) and this

chip looks after keyboard handling, interrupt timing, peripheral selection, reset control, bus control, etc. Together, the ARM, VIDE, MEMC and IOC form the heart of Acorn's current prototype computer, code named the A500. This machine, with a prototype operating system affectionately called Arthur, is being used as a development tool for the final production machine. It's my guess that this machine will be unveiled around the end of this year's third quarter, at the September PCW Show perhaps. BBC TV's Micro Live program recently showed the A500 ARM prototype doing very clever things on-screen, though the software still appeared to be some way from being finished, and reliable sources suggest that a BBC Master was being used behind the scenes to control a Winchester hard disc!

Some lucky folk outside of Acorn are already using ARM computers. Software and third party hardware developers around the country have development machines. These are either in the form of a 2nd processor for the Beeb or an IBM PC card. The development systems come in either 2 or 4 megabyte sizes and the PC card features a MEMC chip as well as the ARM. Some software houses even have the full A500 machine, with the full chip-set on board.

ARM SOFTWARE

If Acorn decide you're worth a development system (all enquiries are vetted) and you've paid your £4000, besides the hardware you do get a very comprehensive suite of languages. C, Lisp, Prolog, Fortran, Assembler and of course BBC Basic are all bundled.

BBC Basic, being Roger Wilson's baby, has undergone many subtle transformations in its ARM version V guise. Briefly, there's an ARM assembler built in, multi-line IF-THEN-ELSE, WHILE-WEND, CASE-ENDCASE, an extended LISTO, plus various new toolkit and graphics commands and some new logical operators. ARM Basic has to be the most comprehensive and fastest interpreted Basic anywhere!

In tests on an ARM second processor, which doesn't benefit from a MEMC controller remember, we ran various simple loops and saw a speed increase factor of between 20 and 80 times compared with Basic II on a BBC Model B! A Basic version of the Sieve of Eratosthenes finding prime numbers ran typically 35 times faster. An ARM assembler version of the Sieve program proved 250 faster than ARM Basic!

A favourite demo at Acorn involves running a 6502 emulator on the ARM, executing 6502 machine code practically as fast as a Beeb. I've seen the Master's 6502 coded Basic IV running on the emulator as though it was a second processor to a Master. Drawing pretty pictures across the Tube - you just can't see any difference in speed between the emulator and the real thing.

It seems likely that some sort of BBC Micro emulator will be provided for production models of the ARM computer, and there's even talk of an MSDOS/PCDOS emulator for IBM compatibility. But that's enough speculation for now!

The question is, are Acorn and Olivetti going to guarantee the success that the ARM deserves? Will the new machine create the same stir that the original BBC Micro did six years ago? Acorn aren't just on the verge of launching a new computer, they're on the verge of competing as a microprocessor company, competing with Intel, Motorola, National Semiconductors, and more.

The success of the ARM doesn't lie wholly with Acorn's own machines. The processor and perhaps the whole chip set must succeed in other company's products, OEM third party developments. Assuming the ARM performs as well as it seems to at this stage, the chip should have a bright future - assuming Acorn gets the marketing right! Acorn already have a two year lead in RISC technology, and it's a lead they cannot afford to lose.

STOP PRESS

Since the above article was written we have now learned that Acorn have produced a second version of the ARM chip set using '2 micron' rather than 3 micron' silicon. This offers a staggering throughput of 15 MIPS and includes a 'Booth's Multiplier' to handle a two-operand 32 bit multiply in hardware. Roger Wilson is believed to be re-writing the relevant sections of Basic V to take advantage of this new version of the ARM chips which will be used in all production machines.

THE WORLD BY NIGHT AND DAY

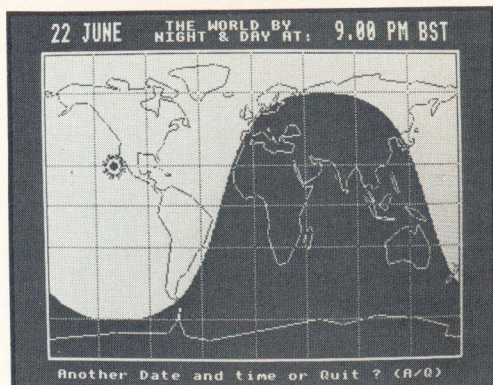
The entire world displayed on the screen showing night and day for any time and date in the year, and if you have a Master (or shadow RAM) then the display can be updated in real time too. W.C. Corbett explains how it's done.

You've seen those clocks in stock-brokers' and travel agents' offices. They tell you the time in Los Angeles, New York, or Tokyo, but they don't give you the feel of the places. Is the day just fading towards sundown, or is the city sweltering under an overhead midday sun? This program will draw a map of the world showing graphically where the sun is in the sky or where it's night at every spot on earth.

As, in the course of the year, the earth orbits the sun, the axis about which it rotates is angled about 66.5 degrees from the plane of its orbit. Thus, at our midsummer, the North Pole "leans" towards the sun, and, as the earth spins, everywhere above latitude 66.5 degrees north (the "Arctic Circle") is in daylight all the time - the "Land of the Midnight Sun". Northwards from the Equator everywhere has progressively more than 12 hours' daylight; everywhere south progressively less until, below latitude 66.5 degrees south the sun never rises. In the northern midwinter, it is the opposite; and at the equinoxes, around March 22 and September 22, the day is just 12 hours long everywhere on earth.

MASTER AND COMPACT

On the Master 128, this program can be further extended (using the Master's real time clock and extra RAM) to produce a real-time display of the world that is automatically updated every 10 minutes. The Master and Compact (and all users with the GXR ROM) have access to an extended graphics capability which will also improve the program. These enhancements are described in this month's Master pages.



If you had a flair for mechanics, and rather more than enough room to swing a cat, you could make a three-dimensional clockwork model on which you could see, for any day and any time, just which half of the spinning globe was in sunlight, and which half in darkness. The old astronomers did just that; but most of us just don't have the skill. Instead, this program produces a detailed screen display of the world by night and day.

Type in the program (keeping strictly to the line numbers as given) and save away to tape or disc. When you run the program, you may enter any date and any time of day for which you want to see the display. The screen will then show, on a map of the world, the current division between daylight and darkness. The date and time (GMT or BST) selected are shown at the top of the screen, and a flaming "sun" sits at the point on the map where the sun is overhead at that time.

If you kept re-running the program covering a period of a year, you would see the track of the "overhead" sun rise from the Tropic of Capricorn, cross the Equator to the Tropic of Cancer, and sink back again. The map itself is drawn on the simplest possible projection (Mercator), with lines of latitude and longitude equally spaced. On this projection you would see the boundary between day and night straighten out from a sinuous curve, with daylight "above" it, in midsummer, until it simply follows two lines of longitude, at the equinoxes. Then it gradually returns to a smooth curve, with the daylight area now "below" it, at midwinter.

Anyone who has looked at a yachting almanac will know that the rules of celestial motion are complex, and of course, many short cuts are taken and approximations made in designing an essentially simple program like this one.

HOW THE PROGRAM WORKS

Here's how it's done. The program assumes that the vernal equinox is always March 22nd. PROCdate counts the days from March 22nd to the selected date. PROCgha multiplies by 360/365 (or 366 in a leap year) to derive an approximation of the sun's "Greenwich Hour Angle". This is used in PROCdec to work out the sun's approximate declination (the latitude at which the sun is overhead at noon).

The "power-house" of the program is PROCctm in which, using the sun's declination, the longitude of the "terminator" (the line between daylight and darkness) is calculated at noon, GMT, for each of 45 equal divisions of latitude over a sector of 90 degrees of longitude. The formula (line 2690) interprets spherical geometry and the 45 calculations, which are passed to the array tha(K%), take several seconds to complete each time the screen is refreshed. Fortunately, we need only calculate for one 90-degree sector, since for the other three sectors of the 360-degree globe the line of the terminator is a mirror image, east/west or north/south. PROCctm draws the terminator by successively reading the array 'forwards' and 'backwards' and successively adding or subtracting the north/south component.

Before the terminator can be drawn it needs to be "offset" westwards according to the time of day. PROCchr works out this displacement, but a refinement is needed. Because the earth's orbit is slightly elliptical, the sun appears up to 16 minutes 'fast' or 'slow' in relation to clock time at different seasons of the year. An inaccuracy of this order would show up on the display. FNeqt refers to a 'look up table' (from line 3370) and supplies the correction.

When the program is first run, and a date and time selected, the screen switches to mode 4 and PROCfr draws a frame within which the graphics window is

contained. Once its co-ordinates have been calculated, the terminator is drawn, and the area of daylight, above or below according to the time of year, is filled by PROCfl.

There are a few days around the equinoxes when the lines are straight up and down and the 'fill' cannot flow from one side of the screen to the other. On these days, PROCdf1 is called to supply a second starting point. When the 'fill' is complete, PROCmap draws the continents and major islands, and PROCsun places the sun at its 'overhead' position.

The program assumes that you will enter the time as GMT or BST as appropriate, and automatically adjusts for this whilst Summer Time is in force during 1987. If your computer survives into 1988 or later, you will need to amend the DATA in line 3470. Just count the days from March 22nd (day 0) on which British Summer Time begins and ends.

The world map of course demands a mass of co-ordinates, beginning at line 4000. For those who want to simplify, or embellish the display, or even devise another map projection, these and all the other co-ordinates calculated within the program are straightforward degrees of latitude and longitude (in that order). All are converted to screen co-ordinates by FNxpx and FNypx.

Now you can leave your micro running on a summer's evening and fascinate your friends who come to dinner! While you have your sherry you'll see that the sun is overhead Mexico City; you'll see the line of warm darkness advancing north and west over Europe; and if it turns into a really good party you might even see the dawn marching down from the northeast, while the sun burns down over Taiwan!

PROGRAM NOTE

The BBC micro (model B) has no built in 'flood fill' graphics facility. This is programmed as the two procedures PROCfill and PROCfill1, using the PLOT77 command, to fill the area of daylight. These procedures are specific to this program and any attempt to use them as a general purpose fill routine will probably fail.


```

10 REM Program WORLDAY
20 REM Version B2.5(B)
30 REM Author Wally Corbett
40 REM Beebug June 1987
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO 270
110 DIM i$(4),tlat(45),tha(45),cp 8
140 MODE7
150 f%=FNintro
160 PROCset:tb$=set$
170 MODE 4:VDU 23,1,0;0;0;0;0:PROCinit:
PROCfr:PROCwait
190 REM * Single data and time
200 PROCscr:IF FNagain=ASC("A") GOTO 1
40 ELSE MODE 7:END
260 :
270 IF ERR=17 MODE 7:CLS:END
280 CLS:REPORT:PRINT" at line ";ERL:EN
D
290 :
1000 DEFPROCwait
1010 COLOUR 0:COLOUR 129
1020 PRINTTAB(11,10)" Wait a moment..."
"
1030 COLOUR 1:COLOUR 128
1040 ENDPROC
1050 :
1060 DEFFNagain
1070 *FX15 1
1080 PRINTTAB(1,31);"Another Date and t
ime or Quit ? (A/Q)";
1090 REPEAT:G=GET AND &DF:UNTIL G=ASC("
A") OR G=ASC("Q"):=G
1100 :
1250 DEFPROCscr
1260 PROCdate:PROChr:PROCgha:PROCdec
1270 PROCctm:CLG:PROCdtm
1280 PROCfl:PROCdlat:PROCdlon:PROCmap
1300 PROChead:PROCsun:ENDPROC
1310 :
1360 DEFFNintro
1370 A$=CHR$141:B$=CHR$133:C$=CHR$134:D
$=CHR$130
1380 FOR I%=0 TO 1:PROCcen(A$+D$+"W O R
L D A Y"):NEXT
1390 PRINT:PROCcen(B$+"The World by Nig
ht and Day")
1400 PRINT:PROCcen(D$+"by Wally Corbett
")
1410 =1
1500 :
1510 DEFPROCset
1520 PRINT":Y=VPOS
1530 PROCcen("Enter details:"):PRINT'TA
B(5)"Day"TAB(5)"Month (first 3 letters)
"TAB(5)"Hours (24-hour clock)"TAB(5)"M
inutes"
1540 :

```

```

1550 FOR I=1 TO 4:INPUT TAB(30,Y+I+1)":
" i$(I):NEXT
1560 FOR I=1 TO 4:IF I=2 GOTO 1580
1570 IF LEN(i$(I))<2 i$(I)="0"+i$(I)
1580 NEXT
1590 nm$="":mx%=ASC(LEFT$(i$(2),1)):IFm
x%>96 mx%=mx%-32
1600 nm$=nm$+CHR$(mx%)
1610 FOR K%=2 TO 3:mx%=ASC(MID$(i$(2),K
%,1)):IFmx%<97 mx%=mx%+32
1620 nm$=nm$+CHR$(mx%):NEXT
1630 set$="xxx,"+i$(1)+" "+nm$+" 1987."
+i$(3)+" "+i$(4)+"00":ENDPROC
1640 :
1650 DEFPROCinit
1660 yq%=500:xg%=640:dlat=4.8:dlon=3.5
1670 bf%=yq%-90*dlat:tf%=yq%+90*dlat
1680 lf%=xg%-180*dlon:rf%=xg%+180*dlon
1690 VDU19,0,4;0;19,7,3;0;:ENDPROC
1700 :
1710 DEFPROCchr
1720 hr$=MID$(tb$,17,2):mn$=MID$(tb$,20
,2)
1730 hr%=VAL(hr$):tn%=hr%*60+VAL(mn$)
1740 RESTORE 3480:READ sta$,stz$
1750 IF sdy>=sta$ AND sdy<=stz$ z$="BST
" ELSE z$="GMT"
1760 IF z$="BST" tn%=tn%-60
1770 IF tn%<720 m$="AM" ELSE m$="PM":EN
DPROC
1780 :
1790 DEFPROCdate
1800 day$=MID$(tb$,5,2):mon$=MID$(tb$,8
,3):yr$=MID$(tb$,14,2)
1810 IF VAL(yr$)MOD4 = 0 yd=366 ELSE yd
=365
1820 mdays=FNmon:sdymdays+VAL(day$)-81
1830 IFsdy<1 sdy=sdymdays
1840 date$=FNnz(day$):ENDPROC
1850 :
1860 DEFPROCgha
1870 gha=sdymdays*360:agt=FNneg:ghq=gha+a
gt:ENDPROC
1880 :
1890 DEFFNmon
1900 dno%=0:RESTORE4290
1910 REPEAT:READmo$,nd$,mth$:dno%=dno%+
nd$:UNTILmo$=mon$
1920 dno%=dno%-nd$:=dno%
1930 :
1940 DEFPROChead
1950 COLOUR1:dt$=date$+" "+mth$
1960 IFhr%>12 hs$=STR$(hr%-12):GOTO1980
1970 hs$=hr$
1980 dh$=FNnz(hs$):pt$=" "+dh$+" "+mn$+
" "+m$+" "+z$
1990 PROCdh(dt$,0,0):xt%=38-LEN(pt$):PR
OCdh(pt$,xt%,0)
2000 COLOUR1:PRINTTAB(14,0)"THE WORLD B
Y";TAB(14,1)"NIGHT & DAY":ENDPROC

```



```

2010 :
2020 DEFPROCdh(A$,K,L)
2030 A%=&A:X%=&70:Y%=&0:D=&70
2040 FOR N=1 TO LEN(A$):B$=MID$(A$,N,1)
: ?D=ASC(B$):CALL(&FFF1)
2050 VDU23,240,D?1,D?1,D?2,D?2,D?3,D?3,
D?4,D?4,23,241,D?5,D?5,D?6,D?6,D?7,D?7,D
?8,D?8
2060 PRINT TAB(K+N,L);CHR$(240);TAB(K+N
,L+1);CHR$(241):NEXT:ENDPROC
2070 :
2080 DEFPROCcen(A$)
2090 PRINTSPC(19-LEN(A$)DIV2);A$
2100 ENDPROC
2110 :
2120 DEFFNnz(day$)
2130 t$=LEFT$(day$,1)
2140 IF t$="0" d$=RIGHT$(day$,1) ELSE d
$=day$
2150 =d$
2160 :
2170 DEFPROCfr
2180 GCOLOR,1:MOVElf%-4,bf%-4:PLOT5,lf%-
4,tf%+4:PLOT5,rf%+4,tf%+4
2190 PLOT5,rf%+4,bf%-4:PLOT5,lf%-4,bf%-
4:MOVElf%,bf%:PLOT5,lf%,tf%
2200 PLOT5,rf%,tf%:PLOT5,rf%,bf%:PLOT5,
lf%,bf%
2210 VDU24,lf%+4,bf%+4,rf%-4,tf%-4;:END
PROC
2220 :
2230 DEFPROCfl
2240 IFtn%<60 fl%=356 ELSE fl%=360
2250 X=ns+FNxpx(fl%):Y=yq%:PROCfloodf(X
,Y)
2260 IFgha>=358 OR gha<3 OR (gha>177.5
AND gha<181.5) THEN PROCdf1
2270 GCOLOR,0:MOVElf%+2,bf%+2:PLOT5,lf%+
2,tf%-2:PLOT5,rf%-4,tf%-2
2280 PLOT5,rf%-4,bf%+2:PLOT5,lf%+2,bf%+
2:ENDPROC
2290 :
2300 DEFPROCdf1
2310 IFtn%<356 X=xg%-626:PROCfloodf(X,Y
)
2320 IFtn%>1080 X=xg%+626:PROCfloodf(X
,Y)
2330 ENDPROC
2340 :
2350 DEFPROCdlat
2360 GCOLOR,1:RESTORE 3380:REPEAT:READla
t:ylat=FNypx(lat)
2370 MOVElf%,ylat+yq%:PLOT21,rf%,ylat+y
q%:UNTILlat=-9999:ENDPROC
2380 :
2390 DEFPROCdlon
2400 GCOLOR,1:RESTORE 3390:REPEAT:READlo
n:ylon=FNxpx(lon)
2410 MOVEylon+xg%,tf%:PLOT21,ylon+xg%,b
f%:UNTILlon=-9999:ENDPROC

```

```

2420 :
2430 DEFFNypx(lat):ylat=lat*dlat:=ylat
2440 DEFFNxpx(lon):xlon=lon*dlon:=xlon
2450 :
2460 DEFPROCmap
2470 GCOLOR,1:RESTORE 4000:FOR M%=0 TO 1
6
2480 READ ylt,xln:y=FNypx(ylt):x=FNxpx(
xln):MOVEx+xg%,y+yq%
2490 REPEAT:DRAWx+xg%,y+yq%:READ ylt,xl
n:y=FNypx(ylt):x=FNxpx(xln)
2500 UNTIL ylt=-9999:NEXT:ENDPROC
2510 :
2520 DEFPROCdec
2530 IFgha<=90 dec=SIN(RAD(gha))*23.5
2540 IFgha>90 AND gha<=180 dec=SIN(RAD(
180-gha))*23.5
2550 IFgha>180 AND gha<=270 dec=SIN(RAD
(gha-180))*-23.5
2560 IFgha>270 dec=SIN(RAD(360-gha))*-2
3.5
2570 ENDPROC
2580 :
2590 DEFPROCsun
2600 GCOLOR,2:X=xg%-FNxpx(nd%):Y=yq%+FNy
px(dec):C=0
2610 FOR psi=0 TO 2*PI STEP PI/16:C=C+1
:L=28+(C MOD 4)*3
2620 MOVEX,Y:PLOT5,X+SIN(psi)*L,Y+COS(p
si)*L:NEXT
2630 GCOLOR,3:FOR K%=16 TO 20 STEP 2:PRO
Ccircle(X,Y,K%)
2640 NEXT:ENDPROC
2650 :
2660 DEFPROCctm
2670 llat=90-ABS(dec):ideg=llat/45:rdec
=RAD(ABS(dec))
2680 tha(0)=90:tlat(0)=0:tha(45)=180:tl
at(45)=llat
2690 FOR K%=1 TO 44:tlat(K%)=K%*ideg:rl
at=RAD(tlat(K%))
2700 tha(K%)=DEG(ACS((-SIN(rlat)*SIN(rd
ec))/(COS(rlat)*COS(rdec))))
2710 NEXT:ENDPROC
2720 :
2730 DEFPROCdtm
2740 GCOLOR,3:IFtn%<=70 J%=-1 ELSE J%=0
2750 nd%=(tn%-720+qn%)/4:ns=xg%-FNxpx(3
60)-FNxpx(nd%)
2760 X=ns:yd=FNypx(tlat(45)):IFghq<180
Y=yq%-yd ELSE Y=yq%+yd
2770 IF J%=-1 MOVE X-FNxpx(360),Y ELSE
MOVE X,Y
2780 IF z$="BST" AND tn%<=60 J%=-1 ELSE
J%=0
2790 FOR K%=J% TO 1:og=ns+FNxpx(180)+FN
xpx(K%*360)
2800 FOR L%=44 TO 0 STEP -1:X=og-FNxpx(
tha(L%)):yd=FNypx(tlat(L%))
2810 IFghq<180 Y=yq%-yd ELSE Y=yq%+yd

```



```

2820 PLOT5,X,Y:NEXT
2830 FOR L%=1 TO 45:X=og+FNxpx(tha(L%))
-FNxpx(180):yd=FNypx(tlat(L%))
2840 IFghq<180 Y=yq%+yd ELSE Y=yq%-yd
2850 PLOT5,X,Y:NEXT
2860 FOR L%=44 TO 0 STEP -1:X=og-FNxpx(
tha(L%))+FNxpx(180):yd=FNypx(tlat(L%))
2870 IFghq<180 Y=yq%+yd ELSE Y=yq%-yd
2880 PLOT5,X,Y:NEXT
2890 FOR L%=1 TO 45:X=og+FNxpx(tha(L%))
:yd=FNypx(tlat(L%))
2900 IFghq<180 Y=yq%+yd ELSE Y=yq%-yd
2910 PLOT5,X,Y:NEXT:NEXT:ENDPROC
2920 :
2930 DEFFNegt
2940 RESTORE 3420:REPEAT:READqh%,qn%:UN
TILqh%>=gha
2950 qd=qn%/4:=qd
2960 :
3090 DEFPROCccircle(x,y,r)
3110 IF r<24 st=PI/4 ELSE st=PI/16
3120 VDU 29,x;y;
3130 MOVE r,0
3140 FOR theta=0 TO 2*PI STEP st
3150 DRAW r*COS(theta),r*SIN(theta)
3160 NEXT theta
3170 VDU 29,0;0;
3180 ENDPROC
3190 :
3200 DEFPROCfcircle(x,y,r)
3220 IF r<24 st=PI/4 ELSE st=PI/16
3230 VDU 29,x;y;
3240 FOR theta=0 TO 2*PI STEP st
3250 MOVE 0,0
3260 MOVE r*COS(theta),r*SIN(theta)
3270 PLOT 36,r*COS(theta+st),r*SIN(thet
a+st)
3280 NEXT theta
3290 VDU 29,0;0;
3300 ENDPROC
3310 :
3320 DEFPROCfloodf(X%,Y%)
3340 PROCfill(X%,Y%)
3350 ENDPROC
3360 :
3370 REM lines of lat/long
3380 DATA -180,-135,-90,-45,0,45,90,135
,180,-9999
3390 :
3400 REM Equation of time
3410 DATA 0,-7,3,-6,6,-5,9,-4,12,-3,15,
-2,19,-1,23,0,26,1,30,2,36,3,45,4,68,3,7
3,2,79,1,85,0,89,-1,95,-2,99,-3,105,-4
3420 DATA 111,-5,126,-6,139,-5,145,-4,1
50,-3,153,-2,156,-1,159,0,162,1,167,2,17
0,3,175,5,180,7,186,9,191,11,198,13,207
3430 DATA 15,219,16,233,15,239,13,246,1
1,251,9,256,7,261,5,265,3,272,1,277,-1,2
83,-3,286,-5,292,-7,297,-9,304,-11,308

```

```

3440 DATA -12,314,-13,326,-14,336,-13,3
1,251,9,256,7,261,5,265,3,272,1,277,-1,2
83,-3,286,-5,292,-7,297,-9,304,-11,308
3450 DATA -12,314,-13,326,-14,336,-13,3
42,-12,346,-11,353,-9,359,-8,365,-7
3460 :
3470 REM BST begins/ends
3480 DATA 7,216
3490 :
3990 REM Map
4000 DATA 37,-8.5,37.5,-9.5,43.5,-9,43,
-2,46,-2,48.5,-4.5,49.5,0,53.5,5.5,57.5,
9.5,54,9,54.5,21,59,24,60,30,60,21,65,25
,65,22,60,18,55,14,60,10,58,7.5,60,5.6
4010 DATA 63,5.5,70,20,71,25,68,40,66,3
9,66.5,34,64,38,68.5,49,73,70,70,74,75.5
4020 DATA 90,77,113,73.5,111,71,140,72.
5,142,68.5,170,70,172,68,180,65,180,60
4030 DATA 170,60,164,51,156,57,156,62,1
63,62,156,59,152.5,59,141,54.5,135,52,14
0.5,48.5,140,43,135,43,130.5,39.5,127,35
4040 DATA 130,33.5,126,39.5,125,39,117.
5,30,121,23.5,116,22,107.5,18.5,106,12,1
09.5,8.5,105,12.5,100,10,95.5,5,103.5
4050 DATA 1.5,104,9,98,16,92,15,94,22.5
,90,15.5,80.5,10,80,7,82,6.5,80,10,80,8,
77,21,72,26,66.5,27,52.5,30,50,29.5,48,2
4,52,26,56,22,60,17,55,12.5,43,28,35
4060 DATA 19,37,10.5,44,12,52,-6,39,-16
,40.5,-33,28,-34,19.5,-17,12,-10,13,3.5,
10,6.5,3,4.5,-8,13,-16.5,21,-16.5,36,-6
4070 DATA 37,11,34,10,30,20,32,20.5,31.
5,34.5,36.5,36,37,27,40.5,26,41.5,41.5,4
7,38,47,31.5,42,28.5,40,23,37,23,36.5,21
.5,46,13,45,12,40,18.5,38,16,44,8.5,40,0
,36,-1.5,36,-6,37,-8,-9999,-9999
4080 DATA 46,-53.5,47.5,-59.5,51.5,-57,
50,-66.5,46.5,-71,49,-65,45,-62.5,43.5,-
71,38,-75,35.5,-75.5,31.5,-81.5,26.5,-80
,25,-81.5,30,-84,30,-94,32,-97.5,22
4090 DATA -97.5,18,-94.5,21.5,-87,16,-8
8.5,15,-83,11,-83.5,8.5,-77,12,-70.5,-52
.5,-7.5,-35,-22.5,-42.5,-26,-49,-38,-57.
5,-42,-65,-52,-68.5,-55,-65,-54,-73,-48
4100 DATA -75.5,-18,-70,-5,-81.5,7,-77
.5,8,-79.5,7,-81,12.5,-87.5,16.5,-95,17.
5,-102,31.5,-115,23,-110,28,-115,37.5
4110 DATA -112.5,48.5,-125,60,-140,60,-
153,55,-158,53,-167.5,58,-158.5,62,-166.
5,68,-167,72,-156,70,-128,68,-110,72,-12
5,75,-121,72.5,-105,68,-105,68,-110,69.5
4120 DATA -85,71,-90,75,-87.5,69,-67,66
.5,-61.5,62.5,-66,64.5,-78,68.5,-74,70,-
84,64,-82,60,-95,57,-92,55,-82,51.5
4130 DATA -80.5,52,-79,62.5,-78,60,-64,
52,-56,46,-53.5,-9999,-9999
4140 DATA -22.5,114,-14,127,-12.5,137,-
15,135.5,-18,140.5,-11,142.5,-24,152,-30
,153,-37.5,150,-37.5,140.5,-34,133.5,-32
,128.5,-35,116,-22.5,114,-9999,-9999

```


EXPLORING ASSEMBLER

(Part 1)

This month sees the start of a brand new series by Lee Calcraft for all beginners to machine code programming. Now's the time to find out what it's all about.

I want to begin this series by briefly setting the context. Why should anyone with BBC Basic at their fingertips bother with machine code and assembler, which is notoriously more difficult to use? The answer comes in several parts. Firstly, as you will already be aware, machine code is vastly quicker than Basic, and there are many applications which need this speed. A machine code sort in a database can speed up an average sorting operation from several hours to a few minutes. Screen handling is another classic application. The game of Elite could not have been written in Basic; its extremely fast screen displays would have been utterly out of the question in Basic.

But it is not just speed which machine code offers the inventive programmer. There are many programming tasks which interface with the machine's operating system in ways which are not possible from Basic. You cannot for example write star commands in Basic, nor can you read a disc catalogue from Basic, so if you are writing a disc menu program, you must resort to machine code.

But this series is not just for people who wish to write their own machine code utilities. It is hoped that it will also benefit those who occasionally make use of machine code or assembler programs in magazines such as our own, without really understanding what is going on. Furthermore, assembler programming is an enjoyable and rewarding pastime in its own right, and this alone is more than sufficient reason for its pursuit.

ASSEMBLY LANGUAGE

But first of all, what is assembler? Assembler is a language in its own right,

just as Basic is a language. But assembly language is very much closer to the raw machine code of the Beeb's 6502 microprocessor. In fact each assembler instruction is the direct equivalent of one instruction in 6502 machine code. The names of the assembler instructions have even been chosen to act as mnemonics for the set of machine code instructions. For example the assembler instruction "RTS" means "ReTurn from Subroutine". Its equivalent in machine code is a single number of value 96.

LEAST SIGNIFICANT DIGIT

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|----|----|----|-----|-----|-----|-----|-----|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | A | B | C | D | E | F | 10 | 11 | 12 | 13 |
| 2 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| 3 | 1E | 1F | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 4 | 28 | 29 | 2A | 2B | 2C | 2D | 2E | 2F | 30 | 31 |
| 5 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 3A | 3B |
| 6 | 3C | 3D | 3E | 3F | 40 | 41 | 42 | 43 | 44 | 45 |
| 7 | 46 | 47 | 48 | 49 | 4A | 4B | 4C | 4D | 4E | 4F |
| 8 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| 9 | 5A | 5B | 5C | 5D | 5E | 5F | 60 | 61 | 62 | 63 |
| 10 | 64 | 65 | 66 | 67 | 68 | 69 | 6A | 6B | 6C | 6D |
| 11 | 6E | 6F | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 |
| 12 | 78 | 79 | 7A | 7B | 7C | 7D | 7E | 7F | 80 | 81 |
| 13 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 8A | 8B |
| 14 | 8C | 8D | 8E | 8F | 90 | 91 | 92 | 93 | 94 | 95 |
| 15 | 96 | 97 | 98 | 99 | 9A | 9B | 9C | 9D | 9E | 9F |
| 16 | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
| 17 | AA | AB | AC | AD | AE | AF | B0 | B1 | B2 | B3 |
| 18 | B4 | B5 | B6 | B7 | B8 | B9 | BA | BB | BC | BD |
| 19 | BE | BF | C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| 20 | C8 | C9 | CA | CB | CC | CD | CE | CF | D0 | D1 |
| 21 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | DA | DB |
| 22 | DC | DD | DE | DF | E0 | E1 | E2 | E3 | E4 | E5 |
| 23 | E6 | E7 | E8 | E9 | EA | EB | EC | ED | EF | F0 |
| 24 | F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | FA |
| 25 | FB | FC | FD | FE | FF | 100 | 101 | 102 | 103 | 104 |

Decimal to Hexadecimal Conversion Table 1

In practice, a 6502 machine code program consists of nothing more or less than a series of numbers in the range 0 to 255, each value having a different meaning for the computer's micro-processor. And for reasons which may become clearer as the series progresses, these single numbers are usually expressed in hexadecimal rather than decimal notation. The hex equivalent of 96 is &60 (the "&" indicating that the number is in hex). Table 1 gives hex and decimal equivalents.

Table 2 shows a complete machine code program, expressed both in decimal and hexadecimal notation. Neither listing is either instructive or memorable, and it is for this reason that an assembler is so essential for programming in machine code. An assembler allows you to write programs

in terms of memorable mnemonic instructions, which are subsequently translated into machine code for you.

| Hex | Decimal |
|-----|---------|
| A9 | 169 |
| 41 | 65 |
| 20 | 32 |
| EE | 238 |
| FF | 255 |
| 60 | 96 |

Table 2
Machine code program listing

Unlike most other micros, the Beeb has an assembler built into its Basic ROM (Read Only Memory), and one cannot praise too highly the design philosophy which resulted in this powerful combination. Because the Beeb's assembler is so easy to use, it is worth typing in and testing a brief assembler program even at this very early stage in our foray into machine code. We can usefully illustrate a number of principles in this way, and we will use the same program as that given in machine code in Table 2.

```
100 REM PRINTS "A"...>asstest3
110 FOR pass=0 TO 1
120 P%=&900
130 [
140 OPT pass*3
150 LDA #65
160 JSR &FFEE
170 RTS
180 ]
190 NEXT
```

Table 3
Assembler Program

ASSEMBLING A PROGRAM

Thus, without further ado, you should type in the very short listing which appears in Table 3, treating it exactly as you would any Basic program. Now, save it away BEFORE running it. It is particularly important to save assembler listings before attempting to execute the resultant machine code, since running machine code which contains bugs may hang your machine completely, with the resultant loss of the contents of user memory. If you now RUN this Basic program, you will see it generate a listing on the screen, similar to that given in Table 4.

```
0900
0900
0900 OPT pass*3
0900 A9 41 LDA #65
0902 20 EE FF JSR &FFEE
0905 60 RTS
```

Table 4
Output from the Beeb's Assembler

The listing is a representation of the code which the program has assembled in response to your RUN instruction. This listing has three fields (or columns). The far left contains 6 four-digit hex numbers. To the right of these are up to three pairs of hex characters representing the machine code which the assembler has generated for each assembler instruction. Then in the right hand column is the original assembler instruction mnemonic. We will look at the meaning of these instructions in a moment. But first let us look more carefully at what happens when the program is RUN.

EXECUTING THE MACHINE CODE

Now although you have RUN the program, you may be surprised to discover that the machine code is not executed at this point. What the assembler has done is to translate the mnemonic instructions into a series of 6502 machine instructions, and to place these in an area of user RAM (Random Access Memory), as defined in the assembler program. It is line 120 of the program where the RAM starting point is defined, and we have set it to &900 (this is equivalent to 2304 in decimal).

To execute the assembled machine code, we can use directly a Basic command:
CALL &900

If you try this, you should see a letter "A" printed at the cursor. You can repeat this as many times as you wish without any need to reassemble the code.

SAVING THE CODE

So far we have saved the assembler code as a Basic program, and you can load this in and run it any time that you wish to reassemble the machine code routine. But it is also possible to save the assembled machine code directly using:

*SAVE name 900 906

This simply saves the code residing between memory locations &900 and &905 inclusive, giving it the filename "name".

To load it back in again, use:

*LOAD name

No addresses are necessary when loading back in if you wish to place the code exactly where it was saved from. Now if you execute:

CALL &900

you should again see the "A" displayed at the cursor, indicating that the code has loaded and run correctly.

1. Enter your program in assembler mnemonics using Basic's built-in assembler.
2. Save it away just as you would with any other Basic program.
3. Run the program. This will assemble the program in machine code and place it directly into the computer's memory at an address starting at the value held in the variable P%.
4. Optionally, save the machine code directly using *SAVE.
5. Execute the machine code using CALL (or *RUN etc).

Table 5
Major steps in using
the resident assembler

There is a further command in the armoury of Basic which can be useful here. Try executing:

*RUN name

This will load in the machine code from cassette or disc, and automatically execute a CALL to &900. Now the "A" should appear without you having to type in the CALL command. In fact, disc users can simplify this even further to:

*name

and the code will again be loaded in and executed. As you can appreciate, this is very convenient because it allows the user to build up a library of additional machine code commands on disc, which may be called in the same manner as any other machine-resident star command.

At this point in the proceedings, it is probably worth glancing at Table 5. This summarises the five main stages involved in using the Beeb's resident assembler. It does not give a full picture of the process of assembly language programming however, because it misses out

a number of important tasks. In particular, programs need to be carefully planned and coded before using the assembler. And even once executed for the first time, they still need to be thoroughly tested and debugged.

A CLOSER LOOK AT THE LISTING

We have not as yet made any attempt to understand the program which we have been running. In fact the assembly listing contains just three instructions, on lines 150 - 170. The first (LDA #65) tells the processor to Load the Accumulator, its principal register, with the number 65. This is in fact the ASCII code for the letter "A" (see glossary). You can see how the assembler has handled this instruction if you look again at Table 4. It shows that the first line of the assembly listing has been assembled as two hex numbers, &A9 and &41. The first is the 6502 instruction to load the accumulator with a number, and the second (&41) is the hex equivalent of 65 decimal (=ASCII "A"), indicating the value of the number to be loaded. The hash symbol (#) preceding the 65 indicates that it is the number 65 itself which is to be loaded, rather than the contents of memory location 65.

The assembler instruction on line 150 (JSR &FFEE) tells the processor to Jump to a SubRoutine at the address in memory of &FFEE. This is actually an address in the computer's ROM operating system (see glossary) which contains a very useful piece of code. The code has been christened OSWRCH (for Operating System WRite Character), and it sends to the screen whatever printable character happens to be in the 6502 accumulator when it is called. We have put the ASCII code for the letter "A" into the accumulator in the immediately previous instruction, so OSWRCH sends an "A" to the screen. The last instruction is one which we have already come across: RTS. This signifies ReTurn from Subroutine, and in this case it returns control to Basic.

ASSEMBLER PARAPHERNALIA

So much for the assembler code itself, but what of the lines between which it is sandwiched? The assigning of P% on line 120, we have already briefly mentioned. P% is a special variable which is taken by the assembler to hold the address at which the programmer wants the code to be assembled. In this case we have set it to

&900, which is usually free on most versions of the Beeb, unless you are simultaneously saving data files (not program files) to cassette. We will use this address in our early experiments with assembler, but we will discuss other places to store machine code programs later in the series.

The next line, line 130, contains a solitary left hand bracket (which appears as a left-pointing arrow in mode 7). This is used by BBC Basic to indicate the beginning of a piece of assembler code. As you can see, a right hand bracket (a right-pointing arrow in mode 7) is used in line 180 to indicate the end of that code. Everything within the brackets must be an assembler instruction, and everything outside must be in Basic, otherwise you will get a syntax error. Try inserting:

```
145 REM Basic not allowed here
RUNning the assembly listing will now give an error.
```

Lastly, but by no means least, you will see that all the code which we have so far discussed is enclosed in a FOR-NEXT loop at lines 110 and 190. This ensures that the whole assembly operation is executed twice in succession. This is called "two-pass" assembly, and the reason for it is that it allows the assembler to cope with certain kinds of forward jumps in a very simple manner, as we shall see at a later stage in the series.

You will note that line 140 contains an odd-looking statement. The effect of this is to assign to a special "pseudo-variable" named OPT a value of 0 at the first pass, and 3 at the second. This just tells the assembler that we only require the reporting of errors, and the assembly listing display on the second pass, once the assembler has fully come to grips with our code.

It is worth stressing at this point that the assigning of P% should occur OUTSIDE the assembler square brackets (since it is performed from Basic), but INSIDE the two-pass assembler loop. If you place the assignment of P% outside the FOR-NEXT loop, P% will be at different values for each pass, with a consequent foul-up. This is because the assembler updates P% as it assembles, so that it always holds the address of the next instruction to be assembled. You can check

this by typing:

```
PRINT P%
```

after the assembly has taken place. It will have the value 2310, whereas it was assigned the value 2304 decimal (i.e. &900 hex) at the start of the program.

Next month we will introduce a sub-set of assembler instructions, and put them to work in a variety of applications. P

ASCII

American Standard Code for Information Interchange. A numeric code in which each printable character is represented by a numeric code in the range 32 to 126. Codes outside this range are used for control purposes, and on the Beeb for graphics characters. See the BEEBUG Giant Reference Card for a table of ASCII codes.

Assembler

An assembler is a piece of software which accepts programs in the form of mnemonics, and converts this into machine code.

Machine code

The raw programming code of the microprocessor. It takes the form of a series of instructions stored in memory which the processor obeys in sequence.

Operating System

A complex piece of machine code resident within the computer, and stored on ROM in the BBC micro. It handles all low-level activities, such as checking the keyboard, sending characters to the screen etc. In fact the Beeb's operating system contains many useful routines which may be called from assembler. They are partly documented in the User Guide, and more fully in the Advanced User Guide for the BBC Micro, and in the Master Reference Manual Part one.

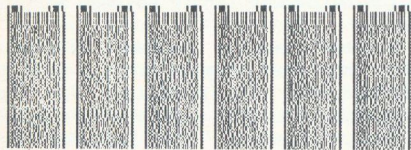
RAM

Random Access Memory. Chip-based memory which can be both read and written to. All information held in RAM is lost when the power is removed.

ROM

Read Only Memory. Non volatile memory used for storing machine code programs and data on a chip. Information stored in ROM chips is not lost when power is removed.

Table 6
Glossary of Terms



Product : Softstrip reader, inc. stand, power supply, cables, software on disc and assorted softstrips.

Supplier : Softstrip International
53 Bedford Square,
London WC1B 3DB.
Tel. 01-631-3775

There have been several attempts to bridge the gap between what humans and computers can read. These vary from the peculiarly shaped numbers on cheques to the bar codes you find on nearly every product sold in supermarkets.

Bar codes have caught on in a big way and some bright spark in the States had the idea of adapting the system for storing and distributing software. The result is the Softstrip, a strip of dots and lines which can be printed with a dot matrix printer (such as the Epson) and read with a special reader, looking something like a Penman plotter in a cylindrical box.

Softstrip readers are now available for a variety of micros, although each strip is machine-specific, so there is still no easy way to swap programs from machine to machine. The BBC micro version, which is supplied for use with the standard DFS, comes complete with programs to read and print the strips, and a book full of pre-printed strips with 'useful' programs on them.

and sockets for power and data leads. Power is supplied via an integral transformer in the mains plug, and the data lines connect straight into the RS423 socket of the BBC Micro. A stand is also supplied for the reader, which protects the head from dust when the unit is not in use.

The reader works by firing an infra-red beam from the head as it moves slowly along the strip, and interpreting the changes in the pattern reflected from the paper. The actual frequency of the infra-red beam has been chosen to minimise interference from coloured inks. It is quite possible to read a Softstrip that has been written over in felt tip pen.

The head reads the reflections using eight rotating lenses, and the whole procedure is software controlled. All you have to do is to ensure that the reader is lined up square to the strip. As the head trundles down the inside of the reader (it reads through an inch wide slot in the bottom of the casing), a stepper motor and internal electronics ensure that each line of the Softstrip is scanned, between four and sixteen times. Each line includes its own check bit, so small errors in the printed strip can be eliminated by error checking.

The system is very easy to set up and is well explained, at both beginner and expert levels, in the accompanying user manual. All three sections of documentation are supplied as loose leaf A4 sheets, which you can clip into a ring binder. This technique makes it

particularly easy to remove individual sheets when you want to read the strips printed on them.

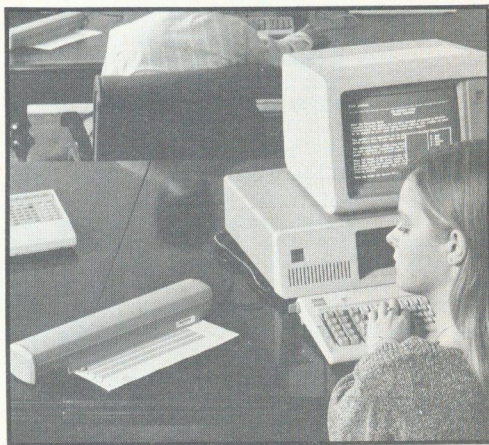
The only program supplied on disc is the Softstrip reader. Once you've loaded this, you can read anything else from the strips supplied. Each strip is printed with location markers, a dot and a short line, which you use to ensure the reader is correctly aligned. The rest of the process is completely automatic, with a byte count on the screen showing how much of the program has been input. When the whole strip has been read, you are prompted for a filename to save the file to disc. There's no option to run a program straight from memory, which is probably wise.

A 6" strip printed lithographically, as are all those supplied with the reader, can hold slightly over 2K of Basic program, machine-code or text, and any data beyond that length has to be read from more than one strip. Again, the software takes care of multi-strip reads, and once you've given it a filename, will keep adding to that file on disc until complete. The header information, at the start of the strip, holds details of the length of the strip, and which number it is in a multi-strip program. The program automatically prompts for further strips if the software you're reading runs over more than one.

Strips produced by printing on an Epson (or dot compatible) printer are only about half the density

of the lithographic ones, holding just over 1K each. This means that you have to read quite a few strips to transfer a program of decent length between machines.

Printing strips yourself is done via another supplied program (itself read from a Softstrip). The process is again automated, with the user simply supplying the source filename. The software tells you how many strips will be required, and prints up to eight of them on each page. It prints fast, but your printer must produce a good quality graphics print if your strips are to be reliable. Any slippage in the paper feed can corrupt all eight strips and render them unusable. You'll also have to disable the line feed on your printer, as the Softstrip program needs to be able to advance by a non-standard amount.



The data I turned into Softstrips as part of the review proved very reliable. This included program listings and the text of the review itself. I also tried reading it through a clear plastic wallet, as found in those albums used by travelling sales reps, with no problems at all. The reader also had no trouble with a strip over which I scribbled a variety of felt tips. The only thing which did affect a successful reading was any misalignment of the reader, even by only a few degrees. The reader conducts its own error checking, but once the strip gets too wide of the reader's path, the head returns to the start of the strip and an error message is displayed.

The software supplied with the reader is varied, but not very stimulating, nor particularly well written. It's good spaghetti Basic in the old tradition, and is full of loan and mortgage checkers, mastermind games and graphics demos. They seem to have been chosen more for their length than lasting appeal. One of them, a mental arithmetic game, consistently refused to load.

As a product, the Softstrip reader worked well and reliably under almost all the tests to which it was put. This is really only part of the equation, though. As with any new form of data transmission, it has to prove sufficiently attractive to outweigh the cost and inconvenience of adding it to your repertoire. The main advantage of the Softstrip reader is that you can transmit data via paper in a machine-readable form. The package allows you to produce your own strips as well as

reading those printed by other people, so you have a complete Softstrip system.

At close on £200 though, it's not really a competitor for the floppy disc. If there was a stack of Stripware (as Softstrip software is called) printed on Softstrips which we could all take advantage of (albeit quite slowly), then the investment might be worthwhile. As it is, with a strip only holding 2K, and the reader taking nearly half a minute to read each one, you can't help concluding that it's an inefficient way of holding software.

To some extent it's a chicken and egg situation, as Stripware isn't likely to become freely available until there are a lot more Softstrip readers around, and few people are going to invest in readers unless there's a steady flow of programs.

← 12 WORLD BY NIGHT AND DAY

4150 DATA -79,-180,-75,-120,-75,-75,-68
-66.5,-57,-63.5,-67.5,-65,-76,-60,-78.5
-40,-66.5,55,-70,76,-65,106,-66.5,140,-
72,170,-78,165,-79,180,-9999,-9999

4160 DATA 78,-70,76,-68,75.5,-60,72,-55
62.5,-51,60,-44,66.5,-38,70,-22,80,-18,
82.5,-30,80,-65,78,-70,-9999,-9999

4170 DATA 55,-6,54.5,-3.5,53.5,-3.5,50
-8,51,1,52.5,2,56,-2.5,57.5,-2,58.5,-3,
58.5,-5.5,52,-6.5,51.5,-10,54,-10,57,-7.
5,55,-6,-9999,-9999

4180 DATA 63.5,-19,65,-14,66.5,-16,66,-
20,66.5,-23,64,-23,63.5,-19,-9999,-9999
4190 DATA 5,95,-7.5,105,-8.5,112,-9,125
-7.5,111,-7,105.5,0,104,5,95,-9999,-999
9

4200 DATA 0.5,109,7,117,5.5,119,-4,115,
-2.5,110,0.5,109,-9999,-9999

4210 DATA 18.5,121,7,126,7.5,122.5,11.5,
122,16,119.5,18.5,121,-9999,-9999

4220 DATA 32,130,34,130,38,139,45,142,4
4,145,42,140.5,35,140.5,32,130,-9999,-99
99

4230 DATA -1,131,-3,141,-10,150,-8,140,
-1,131,-9999,-9999

4240 DATA -35,173,-37.5,178.5,-41.5,175
5,-40.5,172.5,-46,166.5,-46.5,170,-38,1
75,-35,173,-9999,-9999

4250 DATA -12.5,49,-15.5,50.5,-25.5,47,
-22,43,-16,44.5,-12.5,49,-9999,-9999

4260 DATA 22.5,-84.5,23,-82.5,18,-68.5,
18,-74.5,20,-73.5,20,-77.5,22,-79,22.5,-
84.5,-9999,-9999

4270 DATA 20.5,-156,19,-156,19.5,-155,2
0.5,-156,-9999,-9999

4280 DATA 68.5,-180,66,-170,64.5,-174.5
65,-180,-9999,-9999

4290 DATA Jan,31,JANUARY,Feb,28,FEBRUAR
Y,Mar,31,MARCH,Apr,30,APRIL,May,31,MAY

4300 DATA Jun,30,JUNE,Jul,31,JULY,Aug,3
1,AUGUST,Sep,30,SEPTEMBER,Oct,31,OCTOBER

4310 DATA Nov,30,NOVEMBER,Dec,31,DECEMB
ER

4320 :
5000 DEF PROCfill(x,y):newy2=-100:inc2=
0

5010 PROCfill1(x,y,4)
5020 PROCfill1(x,y,4,-4)
5030 IF POINT(12,newy2)=0 THEN PROCfill
1(12,newy2,inc2) ELSE IF POINT(126,newy
2)=0 THEN PROCfill1(126,newy2,inc2)

5040 ENDPROC
5050 :
5060 DEF PROCfill1(x,y,inc)
5070 fillflag=FALSE

5080 newy=y:background=POINT(x,y)
5090 REPEAT

5100 PLOT77,x,newy
5110 IF fillflag THEN 5170

5120 X%=cp:Y%=cp DIV256
5130 A%=&0D:CALL&FFF1

5140 lx=(!cp AND65535)
5150 rx=(!cp+4)AND65535)

5160 IF lx=12 AND rx=1264 newy2=newy-inc
c:inc2=-inc:fillflag=TRUE

5170 newy=newy+inc
5180 UNTIL POINT(x,newy)<>background

5190 ENDPROC

Intelligent Auto-Save Utility

This utility by Alan Webster is ideal when saving a new version of your latest program. Not only does it prevent you from inadvertently overwriting the previous version, but the latest name and version number are automatically included in the program listing for reference.

WHY USE AUTO-SAVE?

This utility is designed to automate program saving. As the subtitle suggests, it warns of overwriting previously saved versions. But it does much more than this. The idea is developed from a trick used in the Master Editor. Essentially Auto-save allows you to keep the filename of a program anywhere in the first three lines of the program itself. It must be preceded by the two identifying characters ">", but may appear within either a REM statement or a PRINT statement. Thus:

```
10 REM My Program is Called
20 REM .>Prog1
```

would form a valid header. Once the machine code routine is installed, it adds the new Basic command "AS" (for Auto Save). If you type:

```
AS
```

when the above header is in the computer, you will be prompted with:

```
SAVE Prog1
```

And if that is the name you wish to save the program under, just press Return. If not, you may edit the name supplied, by deleting and retyping. When you are satisfied with the new name, which might perhaps be incremented to "Prog2" to indicate a new version, just press Return.

When Return is pressed, the disc will be checked to see if a file of the name entered already exists. If so the user is prompted with the Wordwise-like message:

```
Replace old file? (Y/N)
```

A negative response will return you to Basic, while a positive one will cause the filename to be written back to the Basic program listing before it is saved away.

As can be seen, the routine is very easy to use, and makes frequent saving of new program versions a positive delight.

It has the added advantage that the filename under which the program is saved is recorded in the program itself, so that if you have a printed copy of any given program, you automatically know what filename you have saved it under.

GETTING GOING

To get the program working, first type in the listing. It is reasonably clearly documented, though all remarks can be left out to speed up code entry. Once typed in, the program should be saved away before use. When it is run, the program will automatically save away the assembled machine code. The name used for this reflects the version of Basic on which it has been assembled. On Basic I it is called "Auto1", Basic II, "Auto2" and so on. As you will gather, the assembled code is machine specific, so if you wish to run the program on a different machine version, you should re-assemble it first.

Once the code has been automatically saved away, you need only enter:

```
*AUTO1 (or AUTO2 etc)
```

to load and initialise the code. Now type:

```
AS
```

and you will be prompted with the "Save" message, as described above.

Pressing Break at any time will disable the Auto-save feature, which may be reinstated either with:

```
CALL &900
```

or by repeating the *Auto call. But neither of these commands should be issued if the code is still active; neither should you re-run the assembler program when the code is active.

ABOUT THE PROGRAM

You may, if you wish, change the name of the Basic command provided by the routine. To do this, just alter the name in quotation marks in line 1450. You may also wish to change the area of RAM used by the routine. The code is currently located at &900, and extends up to &B81. On a model B, this means that it overwrites any function key definitions, though these can easily be reinstated after use. It also means that pressing the function keys or the Break key will print rubbish on the screen, though without any other adverse effects. To clear this, use Ctrl-Break, but this will also corrupt the Auto-save code. Note also that if the program which you are developing (on the B

or B+) defines any function key, you will need to reload in Auto1 (or Auto2) before calling the Auto-save. Such problems should not arise on a Master or Compact.

The program works equally well on the DFS and ADFS. ADFS users should note that the program uses the area above &380 for storage of the filename, allowing filenames (including pathnames) of up to 79 characters in length.

```

10 REM Program Basic Autosave
20 REM Version B1.10
30 REM Author Alan Webster
40 REM BEEBUG June 1987
50 REM Program Subject to Copyright
60 :
100MODE 7:W%=&380:ON ERROR GOTO 170
110PROCassem
120PRINT "Start address &;~vector
130PRINT "End address &;~P%
140PROCsave
150END
160:
170 ON ERROR OFF:MODE 7:IF ERR=17 END
180 REPORT:PRINT" at line ";ERL
190 END
200 :
1000 DEFPROCassem
1010 R%=(?&8015)-48
1020 IF R%<1 OR R%>6 OR R%=3 OR R%=5 PR
INT "Will not run on Basic ";R%":END
1030 dummy=EVAL("FNbas"+STR$R%)
1040 :
1050 FOR PASS=0 TO 3 STEP 3
1060 P%=&900:(OPT PASS
1070 :
1080 .vector
1090 LDA &202:LDX &203
1100 CMP #start MOD 256
1110 BNE notdone
1120 CPX #start DIV 256
1130 BEQ done
1140 .notdone
1150 STA brkv:STX brkv+1
1160 LDA #start MOD 256:STA &202
1170 LDA #start DIV 256:STA &203
1180 .done:RTS
1190 .start:PHA:TXA:PHA
1200 LDY #0:LDA (&FD),Y:CMP #4
1210 BEQ mistake
1220 .notours:PLA:TAY:PLA
1230 JMP (brkv)
1240 :
1250 .mistake
1260 LDY &A:DEY:TYA:CLC:ADC &B:STA &7E
1270 LDA &C:ADC #0:STA &7F:JSR nxtwrld
1280 BCS notours
1290 DEY:TYA:CLC:ADC &A:STA &A

```

```

1300 PLA:PLA:PLA:PLA:PLA:PLA
1310 JMP (&7E)
1320 :
1330 .nxtwrld:LDX #0:.nxt1:LDY #0
1340 .nxt2:LDA (&7E),Y:STA &72
1350 LDA table,X:CMP #&FF:BEQ not
1360 CMP #0:BEQ command:CMP &72
1370 BEQ retest:.nxt3:INX
1380 LDA table,X:BNE nxt3:INX:INX:INX
1390 JMP nxt1:.retest:INX:INX:JMP nxt2
1400 .not:SEC:RTS
1410 .command:INX:LDA table,X:STA &7E
1420 INX:LDA table,X:STA &7F:CLC:RTS
1430 :
1440 .table
1450 OPT FNEQUS("AS"):OPT FNEQUB(0)
1460 OPT FNEQUW(asave):OPT FNEQUB(255)
1470 .brkv
1480 OPT FNEQUW(!&202)
1490 :
1500 .asave
1510 JSR chkend
1520 \ Search for Basic program
1530 JSR chkprg
1540 \ Now search for filename
1550 LDA &18:STA &73:LDA #1:STA &72
1560 LDY #0:LDA (&72),Y:CMP #&FF
1570 BEQ noname:LDA #1:STA &74
1580 .lineloop:LDY #2:LDA&72:STA&76
1590 LDA &73:STA &77:LDA (&72),Y
1600 STA &75:SEC:SBC #3:TAX
1610 :
1620 .midloop
1630 CPX #0:DEX:BEQ endline:INX
1640 LDA (&72),Y:CMP #ASC(".")
1650 BNE midloop:INX:LDA (&72),Y
1660 CMP #ASC(">"):BNE midloop
1670 :
1680 \ Found '>'
1690 JMP getname:.endline:INC &74
1700 LDA &74:CMP #4:BEQ noname
1710 CLC:LDA &72:ADC &75:STA &72
1720 BCC lineloop:INC &73
1730 JMP lineloop
1740 :
1750 .noname:BRK
1760 OPT FNEQUB(127)
1770 OPT FNEQUS("No filename found")
1780 OPT FNEQUB(0)
1790 :
1800 .save:OPT FNEQUS("SAVE ")
1810 :
1820 .getname
1830 LDX #255:.sav1:INX:LDA save,X
1840 JSR &FEE:CMP #32:BNE sav1
1850 LDX #255:.getname2
1860 INY:INX:LDA (&72),Y:STA W%,X
1870 CMP #13:BEQ endname:CMP #32
1880 BNE getname2:LDA #13:STA W%,X
1890 .endname:LDY #255:.insert:INX

```



```

1900 LDA W%,Y:STA &75:CMP #13:BEQ endin
1910 TYA:PHA:LDA #138:LDX #0
1920 LDY &75:JSR &FFF4
1930 PLA:TAY:JMP insert
1940 .endin:LDX #work MOD 256
1950 LDY #work DIV 256:LDA #0
1960 JSR &FFF1:BCC noes:JMP escape
1970 .noes \ Got name, now save
1980 JSR testfile:CMP #0:BEQ saveit
1990 JSR message:BEQ saveit
2000 JMP cont \Don't save
2010 :
2020 .saveit
2030 JSR updateline
2040 LDA &18:STA pblock+3:STA pblock+&B
2050 LDA #0:STA pblock+2:STA pblock+&A
2060 LDA &FF:STA pblock+&C
2070 STA pblock+&D:LDA &12:STApblock+&E
2080 LDA &13:STA pblock+&F
2090 LDA &FF:STA pblock+&10
2100 STA pblock+&11:LDX #pblock MOD 256
2110 LDY #pblock DIV 256:LDA #0
2120 JSR &FFDD:JMP cont
2130 :
2140 .testfile
2150 LDX #pb2 MOD 256:LDY #pb2 DIV 256
2160 LDA #5:JSR &FFDD:CLC:RTS
2170 .pb2:OPT FNEQUW(W%)
2180 OPT FNEQUW(0):OPT FNEQUW(0)
2190 OPT FNEQUW(0):OPT FNEQUW(0)
2200 OPT FNEQUW(0):OPT FNEQUW(0)
2210 OPT FNEQUW(0):OPT FNEQUW(0)
2220 :
2230 .message
2240 LDY #255:.loop:INY:LDA mess,Y
2250 JSR &FFEE:CMP #63:BNE loop
2260 .quest:LDA #7:JSR &FFEE:JSR &FFE0
2270 BIT &FF:BMI escape:CLC:AND #223
2280 CMP #78:BEQ nope:CMP #89:BNE quest
2290 JSR &FFEE:JSR &FFE7:LDA #0:RTS
2300 .nope:JSR &FFEE:JSR &FFE7:LDA #1
2310 RTS
2320 :
2330 .mess:OPT FNEQUS("Replace old file
(Y/N) ?")
2340 .escape:LDA #126:JSR &FFF4
2350 JSR &FFE7:JMP cont
2360 :
2370 JMP cont
2380 .work:OPT FNEQUW(W%)
2390 OPT FNEQUB(&80)
2400 OPT FNEQUB(32)
2410 OPT FNEQUB(127)
2420 .pblock
2430 OPT FNEQUW(W%)
2440 OPT FNEQUW(0):OPT FNEQUW(-1)
2450 OPT FNEQUW(exec):OPT FNEQUW(-1)
2460 OPT FNEQUW(0):OPT FNEQUW(-1)
2470 OPT FNEQUW(0):OPT FNEQUW(-1)
2480 :

```

[Thanks to Roger Cullis for help with this program.]

```

2490 .updateline
2500 LDY #0:LDA (&76),Y:STA &2B:INY
2510 LDA (&76),Y:STA &2A:INY
2520 LDA (&76),Y:SEC:SBC #3:TAX
2530 .inlp:INY:LDA (&76),Y
2540 STA &700,Y:CMP #ASC("."):BNE inlp
2550 INY:LDA (&76),Y:STA &700,Y
2560 CMP #ASC(">"):BNE inlp:TYA:TAX
2570 .omitl:INY:LDA (&76),Y:CMP #32
2580 BEQ enfil:CMP #13:BNE omitl
2590 .enfil \ New name!
2600 STY &74:LDY #255:.newlp:INX:INY
2610 LDA W%,Y:STA &700,X:CMP #13
2620 BNE newlp:LDY &74
2630 .restofline:LDA (&76),Y:STA &700,X
2640 INX:INY:CMP #13:BNE restofline
2650 LDY #3:JSR linein:RTS
2660 :]
2670 NEXT
2680 ENDPROC
2690 :
2700 DEFFNbas1
2710 cont = &8B0C:chkend= &9810
2720 chkprg= &BE88:exec = &801F
2730 linein= &BCAA:=0
2740 :
2750 DEFFNbas2
2760 cont = &8B9B:chkend= &9857
2770 chkprg= &BE6F:exec = &8023
2780 linein= &BC8D:=0
2790 :
2800 DEFFNbas4
2810 cont = &9005:chkend= &9BA6
2820 chkprg= &BDE5:exec = &8043
2830 linein= &BB15:=0
2840 :
2850 DEFFNbas6
2860 cont = &8FE6:chkend= &9B86
2870 chkprg= &BD45:exec = &8043
2880 linein= &BA65:=0
2890 :
2900 DEF FNEQUB(B%)
2910 ?P%=B%:P%=P%+1
2920 =PASS
2930 :
2940 DEF FNEQUS(SS)
2950 $P%=$S:P%=P%+LEN(SS)
2960 =PASS
2970 :
2980 DEF FNEQUW(W%)
2990 !P%=W%:P%=P%+2
3000 =PASS
3010 :
3020 DEFPROCsave
3030 AS="SAVE Auto"+STR$R$+" "+STR$~vec
tor+" "+STR$~P%
3040 $W%=AS:X%=W%:Y%=W% DIV 256
3050 CALL &FFF7
3060 ENDPROC

```


All options perform a number of checks at various points. Any potential error is flagged by an audible 'bleep', and either a message displayed if there is an obvious choice, or a return made to the menu.

This program has proved very useful in the author's work, and it has been made as robust as possible so that it can be used by people with no knowledge of computers. However, pressing Break will cause a loss of data if this has not yet been saved.

PROGRAM NOTES

Before using the program, check that the value of Q% (line 150) is higher than the total number of marks or maximum mark possible (whichever is the greater). This value dimensions the arrays in line 150. The procedure PROCName at line 2280 sets a limit of 7 characters on the maximum length of a file name but you may readily change this if another limit is preferred (for use with the ADFS, for example).

```

10 REM Program STATS
20 REM Version B1.12
30 REM Author David Andrews
40 REM Beebug June 1987
50 REM Program subject to copyright
60 :
MODE 7:K%=0:ON ERROR GOTO 2490
110 REM Q% must be set to equal
120 REM total number of marks or
130 REM highest mark possible,
140 REM whichever is the greater.
150 Q%=250:DIM C%(Q%),D%(Q%):L$=STRING
$(40,"="):name$=""
160 :
170 REPEAT
180 @%=10:VDU3,15,26:CLS:*FX3,0
190 PRINTTAB(0,1)LISTAB(0,6)L$
200 FOR I%=3TO4:PRINTTAB(7,I%)CHR$141"
MARKS ANALYSIS PROGRAM":NEXT I%
210 PRINTTAB(12,5)"by David Andrews"
220 VDU28,0,24,39,7
230 PRINTTAB(7,1)"1 Enter data from ke
yboard"
240 PRINTTAB(7,2)"2 Save data to disc"
250 PRINTTAB(7,3)"3 Read data from dis
c"
260 PRINTTAB(7,4)"4 View and edit data
"
270 PRINTTAB(7,5)"5 Analyse data"
280 PRINTTAB(7,6)"6 End"
290 PRINTTAB(9,10)"Which choice ? ":V
=GET-48:CLS
300 IF V=1 THEN PROCcenter

```

```

310 IF V=2 THEN PROCtodisc
320 IF V=3 THEN PROCfromdisc
330 IF V=4 THEN PROCedit
340 IF V=5 THEN PROCshow
350 UNTIL V=6
360 CLS:PRINTTAB(12,5)"Program finishe
d"TAB(4,8)"Remove disc before switching
off":VDU26,31,0,24:END
370 :
1000 DEFPROCcenter
1010 IF K% PRINT"Overwrite existing dat
a (Y/N)":IF NOT FNok ENDPROC
1020 CLS:INPUT"Description: "title$
1030 REPEAT
1040 INPUT"Maximum mark possible: "max$
:PROCcheck(max$,Q%):IF NOT OK THEN VDU7:
PROCblank
1050 UNTIL OK
1060 M%=VAL(max$)
1070 PRINT"Input marks, Escape to finis
h"
1080 VDU28,0,24,39,13:@%=3
1090 REPEAT:REPEAT
1100 PRINT SPC(3-LEN(STR$(K%+1)))STR$(K
%+1)+": ";INPUT"" mark$
1110 PROCcheck(mark$,M%)
1120 IF NOT OK THEN VDU7:PROCblank
1130 UNTIL OK
1140 K%=K%+1:C%(K%)=VAL(mark$)
1150 *FX15
1160 UNTIL FALSE
1170 ENDPROC
1180 :
1190 DEFPROCedit
1200 @%=3:IF K%=0 VDU7:ENDPROC
1210 REPEAT
1220 VDU14,26:CLS
1230 PRINT"Description : "title$'
1240 PRINT"Use SHIFT to scroll if neces
sary""Escape to return to menu"
1250 VDU28,0,24,39,6
1260 FOR X%=1 TO K%:PRINT CHR$(134)X% C
HR$(135) C%(X%):NEXT
1270 PRINT':*FX15
1280 REPEAT
1290 PRINT"Index number ":INPUT"to alt
er: " index$
1300 PROCcheck(index$,K%+1)
1310 IF NOT OK THEN VDU7:PROCblank
1320 UNTIL OK
1330 VDU15:PROCblank
1340 REPEAT
1350 PRINT"Index number"CHR$(134)index$
CHR$(135):INPUT".. new mark ? "newmark
$:PROCcheck(newmark$,M%):IF NOT OK THEN
VDU7:PROCblank
1360 UNTIL OK
1370 I%=VAL(index$):N%=VAL(newmark$):IF
I%<K% THEN C%(I%)=N% ELSE K%=K%+1:C%(K
%)=N%

```



```

1380 UNTIL FALSE
1390 ENDPROC
1400 :
1410 DEFPROCtodisc
1420 IF K%=0 VDU7:ENDPROC
1430 VDU14:REPEAT:CLS:*CAT
1440 INPUT"Name of file to be created:
"name$:IF name$="" THEN ENDPROC
1450 PROCname(1)
1460 UNTIL OK
1470 A=OPENOUT(name$)
1480 PRINT#A,title$,M%
1490 FOR X%=1 TO K%
1500 PRINT#A,C%(X%)
1510 NEXT:CLOSE#A:VDU15
1520 ENDPROC
1530 :
1540 DEFPROCblank
1550 VDU11:PRINT SPC(39):VDU11
1560 ENDPROC
1570 :
1580 DEFPROCwait
1590 PRINT"Press any key to continue";:
IF GET ENDPROC
1600 :
1610 DEFPROCcheck(m$,limit%)
1620 OK=TRUE
1630 IF m$="" THEN OK=FALSE:ENDPROC
1640 FOR E%=1 TO LEN(m$)
1650 IF ASC(MID$(m$,E%))<48 OR ASC(MID$(
(m$,E%))>57 THEN OK=FALSE
1660 NEXT
1670 IF VAL(m$)>limit% THEN OK=FALSE:EN
DPROC
1680 ENDPROC
1690 :
1700 DEFPROCfromdisc
1710 IF K% PRINT"Overwrite existing dat
a (Y/N)";:IF NOT FNok ENDPROC
1720 VDU14:REPEAT:CLS:*CAT
1730 INPUT"File from which data is wa
nted: "name$
1740 IF name$="" THEN ENDPROC
1750 PROCname(2)
1760 UNTIL OK
1770 A=OPENIN(name$)
1780 INPUT#A,title$,M%
1790 K%=0:REPEAT
1800 K%=K%+1:INPUT#A,C%(K%)
1810 UNTIL EOF#A:CLOSE#A:VDU15
1820 ENDPROC
1830 :
1840 DEFPROCshow
1850 IF K%=0 VDU7:ENDPROC
1860 PRINT"Please wait . . ."
1870 FOR X%=0 TO Q%D%(X%)=0:NEXT
1880 L%=0:N%=0:T%=0:D%=0:max=0:min=Q%
1890 FOR X%=1 TO K%
1900 mark=C%(X%)
1910 IF mark>max THEN max=mark
1920 IF mark<min THEN min=mark
1930 D%(mark)=D%(mark)+1
1940 IF D%(mark)>L% THEN L%=D%(mark)
1950 N%=N%+1:REM number of items
1960 T%=T%+mark:REM running total
1970 D%=D%+(mark*mark):REM total square
s
1980 NEXT X%
1990 CLS:PRINTTAB(5,12)"Printed output
wanted (Y/N) ? ";:IF FNok THEN printer=T
RUE:VDU26,12,2 ELSE printer=FALSE:VDU26,
12,14
2000 PRINT"Description : "title$"File
: "name$
2010 mean=T%/N%:@%=4
2020 PRINT"Total no. of items"TAB(22)N
%
2030 PRINT"Maximum possible mark"TAB(22
)M%
2040 PRINT"Spread of marks"TAB(22)max"
to ";min
2050 @%=&20207
2060 PRINT"Mean"TAB(22)mean
2070 dev=SQR((D%-N%*mean*mean)/N%)
2080 PRINT"Standard deviation"TAB(22)de
v'
2090 *FX3,4
2100 PRINT"Finish analysis (Y/N) ? ";:I
F FNok THEN VDU3,15:ENDPROC
2110 VDU11:PRINT"Standardise marks (Y/
N) ? ";
2120 IF FNok THEN standard=TRUE:PROCbla
nk:PRINT"New mean ?";:INPUTTAB(23)newmea
n:PRINT"New deviation ?";:INPUTTAB(23)ne
wdev ELSE standard=FALSE:newmean=mean:ne
wdev=dev:PROCblank
2130 *FX3,0
2140 IF standard PRINT"STANDARDISED MA
RKS in column E:""New Mean"TAB(22)newme
an"New deviation"TAB(22)newdev
2150 pos=1:old=0:Z%=0
2160 PRINT"A = Mark"SPC(11)"C = Freque
ncy""B = Position"TAB(19)"D = Cumulativ
e %"
2170 PRINTTAB(1)"A"TAB(6)"B"TAB(10)"C"TA
B(14)"D"TAB(18)"E"STRINGS(39,"_")
2180 @%=4
2190 FOR X%=max TO min STEP-1
2200 F%=FNstandard
2210 IF (F%>M% OR F%<0) THEN bar$=" OUT
OF RANGE" ELSE bar$=" "+STRINGS(D%(X%)*1
4 DIV L%,"*")
2220 PRINTSPC(3-LEN(STR$(X%)))STR$(X%);
2230 IF D%(X%)<0 PROCOrder:PRINTpos,D%(
X%),FNpercent,F%,TAB(23)bar$ ELSE PRINT
2240 NEXT
2250 VDU1,12,3,15:*FX21,0
2260 PROCwait
2270 ENDPROC

```


HARD DISCS FOR ALL?

David Graham checks out Technomatic's £399 hard disc system.

Product : Hard Disc System
Supplier : Technomatic Ltd
 17 Burnley Road,
 London NW10 1ED.
 Tel. 01-208 1177

Price : 10 Mbyte £399 plus VAT
20 Mbyte £584 plus VAT

Details of special offer for BEEBUG members at end of review.

Surely it has to be a bargain, 10 megabytes at £399 + VAT? Clearly Technomatic's hard disc drives represent something of a price breakthrough for the Beeb market. At their current price, the 10 meg drives cost not a massive amount more than dual floppy drives, while offering 7 or 8 times the capacity, and much faster access times. No longer, it seems, do you need to have a multi-station Econet system to justify the price of a hard disc. But what are the snags, and how easy are these drives to use?

INSTALLATION

The drives measure just 6in by 3.5 by 12.5in deep, and are considerably smaller than my half-height twin floppy drives. The metal cabinet matches the Beeb in colour and texture, and has a black front panel with air vent and indicator lights. At the rear, there is a socket for the mains lead, a power switch, and a 1 metre ribbon cable which connects to the Beeb's 1 MHz bus. There is also a socket at the rear to extend the 1 MHz bus, either to connect a further Winchester, or for any other 1 MHz equipment.

Somewhat surprisingly, this Winchester carries no installation overhead whatsoever: there are no ROMs to plug in and

no carrier boards to insert. Master users can simply plug the drives into the 1 MHz bus socket, the mains lead (already wired) into a suitable 13 amp socket, and switch on. The reason for this is that all the hard disc code is already present in the ADFS ROM inside the computer. In fact, the ADFS started life as a Winchester filing system, and the floppy disc code was only added afterwards. As you will see in a moment, this accounts for the very poor timings achieved by the ADFS for certain tasks. Incidentally the floppy code in the ADFS ROM has been rewritten by Acorn for the Compact, and this is now said to be some 25% faster. One day Master users may be offered this faster ADFS version as an upgrade.

But back to the Winchester itself. What of model B users? They need to upgrade to the ADFS before they can use a hard disc drive. As you may know, this involves changing the disc controller chip to a 1770 (already present on the B+), and then plugging in an ADFS ROM. When you do this, you will unfortunately find that PAGE has been bumped up to &1F00 (unless you have unplugged the DFS), leaving you with even less space for programs and data. Of course, you can get around this by adding a shadow RAM card or a second processor, but this makes the Winchester upgrade considerably more expensive.

SWITCHING ON

When a hard disc is switched on it tends to make a sound which one reviewer has likened to a jet engine powering up. It takes some 10 to 15 seconds to reach full speed, and then continues to spin at some 3000 revs until the power is turned off. Both the motor itself, and its cooling fan generate noise, which users familiar with IBM PCs will recognise. On the Technomatic unit, this is by no means excessive, though it is certainly not the quietest drive that I have ever heard.

Once powered up, the Winchester must be engaged, either by pressing Break, or by using the Master's *CONFIGURE command.

*CONFIG. HARD will cause the Master to power-up with the hard disc (as drive zero) as the currently selected drive. This command is cancelled with:

*CONFIG. FLOPPY which causes the Master to select the floppy disc (re-allocated to drive 4) as

the currently selected drive at the time of power-up.

This renumbering of drives will be unfamiliar to DFS users. When using the ADFS with floppies, the drives are labelled 0 and 1. Drives 2 and 3, familiar to DFS users, disappear when using the ADFS because the ADFS treats both sides of a disc as if it were a continuous surface. When the hard disc is engaged, it takes over the "drive 0" slot, and floppy drives 0 and 1 become drives 4 and 5.

WINCHESTER TIMINGS

| | A D F S | | |
|--------|---------|--------|-------|
| | DFS | Floppy | Hard |
| Test 1 | 36.95 | 40.53 | 16.82 |
| Test 2 | 35.62 | 25.55 | 2.86 |
| Test 3 | N/A | 29.08 | 2.36 |

Test 1: Write 20-byte string, 1000 times
Test 2: *LOAD 20K file 10 times
Test 3: Change directories (2 deep) 20 times

All timings are in seconds. Note that timings will vary from one test to another, depending upon the position of files on the disc surface. The settings of the disc drive timing links will also affect floppy disc timings. For the above tests, default settings were used.

Table 1

Once the hard disc has been selected, the user will be rewarded by the following response to *FREE:

10,400,000 Bytes free

1,000 Bytes used

Technomatic's Winchesters come ready formatted, and contain just a single file, which holds data on any bad areas of the disc, allowing them to be avoided by the disc management system. No formatter or front end software is supplied. It is unlikely that you will want to reformat your disc, but the lack of utilities is a little disappointing, though this gap can be bridged by some of the utilities which have been published in BEEBUG for Master users, or by ACP's Disc Toolkit or BEEBUG's new Master ROM (see later).

WINCHESTER TIMINGS

At 3000 revs, the Winchester rotates about ten times faster than conventional floppy drives, and given that it also does not need to gather speed each time that a disc access is called for, we might expect it to perform better than a floppy drive under similar circumstances. This is certainly the case, as the various timings suggest (see Table 1). The screen loading test illustrates this well. It took the machine just 2.86 seconds to load ten 20K mode 2 screens, while the ADFS with floppies took 25.55 seconds, or nearly 10 times longer. Similar contrasts are obtained for loading and saving Basic programs.

The directory-changing test is also quite revealing. This time the Winchester performed some 12 times better than the floppies, with the latter taking an appalling 29 seconds to change directory just 20 times. In speed terms the Winchester in general, and Technomatic's Winchester in particular, is a joy to use; especially if you have become used to the drudging speed of the ADFS with floppies. This massive speed opens up one or two interesting possibilities. First of all, it makes it quite feasible to use the hard disc as virtual memory, saving and loading data or program segments continuously as a program runs, instead of trying to use the computer's very limited user RAM, whether this be on the model B or Master.

Secondly, it opens up the possibility of real-time animation. By simply *LOADING screens one after another, you can get an animation rate of some 10 frames a second in mode 4, or about half that in modes 1 or 2. Intrigued by this possibility, I tested out the principle using the Spitfire program from BEEBUG Vol.3 No.1. This draws a Spitfire aeroplane at any size and angle in mode 4, and I just added a line to save the screen under an incrementing file name at each press of the space bar. A three-line program was then used to *LOAD in sequence the 30 or so screens thus created. The effect was extremely good, though it would be considerably improved by using solid rather than wire-frame images - which would of course take no longer to load in (though quite a bit longer to draw in the first place).

TREES, ROOTS AND NESTING.

DFS users will probably have heard extolled the great advantages of the tree directory structure of the ADFS. It has a root directory (\$) which can contain up to 47 objects. These may either be files or further directories. Each sub-directory may contain a further 47 objects, and so-on, almost ad infinitum.

There are indeed many advantages to such an arrangement, but there are also inherent problems. The deeper you nest, the more work you have in specifying any given file. For example, it is tedious, and error-prone to execute commands like the following:

```
*COPY :0.$MAY-WORK.PROGS.UTILS.filena  
me :1.$MAY-BACKUP.PROGS
```

when all you want to do is to copy a file from one disc to another. Imagine copying a group of such files, where wildcards were inappropriate. And the same kind of problem arises when you just want to move around the disc from directory to directory - an extremely common activity if you are making good use of the ADFS directory structure.

Using floppies with the ADFS, this is all really rather a pain. You can reduce the pain somewhat by making sure that you do not nest too deeply. But in so doing you reduce the advantages of using the ADFS, and in any case, it does not solve the problem. Now we get to the point: using a Winchester, you are forced to nest reasonably deeply if you are going to make use of any more than a small fraction of the hard disc's potential. Some kind of ADFS front end then becomes absolutely essential.

About the best that I have so far come across is Advanced Computer Products' ADT ROM. This offers an automatic ADFS disc menu, which facilitates the rapid moving around between directories. It works well, and at lightening speed with a hard disc system, and also offers a number of other useful commands such as *CATALL which will catalogue the whole disc, or any part of it; and *DIRALL which will list all directories below the currently selected directory. There are still many gaps, however. ADT offers no easy means to copy or delete files at the press of a key;

though it is to be hoped that many of these gaps will be filled by BEEBUG's new Master ROM.

BACKING UP

If taking back-ups is important when using 600K floppies, it is absolutely essential when using a 10 or 20 megabyte hard disc. Though usually extremely reliable, hard disc systems do crash from time to time, and back-ups should therefore be taken regularly. The traditional way to back up a Winchester is to use a tape streamer, which will store the whole contents of the disc onto magnetic tape in a special format (working considerably faster than 1200 baud!).

But tape streamers are generally more expensive than the £399 paid for the hard disc itself, and it is considerably cheaper to use floppies. But if you are trying to back up a whole Winchester onto floppies, it will take maybe 20 discs, and a good deal of patience. Probably the best approach is to make back-ups to floppy disc on a file-by-file basis as and when files are created. This carries very little time overhead, and costs virtually nothing.

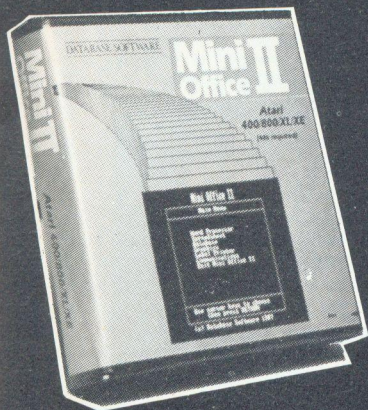
In conclusion, I must fully recommend the Technomatic drives. They are exceedingly keenly priced, and represent excellent value for money. The massive speed and capacity offered by these drives really puts the BBC Micro, and more particularly the Master 128 into the serious league, while at the same time remaining within the reach of many home users.

B

MEMBERS' SPECIAL OFFER

Technomatic will shortly be increasing the price of their 10Mbyte drive. If they do so before 30th June, they have agreed to hold their price to BEEBUG members for orders placed by that date. The 20Mbyte drive (£584 plus VAT) is available to BEEBUG members for £554 plus VAT if purchased by the same date. The Acorn ADFS ROM is also available if ordered with the 20Mbyte drive at the special price of £15 plus VAT. Please quote your membership number with your order.

Mini Office II



Mini Office II is a comprehensive integrated suite of programs setting a new standard in home and business software.

Most of the wide range of features – many of which are usually only available on software costing hundreds of pounds – are easily accessed by using cursor keys to move up and down a list of options and pressing Return to select.

Mini Office II takes over where the original, highly successful Mini Office left off, with numerous extra features, two additional modules, a program to convert existing Mini Office files to Mini Office II format, and a 60 page, easy to follow manual.

And it's at a price everyone can afford!

| | |
|----------------|-------------------|
| B, B+ | Tape £14.95 |
| Master | 5.25" disc £16.95 |
| Master Compact | 5.25" disc £19.95 |
| | 3.5" disc £21.95 |

| | |
|----------------|--------------|
| B, B+, Master, | 4 by 32k Rom |
| Master Compact | board £59.95 |

6 powerful home and business programs in just one package – at a price that simply can't be matched!

- **WORD PROCESSOR:** Compose a letter, set the print-out options using embedded commands or menus, use the mail merge facility to produce personalised circulars – and more!
- **DATABASE:** Build up a versatile card index, use the flexible print out routine, do powerful multi-field sorting, perform all arithmetic functions, link with the word processor – and more!
- **LABEL PRINTER:** Design the layout of a label with the easy-to-use editor, select label size and sheet format, read in database files, print out in any quantity – and more!
- **SPREADSHEET:** Prepare budgets or tables, total columns or rows with ease, copy formulae absolutely or relatively, view in either 40 or 80 column modes, recalculate automatically – and more!
- **GRAPHICS:** Enter data directly or load data from the spreadsheet, produce pie charts, display bar charts side by side or stacked, overlay line graphs – and more!
- **COMMS MODULE:** Using a modem you can access services such as MicroLink and book rail or theatre tickets, send electronic mail, telex and telemessages in a flash – and more!

INSTANT Mini Office II

Now available on a power-packed rom board!

- Four 32k roms on one board packed with 128k of super fast machine code.
- Easy to install – no soldering needed.
- Split second application selection.
- Frees valuable ram space for much more data.
- Fully compatible with the whole BBC Micro range, Aries or Watford shadow ram board, and tape, DFS, ADFS and ECONET filing systems.

The full selection of Mini Office II is available through BEEBUG Retail at MEMBER'S DISCOUNT. Please refer to your BEEBUG Retail Catalogue

DATABASE SOFTWARE

Europa House, 68 Chester Road,
Hazel Grove, Stockport SK7 5NY

BEEBUG

MASTER ROM

Filing
DFS

Library

Butes Free
181136

MARK FILE

AD2

AD3

AD4

AD5

AD6

AD7

AD8

AD9

AD10

AD11

AD12

AD13

AD14

AD15

AD16

AD17

AD18

AD19

AD20

AD21

AD22

AD23

AD24

AD25

AD26

AD27

AD28

AD29

AD30

AD31

AD32

AD33

AD34

AD35

AD36

AD37

AD38

AD39

AD40

AD41

AD42

AD43

AD44

AD45

B COMMAND1

D COMMAND4

F ROMIT

H DISCMST

J TELTEXT

L QCALC

N STUDIO

P SPELLCK

Q TOOLKIT

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

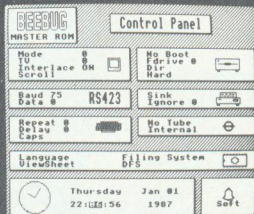
Q ADFSMF2

Q ADFSMF2

Q ADFSMF2

DIARY & ALARM

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



CONTROL PANEL

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

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

DISC MENU

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

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

SIDEWAYS RAM COMMANDS

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

BEEBUGSOFT

Master ROM

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

DISC COMMANDS

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

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

Price**£39.00****MEMBERS PRICE****£29.25**

Supplied on ROM with spiral bound manual.

NEW

BEEBUGSOFT

Command

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

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

COMMAND has to be one of the most comprehensive and flexible pieces of modern software around...
It works impressively well, without any problems...

A & B COMPUTING MAY 87

...has an amazing 50 commands in its repertoire.
...it's one of the best Beeb terminal packages to have been released in a couple of years, and full of potential.

MICRONET

It is very flexible and powerful...
Despite its incredible power it is easy and simple to use.

EDUCATIONAL COMPUTING MARCH 87

VIEWDATA TERMINAL

A full feature Prestel Terminal offering:

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

SCROLLING TEXT TERMINAL

Ideal for accessing Telecom Gold and Bulletin Boards.

Features include:

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

SPECIAL FEATURES

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

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

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

COMMAND

Copyright (C) Beebug Limited 1986

| | | |
|-----------------------------------|------------------|------------------|
| Record Name | PRESTEL | |
| Number | 01-618-1111 | |
| Sign-on Terminal | 4444444444444444 | |
| Filter Rate | Transmit Off 75 | Receive Off 1200 |
| Standard 2 (7+1 bits even parity) | | |
| Echo | Off 0 | Monitor Off 0 |
| Xon/Xoff | Off 200 220 | Band 0 |
| Colour | 7 0 | Mode 7 |

Price

£39.00

MEMBERS PRICE

£29.25

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

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

Command Telephone Directory

COMMAND INSTRUCTION SET

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

Personal Ads

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 30p 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.

UPGRADING to 80-track, therefore have two Cumana CS100 drives (40T SS with PSU) at £60 each or £115 the pair. Can be seen running. Tel: North Weald, Essex (037882) 2622 evenings.

WORDWISE Plus - unused, includes ROM, manual, reg card, key strip & UK p&p £25. Tel: Scunthorpe (0724) 867220 evenings.

BBC B, ATPL Sideways ROM/RAM board, Viglen disc drive, Taxan printer, Taxan colour monitor, Teletext receiver, Ferguson tape recorder, Plinth, plus dozens of disc and tape programs. A very comprehensive system £1200 o.n.o., phone or write to: J. Phillips, Morwetha, Lighthouse Hill, Portreath, Cornwall. TR16 4LH. (0209) 842737.

CUMANA single 100k disc drive with PSU £50; Addison Wesley Bar Code Teaching Pack (disc) £45; ATPL ROM board £20; BEEBUGSOFT Toolkit ROM £10; Wordwise Plus £25; Disc Doctor £15; RH Electronics Family Finance cassette £3; The Hobbit cassette £5. All manuals included. Tel: Godstone, Surrey (0883) 842691 evenings.

ROMEX ROM expansion board (buffered) £20; CC Disc Doctor £15; CC Graphics ROM £15; BEEBUG Wordease ROMs £12, Toolkit £10; BEEBUG Sprite Generator £6, BEEBUG Spellcheck £5, all on 80T disc and boxed with manual. Tel: Wells, Somerset (0749) 78705 any time.

WORDWISE PLUS ROM inc tape and manuals, original p&p included at £30. Tel: Doncaster (0302) 781528.

BOOKS: Various BBC B subjects. Tuition, Disc (DFS), ROMs, Graphics, etc. Tel: Southampton (0703) 845104.

20 Original Games for the BBC B, including Frak! and Repton 2. All in excellent condition. From £1.99 each or £35 the lot. Tel: Neath (0639) 57219 evenings.

MANNESMANN Tally MT80 printer "Enterprise" badged, Epson compatible with manual and BBC cable, as new £105. Tel: Wokingham (0734) 784897.

ARIES B32 shadow RAM board £60; Aries B12 ROM board £27; Wordwise Plus £25, Intersheet £25; Interchart £15, Floppy-wise Plus £15; Gemini DataGem £35; Acornsoft Logo £35; Elite, Revs, Aviator, Citadel, Repton 2, Doctor Who, Mini Office, Adrian's Diary £7 each; all games on disc with original manuals. Tel: 01-589 6122 evenings.

MASTER 128 still under warranty including Dumpmaster and Advanced Toolkit £360; Akhter MD400A disc drive £75 including miscellaneous discs also under warranty. Master + Disc £400 for quick sale. Write or call: G. Dutton, 40 Grafton Road, Harwich, Essex CO12 3BD.

BUILD A SYSTEM TO SUIT YOUR INDIVIDUAL NEEDS

START WITH SYSTEM DELTA AND CARD INDEX, THEN EXPAND AS YOUR NEEDS DICTATE

You can either add programs into the system yourself or choose from the existing range of over 12 other modules (with many more coming soon).

The CARD INDEX, on disc, is the simplest yet most powerful and flexible database available today. It's fast, it's efficient and it outdates all other databases with its revolutionary features. It can find any single record in a file over over 8000 records in less than 3 secs! It's the ONLY database which allows you to transfer your data in any way required into a vast range of other software including wordprocessors, spreadsheets, graphics and specialist advanced products like a report writer or mailing program. Powerful functions can be achieved without any programming knowledge! However, it is programmable so may be customised if required.

The SYSTEM DELTA ROM gives you 150 extra commands instantly at your disposal to rapidly write RELATIONAL programs with up to FIVE files open at a time. Imagine, writing a graph program in only 13 lines and less than 10 minutes! or a fully relational Costings program in less than 3 hours!

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



Cheque/Access No. Signature

NAME

ADDRESS

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

Events

Acorn User Show
Barbican Centre, London
23-26th July 1987

PCW Show
Olympia, London
3-8th September 1987

Electron and BBC Micro User Show
Old Horticultural Hall, Westminster
13-15th November 1987

The Fourth High Technology in
Education Exhibition
Barbican Centre, London
20-23rd January 1987

Note: We will, as usual, have a stand at the Acorn User Show in July, and we look forward to meeting many BEEBUG members there.

Advertising in Beebug

For advertising details, please contact

Yolanda Turvelo
on
(0727) 40303

or write to:

Dolphin Place
Holywell Hill
St. Albans AL1 1EX

VIEWSPELL package for BBC complete, ROM, disc, user disc, reference card, and installation instructions in original packaging £20. Tel: Oakham (0572) 812915.

BBC B complete with OS1.2, Basic II, Solidisc DDFS 2.2, and 1770 disc interface upgrade, Wordwise fitted £200 o.n.o. Tel: Stirling (0786) 85633.

SOFTWARE all boxed as new - Acornsoft tapes: Rocket Raid, Philosopher's Quest, Castle of Riddles, Algebraic Manipulation. Acornsoft Elite disc. BBCsoft Taxcalc (82-83) tape. Bug-Byte Golf. Program Power tapes: Munchyman & Chess. Level 9 Colossal Adventure tape. £16 the lot (inc postage). Write to: A. S. Clarke, 74a New Road, Clanfield, Hampshire PO8 0NN.

TORCH Z80 2nd Processor. Includes CP/N, Perfect software & manuals. Cased with own power supply, only £175. Tel: 01-857 7369 evenings.

WS2000 Modem with BBC B interface cable and manual. Excellent condition with original packaging, supplied with FREE registration to Microlink and Telecom Gold. Only £99. Tel: Paignton, Devon (0803) 526058.

BBC B with series 3 board, 1.20S, Basic II, Viglen ZIF socket + 3 cartridges, Watford DFS 1.30, Dual Cumana SS 80T drives with integral PSU (one 40/80 switchable), WW Plus, AMX Mouse, AMX Utils & Super Art, Graphics Dump ROM, Joysticks, much software inc Music System, BEEBUG Design, Hershey, Paintbox, and many classic games, over 20 SS discs and many cassettes - £350 the lot. Tel: Oxford (0865) 61249.

INTERWORD complete, an unwanted gift - £25; Watford single 40/80 switchable disc drive 4 months old (as new) - £90 o.n.o. Tel: Stevenage 369388 after 6pm.

ACORN Bitstik £165, boxed, as new. Global Products P8000 industrial EPROM programmer £275. Tel: Derby (0332) 362798.

VIEWSPELL for Master Compact complete with manuals on 3.5" ADFS disc, only 3 months old, half price £20. Tel: 01-472 7122 evenings & weekends, 01-554 3815 days.

ACORN/BBC Z80 Second Processor including all CP/M packaged software, books plus Wordstar and dBase II £199 o.n.o. Tel: Bedford (0234) 67067 evenings.

NIGHTINGALE modem with Commstar ROM, used twice £60 o.n.o. Beebug Discmaster, Disc Starter Pack, Toolkit £10 each o.n.o. Tel: Stockport 061-477 0315.

DISC drive, 40T SS with external PSU. Complete with formatting disc and Viglen manual £50. Tel: (024369) 3705 evenings.

MASTER 128 with 512k co-processor, twin 40/80 track drives, Hi-Res Microvitec Colour monitor and a variety of software (incl. Masterfile II, Viewspell, Graphito, Chess, Scrabble etc. and the GEM collection). All boxed complete with mouse (1987 purchases) - £1050. Also Zenith monochrome monitor £50 o.n.o. Tel: Cambridge (0223) 324902 evenings.

ACORN Z80 Second Processor, complete with software and documentation £180. Tel: 01-658 5602.

MINOR Miracles WS2000 modem, with AAL auto-answer and dial board, BBC lead, instructions. Unused since purchase. Offers? BBC model B with Acorn DFS, various ROMs and manuals - £220 o.n.o. Also full set of BEEBUG mags, offers? Tel: 01-624 3698 evenings/weekends.

TANDY CGP115 colour printer/plotter, little used, includes power supply, spare pens and paper. £60 o.n.o. Tel: Doncaster (0302) 784460.

MASTER Reference manuals parts 1 and 2, as new - £20 the pair. M. Crane, Bryn-Mor, Uffculme Road, Willand, Near Cullompton, Devon EX15 2SA.

MICRO USER tapes 1-1 to 3-12, BEEBUG tapes 1-1 to 3-9, Merlin Database. Offers - Tel: Rainham, Essex (04027) 22994.

TORCH ZEP100 and Perfect software £125. Z80 BBC Basic, Microsoft Basic, TLO Program Generator, Turbo Pascal £50 each or £275 the lot. Tel: 031-667 4180 evenings.

WANTED for BBC B, ATPL sideways ROM board, Teletext adaptor, mouse, modem, sideways RAM board, Second Processor. Cash waiting, will also consider other items. For sale: 9 inch Motorola monitor and 12 inch black and white TV. Offers - Tel: Burgh Heath (07373) 52038.

BBC B, OS 1.2 with 1770 disc interface fitted, User Guide, Advanced User Guide and 'Assembly Language Programming' (Birnbaum) £200. Can also deliver F.O.C within sensible radius. Tel: (08444) 2311.

CANON 210 single disc drive, double sided 40 track with own PSU £65 plus carriage. Tel: (0246) 418215 evenings.

BBC Model B, plus DFS and Wordwise. As new £220 o.n.o. Tel: (0742) 660251.

SHINWA CP80 printer with brand new printhead (receipt available), perfect working order - owner upgrading £120 o.n.o. Tel: (0272) 737427.

VIEW and Viewsheets ROMs with manuals £25 each. Solidisk sideways 64k RAM with software and manual £60. BEEBUG View Spellchecker and much software I can't use on my Master 128. Also double plinth for BBC B. Tel: 01-229 7329.

BBC Model B with DFS plus Wordwise. Excellent condition. £220 or nearest offer. Tel: Sheffield (0742) 660251.

BROTHER EP44 electronic typewriter (computer printer). Mains/battery, 4k memory, multi character set. £100. Tel: Hastings (0424) 429746 evenings weekends.

TEAC 100k 40T disc drive £65, Spellcheck II 40T £20, Wordbase (40/80T) £10, Elite disc £10, Aviator disc £10. Tel: Windsor (0753) 853193 after 6pm weekdays.

ACORN Z80 processor complete with ALL manuals and discs offered at special bargain price of only £255 o.n.o. Tel: Huntingdon (0480) 215092 any time.

GRAPHITO disc & manual £15; Musicsoft collection disc & manual, Mini Office II disc & manual £10 each o.n.o. Tel: (0698) 820738.

BBC MASTER £375, 3 ROM cartridges £5 each, Acorn 6502 2nd processor £125, Watford 40/80 disc drive £70, WS2000 modem + autodial £80, Epson CX21 acoustic coupler £75, all in excellent condition and boxed with leads. Tel: St Albans (0727) 66372.

NOVACAD + Plot disc £60, Wordwise Plus £20, Viewstore £25, Commstar II boxed and unused £20, all with original manuals. Tel: St Albans (0727) 66372.

BEEBUG Magazine cassettes Vol.1 No.10, Vols.2 and 3 complete, Vol.4 Nos.1 to 7 - £10 the lot + £2.50 p&p. Tel: Woking (04862) 63327.

TORCH Z80 second processor with twin dual 80T drives and Perfect software, £300 o.n.o. BBC Master ROM sockets, £5. BBC Master Peartree 32k RAM socket, £17.50. 44" Epson printer lead £5. Tel: Codnor, Derbyshire (0252) 22521.

AMBER 4000 printer for BBC Model B, prints out on 2.25" till roll. Ideal for listings, £25 or exchange for 8271 chip. Tel: Penrith, Cumbria (0768) 64890.

INTERSHEET ROM £25, Accelerator ROM £30. Tel: Lancing, Sussex (0903) 755412.

WANTED - 6502 second processor, 40/80T switchable disc drive, preferably dual with PSU, Interword. Tel: St Albans (0727) 30264.

SOLIDISK 32k SWR £12.50, Masterfile II £8, Discmaster £8, Advanced Disc User Guide £10, many 6502 books (Birnbaum, Leventhal, Griffiths etc) half price. Tel: (0705) 829346.

BBC B with Econet, Basic II, OS1.2, Disc interface, DNFS ROM £290; ATPL ROM board with 16k RAM and battery & switch £39; View A2.1 £22; Communicator 1.04A £22; CC Graphics (latest) £13; CC Printmaster (Epson) £13; Oxford Pascal £33; BEEBUG Toolkit £8; Atom Word Pack £8; "Image" (Homerton College Cambridge) - offers; Hitachi green monitor chassis £15; BBC Books. Tel: Lewes, Sussex (0273) 475499.

Business Ads

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

ADD MORE THAN 30 COLOURS to your own Basic (Mode 2) programs. Stand-alone code. Palette designer included. BBC Model B only, 40/80T disc - £7.99 inc p&p. M.J. Davis, 13 Burnham Way, London. W13 9YE.

New User Groups (U.K.)

SCOTLAND (Northeast)

Grampian Amateur Computer Society holds weekly meetings in Aberdeen. New members and visitors always welcome. Contact Paul Cuthbertson on (0467) 24030 or Bruce Edelsten on (0224) 639911.

BURTON-UPON-TRENT

For details of the BBC User Group in this area contact Mrs Linda Yeomans, 13 Regent Street, Church Gresley, Burton-Upon-Trent, Staffs DE11 9PL, or phone (0283) 216445 (after 2pm).

New User Groups (Overseas)

ZIMBABWE

Green Screen Club
P.O.Box U.A.393
Union Avenue,
Harare,
Zimbabwe.

SOUTH AFRICA

User Group of Pretoria,
P.O.Box 32798
Glenstantia 0100
South Africa.

POINTS ARISING POINTS ARISING POINTS ARISING POINTS ARISING

BEEBUG Filer Vol.5 No.9

In adding the facility for repeated addition of records a bug has arisen which is apparent when attempting to add a record to a full file. To correct this, change line 8090 to be line 8110 and add line 8225:

```
8225IF rec>recn PRINT"File full":flag=1
```

Streamlining Data Entry on the Beeb Vol.5 No.10

The program gives the error message 'Bad DIM at line 100'. Change line 100 to read:
100 DIM os 40:ON ERROR GOTO 920

Turn Your Beeb into a Chart Recorder Vol.5 No.10

The listing of the program on page 58 appears corrupt as lines 4100, 4160, 4180 and 4200 are repeated out of order. Other lines have the initial digit of the line number missing. The relevant lines should read:

```
4100 T=T*60+VAL(T$)  
4110 =T  
4120 :
```

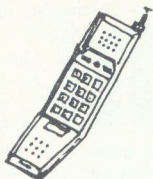
```
4130 DEF Fntimecomp  
4140 TM2$=RIGHT$(Title$)  
4150 =1 AND TM$=TM2$  
4160 :
```

MIDAS Vol.6 No.1

The argument of the procedure PROCr is missing in line 1440 which should read:
1440 :CALL &FFDD:PROCr(F\$):PROCF:PROCD:PROCV:ENDPROC

The program also performs better if PROCD in line 1400 is replaced by PROCsh. Note that the program does not recognise all file types in all circumstances. We hope to clarify this in the next issue. Our thanks to those readers who have phoned in about this aspect of MIDAS, which has proved to be a very popular program with readers.

THE COMMS SPOT



Peter Rochford discusses the ideas behind the inexpensive intelligent modems now becoming available to the home user, and the feasibility of higher baud rates.

In this month's column we discuss the general features of 'intelligent' modems. These latest generation modems boast high levels of sophistication and yet are easy to control from your own micro using simple commands. In the next issue, I hope to review six of these latest modems.

SMART?

So, what makes a modem 'intelligent'? Well, simply the fact that, like your Beeb, they contain a microprocessor and controlling software, together with some RAM for storage and workspace. These new modems are in effect small microcomputers. They communicate between the terminal (your Beeb) and the host computer to which you are connected, checking on what each end of the communication link is doing, acting accordingly, and passing data from one to the other.

Bear in mind that many of the features found in 'smart' modems can also be obtained with an ordinary modem when combined with one of the sophisticated communications ROMs now on the market.

The main advantage of a 'smart' modem is when used in conjunction with a relatively unsophisticated micro, or one that does not have the facility to plug in communications ROMs. Another advantage would be if you required your micro to be performing other tasks at the same time.

AUTO-BAUD-RATE DETECTION

Smart modems can set up their own baud rate to match the incoming signal. They can also detect the transmit/receive baud rate that you have set up on your terminal, allowing you even to have the terminal set to 1200/1200 baud whilst the host is operating at 1200/75 baud!

AUTO-DIAL

All of the latest intelligent modems feature auto-dial, this being automatically carried out by the modem after a command is issued from the terminal. You can, however, abort the process from the terminal by issuing an 'Escape' command.

Checking on the progress of your call with older modems is usually limited to keeping an eye on the carrier-detect light to establish whether you have managed to make a connection. If the line was engaged or out of order, often you had no real idea unless you lifted the telephone receiver to check. Therefore I really welcome the inclusion of a speaker for audible monitoring of call progress. There is definite satisfaction in being able to hear what is happening on the telephone line, instead of a stony silence except for the initial clicking of the modem's dialling relay. Once a carrier has been detected, then the modem will automatically mute the loudspeaker.

Many of the smart modems I have tested have front panels bristling with LEDs. These are for monitoring the status of the modem and the main RS232 lines. This is another feature that I regard as essential on a modem. The more you about what the modem is doing the better.

These new modems really live up to their name when you discover that they will tell you in plain English on your screen how your call is progressing, and provide feedback about the state of the telephone line. If it is engaged, the modem can detect this, report to the terminal and display 'BUSY'. Having aborted the call, the line is re-set and the modem will then auto re-dial the number after a preset time interval. Some smart modems can detect whether there is a dial tone before attempting to dial. If none exists, they drop the line and inform the user by displaying 'NO DIAL TONE' on the screen, instead of trying to dial a dead line and wasting your time. More sophistication is available on certain modems; these actually detect whether the exchange line will allow tone or pulse dialling and act accordingly. It is possible by monitoring this type of feedback to write some very sophisticated control programs in any of a number of computer languages. These programs can be written in such a way that they can be

used by those who have little knowledge of comms, enabling them to access on-line information services very easily.

AUTO-ANSWER

Intelligent modems feature auto-answering. This is not new, as many older modems are capable of this too. However, with their on-board microprocessor, some smart modems can do much more than just answer the line and hand over communication to the terminal software. They are able to answer the line and accept data without being connected to a computer, either storing it in their own RAM, or dumping it to a printer.

CONTROL

Control of most smart modems is via the terminal software, and many of these modems have few, if any, front panel controls. Software control can be carried out in one of two ways. This is either by the V25 'bis' command set or by the Hayes command protocol.

V25 COMMANDS

V25 is the less favoured of the two now, most modems being Hayes-only, although some allow both. V25 commands are issued from the terminal in local mode i.e. not connected to a host system. The command set allows all the various functions of the modem to be controlled and takes the format: CTRL-B <command> CTRL-C; CTRL-B alerting the modem that a command is about to be sent, CTRL-C being used as a terminator. For example, CTRL-B CTRL-N1234567 CTRL-C will tell the modem to auto-dial the number 1234567.

HAYES PROTOCOLS

Hayes protocols are the more widely used and were devised by Hayes Microcomputer Products in the U.S. They have since been adopted worldwide as a standard for controlling intelligent modems. Like all 'standards' however, the Hayes ones are already subject to variations. There are around sixteen commonly used commands, but other modem manufacturers have added their own to the set, with people like Pace featuring over 50 on their Series Four modem.

The Hayes protocols are a very simple set of instructions to control a modem from quite basic terminal software. All

the commands are issued in local mode, the letters 'AT' (for attention) being typed in at the terminal keyboard to alert the modem that an instruction is about to be sent. After the 'AT' comes the actual instruction. All Hayes compatible modems feature the ability to store telephone numbers for recall. Once the modem has accepted an instruction it will respond by sending an 'OK' to the terminal to let you know that it understood, or else return 'ERROR' if sent an invalid command. If the instruction results in the modem dialling a host and going on-line then it will leave the command state and disregard any further AT commands.

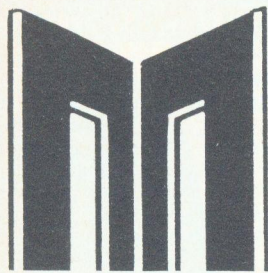
As well as the simple AT commands, Hayes-compatible modems also have a feature known as 'S registers' held in the modem's battery-backed RAM and used to configure the modem to suit the user's own personal needs. This allows the modem to operate in a certain way each time it is switched on. However, a default set of parameters is put in at the factory.

CONCLUSION

I have given but a brief glimpse of what these new intelligent modems can do. They offer some powerful facilities and great flexibility whilst retaining ease-of-use, making older modems seem rather crude and laborious to use, unless they themselves have fairly sophisticated controlling ROMs. At present, some of the intelligent modems are rather expensive, particularly those that are to the V22bis (2400/2400 baud) standard. V22bis modems will hopefully drop in price fairly soon, as the demand increases when BT introduces its new VASSCOM network. This will allow Prestel and Telecom Gold access at 2400/2400. However, even the V21/V23 intelligent modems are falling in price, and are well worth investigating for what they can offer.

In the next issue we will be featuring a review of six intelligent modems. Three of these are to the V22bis standard and at the top end of the price scale. More realistically priced V21/V23 modems will be featured too for the less affluent, like myself. But once you have accessed Prestel at 2400/2400, nothing else will do, I can assure you!

B



THE MASTER PAGES

Devoted to the Master Series Computers

Firstly, a word of thanks to those who have responded to our plea for Master items. Some of these will appear in the next issue – but keep your contributions rolling in. Please mark your envelope "Master Pages".

Amongst this month's features, we have an extension to the World by Night and Day, which uses the Master's real time clock, and which automatically updates the world map every 10 minutes.

There is also an item on using disc menus to boot into Wordwise, View or any language ROM of your choice.

David Graham



MASTER
SERIES

W.C.
CORBETTS'
WORLD

Add these extra lines to the 'World by Night and Day' on page to get a real-time display.

Elsewhere in this issue you will find described a program that produces a detailed graphical display showing the areas of light and darkness covering the earth at any date and time in the year. If the same program is to be run on a Master then various enhancements and improvements can be made to the basic program. Some of these changes can also be applied to the Compact, and any other machine fitted with the graphics Extension ROM (GXR). Indeed, the Worldday program as originally submitted was for a Master only, and contained all these enhancements. We have stripped these away for the benefit of model B users.

The real-time clock fitted to the Master 128 means that the Worldday program can be modified to provide a real-time display showing night and day for the present date and time. This display is automatically updated every ten minutes. In addition, the extended PLOT commands available on the Master (and Compact and in the GXR) enable the graphics to be programmed much more simply, particularly with regard to the 'fill' routine.

The extended memory space of the Master and Compact means that the basic program can be readily extended. The additional lines are all listed together, but you may wish to add only the sections relevant to your system. This is described below. If you add all of these lines you will have a version of the program (as on the magazine cassette/disc that is intelligent enough to adapt itself to whichever machine it is run on (model B, B+, Master or Compact). By deleting certain lines you can alternatively produce a version that is specific to the Master.

REAL-TIME CLOCK

Add lines 120, 160-1210, and 1290-3080 to the original program, keeping strictly to the line numbering given. When you now run the program, you are given a choice. You may enter any date and time of day for the night and day display as before, or you may simply set a "continuous" display running. Then, provided you have correctly set the Master's clock, the screen will show on a map of the world the current division between daylight and darkness. U.K. time (GMT or BST) is shown on a clock-face in one corner of the screen, and the display is re-drawn every ten minutes. As the day progresses you will see the sun, with the boundaries of daylight – the lines of sunrise and sunset – steadily moving westwards.

PROCclk uses the Master's economic circle-filling commands to superimpose an analogue clock-face. Once set running, the clock ticks on every minute, and the daylight/darkness display is re-drawn every ten minutes.

The program assumes you will re-set TIME\$ to reflect British Summer Time, and automatically adjusts for this whilst Summer Time is in force during 1987. To change this for future years, refer to the description of the basic version elsewhere in this issue.

GRAPHICS EXTENSIONS

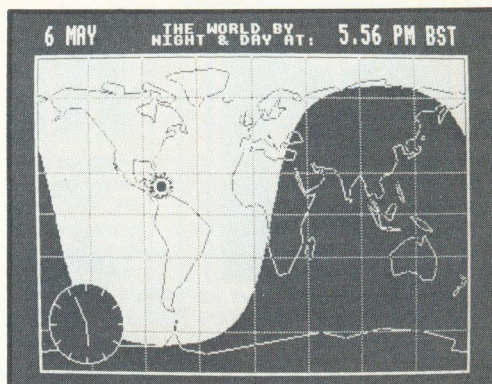
Add lines 130, 1220-1240, 3100-3330 from the accompanying listing. The program will then take advantage of the extended PLOT commands. In particular, PLOT133 can completely replace the use of the fill routine at lines 5000 to 5200. These lines may now be deleted if the program will always be running on a Master, Compact or other machine with the GXR fitted. Similarly, lines 3100 to 3170, and 3210 to 3290 may also be deleted as their function has been replaced by PLOT149 and PLOT157.

There you have it: a display for a selected date and time, or one running in real time, either of which is likely to fascinate (and instruct) young and old alike.

```

120 rtc%=FNrtc
130 mos%=FNmos
160 IF f%=1 PROCset:tb$=set$ ELSE tb$=
FNtime
180 ON f% GOTO 200,220
210 REM * Continuous display
220 PROCscr
230 REPEAT
240 PROCclk
250 UNTIL tn%MOD10=0:GOTO 220
260 :
1110 DEFFNrtc
1120 A$=MID$(FNtime,8,3)
1130 RESTORE 4290
1140 READ a$,b$,c$:IF A$=a$:=TRUE
1150 IF a$<>"Dec" GOTO 1140
1160 =FALSE
1170 :
1180 DEFFNtime
1190 ?&100=0:A%=&E:X%=0:Y%=1:CALL&FFF1
1200 =&0100
1210 :
1220 DEFFNmos
1230 A%=0:X%=1:=(USR(&FFF4)MOD&1000)DIV
&FF
1240 :
1290 IF f%=2 PROCface:PROChands
1320 DEFPROCclk
1330 T%=TIME+6000:REPEAT:tT%=TIME:UNTIL
tT%>=T%

```



```

1340 tb$=FNtime:PROChr:PROChands:ENDPRO
C
1350 :
1410 IF rtc%=0:=1
1420 PRINTTAB(0,8)
1430 PROCcen(C$+"Enter your choice:")
1440 PRINT:PROCcen(C$+"1 - Single date
and time")
1450 PROCcen(C$+"2 - Continuous display
")
1460 REPEAT
1470 R%=GET-48
1480 UNTIL R%>0 AND R%<3
1490 =R%
2970 DEFPROCface
2980 GCOL 0,0:PROCfcircle(150,200,112):
GCOL 0,1:PROCccircle(150,200,112)
2990 GCOL 0,1:FOR K=0 TO 2*PI STEP PI/6
:MOVE 150+SIN(K)*94,200+COS(K)*94:DRAW 1
50+SIN(K)*112,200+COS(K)*112:NEXT:ENDPRO
C
3000 :
3010 DEFPROChands
3020 GCOL0,0:PROCfcircle(150,200,84):ch
a=VAL(hr$)*30+VAL(mn$)/2
3030 chx=150+(SIN(RAD(cha))*60):chy=200
+(COS(RAD(cha))*60)
3040 GCOL0,1:MOVE150,200:PLOT5,chx,chy:
chm=VAL(mn$)*6
3050 cmx=150+(SIN(RAD(chm))*80):cmy=200
+(COS(RAD(chm))*80)
3060 GCOL0,1:MOVE150,200:PLOT5,cmx,cmy
3070 ENDPROC
3080 :
3100 IF mos%>=3:MOVE x,y:PLOT 149,x+r,y
:ENDPROC
3210 IF mos%>=3:MOVE x,y:PLOT 157,x+r,y
:ENDPROC
3330 IF mos%>=3:PLOT 133,X%,Y%:ENDPROC

```




MASTER
SERIES

WORK DISC
GENERATOR

This idea from David Graham takes all the work out of generating new ADFS work discs.

Because of the greater capacity and flexibility of the ADFS, it is quite viable to group files for many different kinds of computing activity on to the same disc, using different directories for each. But when the disc becomes full, it is a tedious task to prepare another. The new disc must be formatted, various directories must be created, and certain files, such as library routines, text headers, etc, must be copied across from the previous work disc. All this can be automated, however, using a simple EXEC file created on the Editor, or with View or Wordwise.

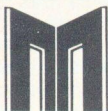
To illustrate the idea, the accompanying EXEC file is the one which I use for creating monthly work discs. Yours will obviously look different, but the principles are easily adapted. When this file is EXECed in from drive zero, it sets up mode 128, then beeps, and prompts for confirmation. An upper or lower case "y" sets it in motion. It first formats the disc in 80 track large format then verifies it. To do this it calls commands in the new BEEBUG Master ROM, which must be resident in your machine. If you have another ROM which you use for formatting ADFS discs, you may need to alter the syntax a little. If you have only the formatter on the Welcome Disc, you must skip this stage, and supply a previously formatted disc, since the Acorn formatter will not let you out of its clutches, once called.

Next, a set of directories is created using the CDIR command, and the root directory catalogued. A number of files are then individually copied over to various directories. In this particular case, the boot file, a menu and a one line EXEC file used for setting the Library directory are copied into the root directory of the new disc, then three copies of a Wordwise text header are copied over. Finally, the entire contents of the Library, Utils and Carryover

directories are copied across to the new disc.

After the copying process has finished, a NEW is performed to clean up the program storage area, since the COPY command writes all over this. The Library directory is then catalogued, and then the boot options are set up on the new disc, via a function key, called by the FX command in the final line. To ensure the correct operation of this last line, it must be followed by a blank line in the EXEC file.

```
*|>FMW1
MODE 128
*|
*| *****
*|      CREATE WORK DISC
*|      ON DRIVE 1
*|      Including Formatting
*|      (ver. 6/5/87)
*|
*|      Ok Y/N ?
*|
*| *****
*|
SOUND1,-15,50,2:REPEAT UNTIL INKEY(-69) OR
INKEY(-86)
IF INKEY(-86) THEN *FX125
*FX15,1
*FORMAT 1 L
YY
*VERIFY 1
*CDIR :1.$.%MAGPROG
*CDIR :1.$.%MAGTEXT
*CDIR :1.$.%MASTERTXT
*CDIR :1.$.%MASTERPRG
*CDIR :1.$.%LIBRARY
*CDIR :1.$.%CARRYOVER
*CDIR :1.$.%UTILS
*CAT :1.$
*COPY :0.$.!BOOT :1.$
*COPY :0.$!.L :1.$
*COPY :0.$!.menu3 :1.$
*COPY :0.$!.+magheader :1.$
*COPY :0.$!.+magheader :1.$.%MAGTEXT
*COPY :0.$!.+magheader :1.$.%MASTERTXT
*COPY :0.$.%LIBRARY.* :1.$.%LIBRARY
*COPY :0.$.%UTILS.* :1.$.%UTILS
*COPY :0.$.%CARRYOVER.* :1.$.%CARRYOVER
NEW
*CAT :1.$.%LIBRARY
*KEY0 *DIR :1.$|M*OPT4,3|M*DIR
:0.$|MSOUND1,-15,200,2|M
*FX138,0,128
```

MASTER SERIES

Master
Hints

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

ADT TOOLKIT BASIC LOAD

ACP's excellent ADT Toolkit, used by many Master owners, behaves unexpectedly when loading Basic programs from its disc menu. It resets the value of PAGE to match the value of PAGE at which the selected program was originally saved. Thus if you save a Basic program on a model B, then run it from a Master using the ADT menu, you will find that PAGE on the Master has been raised to &1900. If you try to escape from the program by pressing Break, the program will vanish. This is because pressing Break resets PAGE to its default value of &E00. To recover the program, you need to reset PAGE to &1900.

GROUPING ADFS DIRECTORIES

When you catalogue an ADFS disc, or use a disc menu, ADFS directories do not tend to stand out very well. You can improve the situation in two ways. Firstly, it helps somewhat if you use upper case for directory names, and lower case for filenames. But even better, you can cause directory names to be grouped together at the start of the catalogue or disc menu listing, by using the percent character "%" as the first character of each directory name. Percent is preferable to the pling character (!) here, as it allows any !BOOT files to remain at the top of any listing.

DOUBLE BAR FUNCTION KEY VDU

In the April issue, we said that you could not terminate VDU calls within a function key on the Master or Compact with the bar character "|" because this confuses the operating system. Thanks to Andrew Benham and Alastair Taylor for pointing out that you CAN terminate VDUs in this way, provided that you use a double bar "||" instead of just one. Thus to program f0 to turn off the cursor, use:

```
*KEY0 VDU23,1|||M
```

The string is terminated with "|M" to give a carriage return, thus making three bar characters in a row.

WHERE IT'S AT

The character "@" can be used with many ADFS commands to signify "currently selected directory". But there is a pitfall for the unwary. When you are using this specifier, remember that it not only signifies "currently selected directory", but, implicitly, "currently selected drive". If you try to give drive numbers, odd things happen. Thus, the following:

```
*MOVE -DISC-:1$.filename -ADFS-:0.@.newname
```

will copy the file to the ROOT directory on drive zero rather than the currently selected directory. To get it right, just remove the ":0." from the above.

SPEEDING UP REPEATED *LOADS

If you are performing a number of *LOAD operations on the ADFS from a directory other than the currently selected directory, you can achieve considerable speed gains as follows. First specifically select the directory containing the files to be loaded, perform the load operations, then return to the required directory. The reason for the speed gain is the considerable time overhead of temporarily selecting the source directory prior to each load, and then returning to the currently selected directory after each operation.

CMOS RAM DISPLAY

Following the brief article in the Jan/Feb issue giving a routine to load and save CMOS RAM values, a number of readers have suggested a formatted display option for the program. The following is an amalgamation of the ideas submitted. The lines should be appended to the published listing. Their effect is to add a third option (D) to display values to the screen. The full program appears on this month's magazine cassette/disc. Thanks to R.F. Swallow and others for their suggestions.

```
5 MODE0
20 PRINT"Read, Write, or Display RAM (R
/W/D) ?"
45 IF A=68 THEN PROCdisp
1140 DEFPROCdisp
1150 A%=161
1160 FOR X%=1 TO 49
1170 Z%=(USR(&FFF4)AND&FF0000)DIV&10000
1180 PRINTTAB(-20*(X%>24),X%MOD25-1)X%,Z%
1190 NEXT:ENDPROC
```




MASTER
SERIES

AUTO-BOOTING
ADFS MENUS

**Lee Calcraft presents
an idea for auto-booting
files from ADFS menus
into the language ROM
of your choice; whether
this be View, Viewsheet,
Inter-word or Wordwise.**

In the issue following the publication of Peter Rochford's ADFS menu (BEEBUG Vol.5 No.5), we suggested an enhancement for Wordwise and View users. If files from either of these languages were selected, the corresponding language ROM would be engaged, and the file automatically loaded. Since then we have had a number of requests for modifications for other language ROMs such as Inter-Sheet, Inter-Word and so on. This article is a response to those requests.

In fact, the technique to be used here is much more flexible than the method first proposed. Moreover, it has been incorporated into two new ADFS menus: a fast machine code menu which we shall be featuring in the next issue, and the more sophisticated menu which is built into the Master ROM. Both employ the technique described here, and users of either of these two menus may pick up some tips on customising from this article.

The principle involved is quite straightforward. Once a file has been selected from the menu by the user, the menu software checks to see if it is a Basic program or a ROM image. If so the file is RUN or *SRLOADED as appropriate. If not, the menu software stores the selected filename in RAM (at &100), and then performs the following:

EXEC !MENU

In other words, it EXECs in the first file which it comes across in the currently selected directory whose name begins with "!MENU".

The idea is that the user should first create one or more !MENU files to call the software of his choice, and load his selected file. He keeps a View !MENU file (perhaps called !MENUview for identification purposes) in all directories which will hold View files, an Inter-Word !MENU file (maybe called !MENUiword) in all directories which will contain Inter-Word

files, and so on. The only drawback to this method is that you cannot use more than one !MENU file in any given directory. However, this is not likely to prove too problematic.

As we will see in a moment, creation of the individual !MENU files is all accomplished very easily. Moreover, the system allows extreme flexibility. For example, the !MENU file for Wordwise could transfer across the date from the Master clock, and set up function keys, initiate a printer buffer, send special codes to the printer etc. The file for View could set the mode, the function keys, and so on. The method is also much quicker than that previously suggested for Wordwise and View files, since it does not need to write to the disc to transfer information across to the language ROM.

MODS TO THE ROCHFORD MENU

To make this technique operate on the Rochford ADFS menu requires very few changes to the program. But you must modify the right version of the program! The simplest is to use the version supplied on the magazine cassette/disc for Vol.5 No.5. Alternatively, use the program as listed in the magazine itself (Vol.5 No.4), but with the bar character (!) corrections given in Vol.5 No.5.

First, delete the Wordwise and View additions, if they are present in your version, as follows. Delete lines 242 and 245, and lines 1000 to 2210. Then just change lines 280 and 290 to read:

```
280 MODE7:PROCK:IFTUBE=TRUE THEN PAGE=
&800 ELSE PAGE=&E00
290 $&100=AS(F%-64):OSCLI("EXEC !MENU*
"):END
```

You should now find that if you run the menu and select any file which is not either a Basic program or a ROM image, the program should search for a file in the currently selected directory whose name begins with "!MENU". It is now a matter of generating some !MENU files appropriate to the various language ROMs that are to be used in conjunction with the menu. Any of these files will work equally well with the Rochford ADFS menu modified as above, next month's machine code ADFS menu or the enhanced menu in the newly released Master ROM. We should add that machine code programs are not trapped by the menu (since View files are sometimes

indistinguishable from them), so that if they are to be loaded in and run by either of the first two menus, they will need a special !MENU file of their own; and one is listed below for this purpose. On the Master ROM, a special option allows machine code files to be *LOADed or *RUN without calling a !MENU file.

PICK & MIX !MENU FILES

A selection of !MENU files accompanies this article for a variety of different purposes. The best way to create such files is to use the Master Editor, and save the file so created using the SAVE option, f3. Alternatively, you could use Wordwise or View. Either way, you should leave a single blank line at the end of each file.

HOW IT WORKS

The principle is very simple, as you can see from the selection of !MENU files listed. The essential part of each file involves loading function key f0 with a set of instructions to call the desired language, and then to get the language to load the file whose name is stored in memory at \$100. This area of memory coincides with the very bottom of the 6502 stack, and should cause no conflicts, since the filename is a maximum of 10 bytes in length. The FX call immediately following the key definition simulates the pressing of f0. Any commands which follow this should be in a form which is understood by the language ROM called by the function key, rather than by Basic.

Using these principles it should be possible for users to create their own !MENU files to suit their precise needs, and to boot into even the most offbeat language ROMs and other software.

WORDWISE 1

Enter Wordwise, and load file.

```
OSCLI ("KEY0 *WORDWISE|M:NEW|M2"+$&100+"|M")
*FX138,0,128
```

WORDWISE 2

Turn on Master ROM printer buffer, take date from Master clock, remove leading zero from single-digit dates, call Wordwise, install adjusted date in Wordwise variable D\$, install selected filename in Wordwise variable F\$, load in selected file. Note D\$ and F\$ may be

inserted into text with <f1>PS D\$<f2> etc.

```
*BUFFON
day$=MID$(TIME$,5,2)
month$=MID$(TIME$,8,3)
year$=MID$(TIME$,12,4)
IFLEFT$(day$,1)="0" THEN day$=" "+RIGHT$(day$,1)
$&380=day$+" "+month$+" "+year$
OSCLI ("KEY0 *WORDWISE|M:D$="+CHR$34+$&380+CHR$34+"|M:F$="+CHR$34+$&100+CHR$34+"|M:LOAD TEXT F$|M|M")
*FX138,0,128
```

VIEW

Turn on Master buffer, select mode 128, call View, load in selected file.

```
*BUFFON
OSCLI ("KEY0 MO.128|M*WORD|MLOAD "+$&100+"|M")
*FX138,0,128
```

VIEWSHEET

Set printer (Epson) to condensed text, set mode to 128, enter Viewsheet, load selected file.

```
VDU2,1,15,3
OSCLI ("KEY0 MO.128|M*SHEET|MLOAD "+$&100+"|M")
*FX138,0,128
```

INTERWORD

Select InterWord, load selected file. This will work if there is already an InterWord file in memory.

```
OSCLI ("KEY0 |M*IWORD|M2"+$&100+"|M|M")
*FX138,0,128
```

INTERSHEET

Select Intersheet, load selected file. This will NOT work if there is an Intersheet file currently in memory.

```
OSCLI ("KEY0 *ISHEET|M2"+$&100+"|M|M")
*FX138,0,128
```

MACHINE CODE

Load and run machine code program.

```
OSCLI ("KEY0 *RUN "+$&100+"|M")
*FX138,0,128
```

For convenience, we have included on this month's magazine cassette/disc a copy of the original Peter Rochford ADFS menu program modified to work immediately with the menu files as described here. A file containing all the example menu files is also provided so that you can edit this to produce your own menu files as you require.

HOW TO WRITE AN ADVENTURE GAME



Jonathan Temple, our own games writing expert, takes the lid off writing adventure games with a complete do-it-yourself game skeleton.

Many people enjoy playing adventure games and may, at some time or another, have toyed with the idea of writing their own. However, there are certain parts of adventures which can prove quite complex to program: so although someone may well have good ideas regarding plots and puzzles, they may find that they haven't the programming skill to put these together in a finished game.

Of course, there are now several "Adventure generators" on the market (GAC and The Quill being the best known examples). All of these are based on the same simple idea, which is that the "nuts and bolts" of an adventure game remain the same, whatever the scenario and plot. Whether a game is set in Middlesex or Middle Earth, it must still have a parser to interpret commands, variables holding information about the various objects and rooms, and some way of displaying messages. These aspects are usually also the most complex parts to write.

For this reason, many software houses use adventure "shell" programs or "skeletons". These consist of the basic parser, message system etc., and can be used over and over again - all the adventure writer need do is add the vocabulary, rooms, objects, messages and "verb programs" appropriate to the adventure being written.

The program presented here provides a skeleton program which you can use to write your own adventures. Although not quite as sophisticated as some commercial products - the parser only accepts verb/noun input, and there is no text compression - it is quite adequate for many purposes, and adventures produced with it will prove considerably more advanced than those in many magazines. Part one, this month, describes the main features of the skeleton program, while part two completes this description, and shows how to implement a complete adventure game called "Jungle Adventure".

The features of this particular "adventure skeleton" include automatic word-wrap for all messages, "IT" to refer to a previous object, and many commands already built in. The latter require no work on your part for them to be understood by the program. For instance, movement in six directions is automatic (with the program handling N,S,E etc.). The skeleton program also contains the necessary coding for the essential commands LOOK, INV, QUIT, SCORE, SAVE, and LOAD/RESTORE.

ENTERING THE PROGRAM

After typing in the program, save it before trying to write any adventures. Some bugs could corrupt the entire program, so be careful! In addition, you MUST keep to the line numbering - otherwise you will find it extremely difficult to add the lines for next month's "Jungle Adventure".

WRITING YOUR OWN GAMES

So, how do you set about writing an actual game? Well, as with any adventure, it is best to plan out as much as possible on paper first. Once this has been done, you can start work on the actual program.

All adventures involve creating lists of data for the rooms, objects, verbs, nouns, and messages for the game. For this program, the actual data is entered in DATA lines, starting from 5000, as explained below. The skeleton presented here also provides several procedures for accessing this data. Always use these procedures and functions rather than trying to access the data directly.

ROOMS: Each room has a room description, a status flag and six other numbers.

These numbers refer to the rooms that can be reached from the current room, in each of the six directions N, S, E, W, U and D. There is an absolute maximum of 255 rooms imposed by the program. There are three further points to note: if movement in one of the six directions is illegal, then the "room reached" should be set to zero; room descriptions are automatically preceded by "You are"; and the room state flags are all set to zero at the start of each game.

As to access, PROCrf(R,V) will set room R's flag to value V, and PROCRe(R,E,V) will set the room travelled to in direction E from room R, to room V. (This latter is especially useful for opening up new exits, such as secret passageways etc.) There are also two corresponding functions, FNrf(R) and FNre(R,E), which will return the information above. With both FNre and PROCRe, E should be in the range 0-5, representing N,S,E,W,U or D.

A room's description can be displayed at any point in the game by using PROCdesc(R). PROCdesc(0) will provide the player's inventory (though the INV command is already built in). As to the actual data format, each room should have a description followed by six numbers for the directions.

OBJECTS: Each object has three variables - its location, the number of the message which describes the object, and the object's state flag. These should be entered in DATA statements after the rooms, and all the numbers should be in the range 0-255.

If an object does not "exist" (such as a cake that has been eaten) at any stage in the game, it should be moved to a room which cannot actually be reached by the player, perhaps room 255. 'Room' 0 should not be used for this purpose, since this is the room in which all the objects the player is carrying are stored.

The state of any object may be changed during the game using PROCOr(N,V), PROCOn(N,V) and PROCOf(N,V) which set V as object N's room, message number and flag respectively. There are three corresponding functions, FNOr(N), FNOn(N) and FNOf(N), which return the values of each.

An example may help: suppose that object 1 was "a lamp", started in room 6

and was off (say, state 0) at the start of the game. Messages 1 and 2 are "a lamp" and "a lit lamp" respectively. The data for this object would be 6,1,0. At a later stage in the game, the lamp might be switched on. In this case, PROCOn(1,2) would be used so that the object's message number becomes 2 (object's description then "a lit lamp") and PROCOf(1,1) would be used to set its state to 1 (on).

The VERBS come next: these are stored from line 7000 onwards. Each verb has a verb number and a number representing how many words are expected. For instance, verb number 1 might consist of both GET and TAKE. Since a noun is also expected, the "words expected" figure should be 2.

For the command HELP, a second word is not expected, so the "expected" figure would be 1 for this verb. In certain cases, however, you may wish the "expected" figure to be ignored. For instance the player might expect both "SWIM" and "SWIM RIVER" to be allowed by your game. For this type of verb, set the "words expected" to zero.

The verb numbers are used by the skeleton program to determine which routine is needed in response to input. You will need one routine for each verb, and these should be programmed between lines 1680 and 2480. The initial line numbers for each of your verb routines should then be entered into the ON-GOSUB at line 1400.

Taking the two example verbs 1 and 2 above, GET/TAKE and HELP, line 1400 might begin:

```
ON V GOSUB 1680, 1720
where line 1680 is the first line of the
"Get object" routine and 1720 is the "Help
player" routine. When writing verb
routines, note that the variable N
provides the number of the object typed
with the verb (if any), and that E will
give the number of words typed. Each verb
routine (a basic sub-routine) should
finish with a RETURN statement.
```

Several aids to writing verb routines have been provided. The functions FNavail, FNheld and FNhere will check if the object N is available (held by the player or in the same room), held by the player, or in the same room respectively. If not, these functions return a FALSE result and

display a suitable message. So the line:

```
IF NOT FNhere THEN RETURN
```

is all that is needed to check that object N is at the player's current location, and to print a suitable message and return for the next command if not.

In addition, the two procedures PROCget(N) and PROCdrop(N) will allow object N to be got or dropped, if it is in the player's location or held by the player, and a suitable message displayed.

The actual data format for the verbs is the verb (which must be in lower case), followed by the verb's number and the words expected. So for GET in the example above the data would be: get,1,2

NOUNS have a somewhat simpler format: the data should be made up by the noun (again in lower case) followed by the noun number. The main thing to remember is that the noun numbers should match up with the objects, so if object 1 is "a rusty old oil lamp" then noun 1 should be "lamp".

However, the skeleton contains two useful extra features regarding noun handling. If the noun's value is -1, then the program displays a "You can't do anything with that" message, whatever the verb; for noun value -2 the program prints "You can't do that."

Why is this useful? Well, players often find that something is mentioned in a room description or similar, but any reference to it is not recognised by the program! The best games usually print a "That's just scenery" message, and the facility above makes this very easy indeed!

NEXT MONTH

Obviously, there is a lot to take in, and careful study of this article is needed to make good use of the adventure skeleton. In part two, we will complete the description of the adventure game skeleton by taking a look at "messages". We will also present the extra lines necessary to create "Jungle Adventure", a short mini-adventure. It will also explain how certain aspects of the game, including the puzzles, have been implemented, so that you can see exactly how to write your own adventures using the skeleton program presented here. In the mean time, you may wish to start planning your own next adventure in anticipation.

```
10 REM Program Adventure Skeleton
20 REM Version B1.3
30 REM Author Jonathan Temple
40 REM Beebug June 1987
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO 210
110 MODE 6:VDU 19,0,4;0;
120 r%=0:t%=0:f%=0
130 v%=0:n%=0:m%=0
140 PROCinit
150 CLS:PRINT''
160 PROCreset
170 IF FNgame GOTO 150
180 CLS:PRINT''
190 END
200 :
210 IF ERR=17 GOTO 150
220 IF ERR>128 REPORT:GOTO 170
230 CLS:PRINT'':REPORT
240 PRINT" at line ";ERL:END
250 :
1000 DEFFNgame
1010 REPEAT PROCdesc(R)
1020 REPEAT L=R:PROCbefore
1030 IF end% OR L<>R GOTO 1420
1040 E=1:V=0
1050 A$="":B$="":X$="":Y$=""
1060 REPEAT INPUT'':" $J%:L%=LEN($J%)
1070 UNTIL L%
1080 FOR N%=1 TO L%
1090 IFN%?K%>64 IFN%?K%<91 N%?K%=N%?K%
OR32
1100 NEXT
1110 IFL%?K%=32 L%?K%=13:L%=L%-1:GOTO11
10
1120 N%=INSTR($J%," ") :X$=LEFT$( $J%,N%-
1):A$=LEFT$(X$,4)
1130 IFN%>0 GOTO1200
1140 PROCdirs:IFN%<6 GOTO1410
1150 IFA$="l" OR A$="look" OR A$="rede"
L=0:GOTO1420
1160 IFA$="i" OR A$="inv" OR A$="inve"
PROCdesc(0):GOTO1420
1170 IFA$="q" OR A$="quit" end%=3:GOTO1
420
1180 IFA$="scor" PRINT"You have scored
";SC; " out of ";TS:GOTO1420
1190 GOTO1280
1200 Y$=MID$( $J%,N%+1)
1210 IFASCY$=32 Y$=MID$(Y$,2):GOTO1210
1220 B$=LEFT$(Y$,4):E=2
1230 IF B$="it" B$=Q$ ELSE Q$=B$
1240 IFA$="go" OR A$="walk" OR A$="run"
OR A$="move" A$=B$:X$=Y$:E=1:GOTO1140
1250 IFA$="save" PROCfile("SAVE",Y$):GO
TO1420
1260 IFA$="load" OR A$="rest" PROCfile(
"LOAD",Y$):L=0:GOTO1420
1270 IFA$="exam" IFB$="room" L=0:GOTO1420
```



```

1280 N=0:V=FNchk(0,v%,A$):IFV X$=W$(N%)
:GOTO1310
1290 C=FNchk(n%,w%,A$):IFC=0 Y$=X$
1300 GOTO1370
1310 IF E=1 OR ASCB$=34 GOTO1390
1320 N=FNchk(n%,w%,B$)
1330 IFN=-1 PROCline("You can't do anyt
hing with the "+Y$+"; try something else
."):PRINT:GOTO1420
1340 IFN=-2 PROCe(1):GOTO1420
1350 IFN Y$=W$(N%):GOTO1390
1360 C=FNchk(0,v%,B$)
1370 IF C PRINT"Eh?" ELSE PRINT"I don't
know the word """";Y$;""""
1380 GOTO1420
1390 IF W% IF E<W% PRINT"Eh?":GOTO1420
1400 ON V GOSUB
1410 PROCafter
1420 UNTIL end% OR R<>L
1430 UNTIL end%
1440 IF end%=2 PROCline("Congratulation
s, you have completed the game with a fi
nal score of "+STR$(SC)+" out of "+STR$(
TS))
1450 IF end%<2 PROCline("Oh dear, you s
eem to have died.")
1460 IF end%>2 PROCline("You scored "+
STR$(SC)+" out of "+STR$(TS))
1470 PRINT""Another game (Y/N) ? "
1480 REPEAT G%=GET AND &DF
1490 UNTIL G%=89 OR G%=78
1500 =(G%=89)
1510 :
1520 DEFPROCbefore
1540 ENDPROC
1550 :
1560 DEFPROCleave(R,D$)
1640 ENDPROC
1650 :
1660 REM Actions
1670 :
2500 DEFPROCafter
2550 ENDPROC
2560 :
2570 DEFPROCdirs
2580 N%=-1:REPEAT N%=N%+1
2590 UNTIL A$=LEFT$(D$(N%),1) OR A$=LEF
T$(D$(N%),4) OR N%=6
2600 IF N%=6 ENDPROC
2610 IF FNre(R,N%)=0 PRINT"You can't go
that way":ENDPROC
2620 F%=TRUE
2630 PROCleave(R,LEFT$(D$(N%),1))
2640 IF F% R=FNre(R,N%):IF L=R PROCdesc
(R)
2650 ENDPROC
2660 :
2670 DEFPROCdesc(R)
2680 LOCAL N%

```

```

2690 IFR=0 PRINT"You are carrying";:GOT
02790
2700 PRINT:PROCline("You are "+R$(R))
2710 T%=0:FOR N%=0 TO 5
2720 IF FNre(R,N%) T%=T%+1
2730 NEXT:D$=" You can go "
2740 IFT%=0 ENDPROC
2750 FOR N%=0 TO5:IFFNre(R,N%)=0 GOTO27
80
2760 D$=D$+D$(N%):T%=T%+1
2770 IFT%>1 D$=D$+" " ELSE IF T%=1 D$=D
$+" or "
2780 NEXT:PROCline(D$+" ")
2790 T%=0:FOR N%=1 TO t%
2800 IF FNor(N%)=R T%=T%+1
2810 NEXT
2820 IFR=0 IFT%=0 PRINT" nothing!":ENDP
ROC
2830 IFT%=0 PRINT:ENDPROC
2840 IFR PROCline("You can see")
2850 FOR N%=1 TO t%
2860 IF FNor(N%)<>R GOTO2900
2870 D$=" "+M$(FNom(N%)):T%=T%+1
2880 IFT%>1 D$=D$+" " ELSE IF T%=1 D$=D
$+" and"
2890 PROCline(D$)
2900 NEXT:PRINT
2910 ENDPROC
2920 :
2930 DEFPROCreset
2940 RESTORE
2950 FOR N%=0 TO 5:READ D$(N%):NEXT
2960 FOR N%=1 TO r%:READ R$(N%)
2970 PROCrf(N%,0)
2980 FOR E%=0 TO 5:READ L%
2990 PROCre(N%,E%,L%):NEXT,
3000 FOR N%=1 TO t%:READ R,M,F
3010 PROCor(N%,R):PROCom(N%,M)
3020 PROCof(N%,F):NEXT
3030 FOR N%=1 TO v%
3040 READ W$(N%),W%(N%,0),W%(N%,1)
3050 NEXT
3060 FOR N%=N% TO w%
3070 READ W$(N%),W%(N%,0):NEXT
3080 FOR N%=1 TO m%:READ M$(N%):NEXT
3090 FOR N%=1 TO f%:A%N%=0:NEXT
3100 R=1:SC=0:TS=100
3110 end%=FALSE:L=0:ENDPROC
3120 :
3130 DEFPROCget(N%)
3140 IF FNhere=0 ENDPROC
3150 PROCor(N%,0):PRINT"OK"
3160 ENDPROC
3170 DEFPROCdrop(N%)
3180 IF FNheld=0 ENDPROC
3190 PROCor(N%,R):PRINT"OK"
3200 ENDPROC
3210 :
3220 DEFNavail

```



```

3230 IF FNor(N)<>R AND FNor(N)<>0 PROCe
(2):=0
3240 =TRUE
3250 DEFFNhld
3260 IF FNor(N)<>0 PROCe(3):=0
3270 =TRUE
3280 DEFFNhere
3290 IF FNor(N)<>R PROCe(2):=0
3300 =TRUE
3310 :
3320 DEFPROCinc(F%)
3330 PROCgf(F%,FNgf(F%)+1):ENDPROC
3340 DEFPROCdec(F%)
3350 PROCgf(F%,FNgf(F%)-1):ENDPROC
3360 :
3370 DEFPROCgf(N%,F%)
3380 A%?N%=F%:ENDPROC
3390 DEFPROCOr(N%,R%)
3400 B%?(N%*3-3)=R%:ENDPROC
3410 DEFPROCOr(N%,M%)
3420 B%?(N%*3-2)=M%:ENDPROC
3430 DEFPROCOf(N%,F%)
3440 B%?(N%*3-1)=F%:ENDPROC
3450 DEFPROCrf(N%,F%)
3460 C%?(R%*7+6)=F%:ENDPROC
3470 DEFPROCRe(R%,E%,V%)
3480 C%?(R%*7+E%)=V%:ENDPROC
3490 :
3500 DEFFNgf(N%)=A%?N%
3510 DEFFNor(N%)=B%?(N%*3-3)
3520 DEFFNom(N%)=B%?(N%*3-2)
3530 DEFFNoF(N%)=B%?(N%*3-1)
3540 DEFFNrf(R%)=C%?(R%*7+6)
3550 DEFFNre(R%,E%)=C%?(R%*7+E%)
3560 :
3570 DEFPROCe(E)
3580 M$=E$(E)+"!":GOTO3620
3590 :
3600 DEFPROCm(M)
3610 M$=M$(M)
3620 H%=INSTR(M$,"#")
3630 IFH% M$=LEFT$(M$,H%-1)+X$+MID$(M$,
H%+1):GOTO3620
3640 PROCline(M$):PRINT
3650 ENDPROC
3660 :
3670 DEFPROCline(A$)
3680 REPEAT L$=(?&30A-9)-POS
3690 F$=LEFT$(A$,L%):A$=MID$(A$,L%+1)

```

```

3700 IFRIGHT$(F$,1)=" " OR A$="" GOTO37
50
3710 REPEAT A$=RIGHT$(F$,1)+A$
3720 F$=LEFT$(F$,LEN(F$)-1)
3730 UNTIL RIGHT$(F$,1)=" " OR F$=""
3740 IFF$="" PRINT
3750 PRINT F$;UNTIL A$=""
3760 ENDPROC
3770 :
3780 DEFPROCfile(A$,F$)
3790 IFF$="" OR LEN(F$)>7 PRINT"Use SAV
E/LOAD <filename>":ENDPROC
3800 $&700="*"+A$+" A."+F$+" "+STR$(A%
)
3810 IF A$="SAVE" $&700=$&700+" "+STR$(
D%+2)
3820 ?D%=R:D%?1=SC:X%=0:Y%=&7:CALL&FFF7
3830 IF A$="LOAD" R=?D%:SC=D%?1
3840 PRINT"OK"
3850 ENDPROC
3860 :
3870 DEFFNchk(S%,L%,F$)
3880 N%=S%:REPEAT N%=N%+1
3890 UNTIL LEFT$(W$(N%),4)=F$ OR N%=L%
3900 IF LEFT$(W$(N%),4)=F$ R%=W$(N%,0):W
%=W$(N%,1) ELSE R%=0
3910 =R%
3920 :
3930 DEFPROCinit
3940 w%=v%+n%
3950 DIM D$(6),E$(5),M$(m%),R$(r%),W$(w
%),W%(w%,1),A% f%,B% t%*3,C% r%*7+7,D% 1
3960 E$(1)="You can't # that"
3970 E$(2)="That isn't here"
3980 E$(3)="You're not carrying that"
3990 E$(4)="You've already done that"
4000 E$(5)="You can't do that here"
4010 J%=&700:K%=&6FF:Q$=""
4020 ENDPROC
4030 :
5000 DATA north,south,east,west,up,down
5010 REM ROOMS
5020 REM Descriptions,directions
6000 REM OBJECTS
6010 REM Locations, messages, flags
7000 REM VERBS
7010 REM Verbs,verb numbers,syntax
8000 REM NOUNS
8010 REM MESSAGES

```

MARKS ANALYSIS

```

2280 :
2290 DEFPROCname(S%)
2300 OK=FALSE
2310 IF LEN(name$)>7 THEN VDU7:PRINT"Th
he filename should contain no more thans
even characters ... ":PROCwait:ENDPROC
2320 A=OPENIN(name$):CLOSE#A
2330 IF S%=1 AND A<>0 THEN VDU7:PRINT"
File already exists .. ":PRINT"Overwrite
(Y/N)";IF NOT FNok ENDPROC

```

```

2340 IF S%=2 AND A=0 THEN VDU7:PRINT"Fi
le does not exist .. ":PROCwait:ENDPROC
2350 OK=TRUE
2360 ENDPROC
2370 :
2380 DEFPROCOrder
2390 pos=pos+old:old=D%(X%)
2400 ENDPROC
2410 :
2420 DEFFNok=(GET AND &DF)=89

```


HIGH RESOLUTION GRAPHICS DUMP

Design Dynamics, publishers of the popular 3D Design package, have produced a very high resolution printer dump. Graphic designer Roger Burg has been looking at this with interest.

Product : Mode-00 Dump
Supplier : Design Dynamics
8 Meadow Way, Ampthill,
Bedford MK45 2QX.
Tel. (0525) 402447
Price : £12.95 (disc only)

Design Dynamics has already produced and enhanced its Interactive 3D (a powerful 3D graphics design program reviewed in previous issues of BEEBUG - see Vol.4 No.10 & Vol.5 No.8). Mode-00 Dump arose from the need to represent wire frame images produced by the 3D package to a higher resolution than is possible on the screen.

The program takes a screen file and reproduces it on a dot-matrix printer, but Mode-00 Dump does not dump the screen, it creates a new and higher resolution image than the original. It uses the screen graphics output from other programs or commercial graphics packages which has been captured in a *SPOOL file. In fact, it will take spooled output from anything which uses standard graphics commands (e.g. MOVE, DRAW, PLOT); it cannot cope with screen 'dumps'.

The program processes the contents of this spooled file, to build an image which is sent to the printer with a resolution of 640 by 512 pixels, twice that of screen mode 0. It relies on the fact that the screen is addressed as a 1279 by 1023 grid, although in mode 0, it only provides 640 by 256 pixels.

Although mode 0 provides the highest resolution of which the BBC micro is normally capable, its pixels are vertical rectangles. Thus lines which are nearly horizontal appear stepped and jagged, and

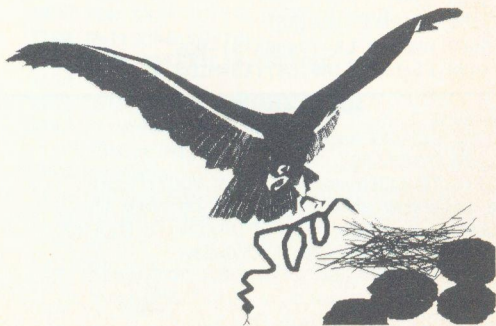
this reduces the impression of resolution. The effect of splitting each oblong pixel into two equal squares is dramatic, and Mode-00 Dump takes advantage of it. It displays significantly more detail than a mode 0 screen, and independently halves the height of its text to blend with the higher resolution graphics.

A simple but adequate manual is provided, together with a function key strip. The disc also contains graph drawing and an ellipse generator programs, both of which demonstrate the advantage of printing their output with Mode-00 Dump. The ellipse generator produces spirograph-like patterns and deserves attention. While this is not original, it is an absorbing toy, and produces high resolution printouts which make most attractive images in their own right.

The program has considerable potential for reproducing diagrams, graphs and charts, and presents new possibilities for computer art. It is no slower than many screen dumps, and I found the processing time quite acceptable in view of the quality of the result.

There are a few minor limitations on the graphic commands which are supported, notably in the printing of text. The results are limited by the size of file which a disc can hold, though the length of a spooled file is usually a fraction of the 10 or 20k screen which it creates.

The Mode-00 Dump is compatible with most dot-matrix printers, and runs on the B, B+ and Master. It is available in various disc formats at £12.95, or £20.00 for Interactive 3D with Mode-00 Dump. **B**



1st course

Plotting for Effect

Mike Williams explains how to make the most of the versatile PLOT com- mand in your graphics displays.

Many newcomers to programming are often attracted to try out the graphics capabilities of their micro, and the BBC micro (and even more so the Master and Compact) are well endowed in this respect. The basic graphics commands are

MOVE and DRAW, but in fact these do no more than just scratch the surface of this subject. MOVE and DRAW are two special examples of the whole family of PLOT commands, and an understanding of these will do much to help you improve your graphics programming on the Beeb.

THE PLOT COMMAND

The PLOT command, available in BBC Basic, is a general graphics command having many different uses. The format of the command is always the same:

PLOT k,x,y

The value of k, which can nominally be any value in the range 0 to 255, determines the exact function carried out by the PLOT command. In fact, MOVE x,y is exactly equivalent to PLOT 4,x,y and DRAW x,y to PLOT 5,x,y. On a model B, the values of k can range from 0 to 87, but on a Master or Compact from 0 to 207 as many additional functions are possible. The values of x and y refer to a point on the graphics area of the screen (1280 points horizontally, 1024 points vertically), and are the same regardless of mode. In the examples, the values of x and y are always absolute screen co-ordinates relative to the origin, though other variations on the PLOT command allow co-ordinates relative to the last position to be used.

FILLING TRIANGLES AND RECTANGLES

The first, and one of the most useful applications of the PLOT command, is in producing filled shapes on the screen. The

command: PLOT 85,X,Y will plot and fill a triangle on the screen using the drawing colour currently set by the GCOL command (or white by default). The

three vertices (or corners) of the triangle are taken to be the last two points previously specified together with the point x,y given in the PLOT command itself. For example, try the following short program:

```
100 MODE 2
110 GCOL 0,3
120 MOVE 200,100
130 MOVE 640,800
140 PLOT 85,1080,100
150 END
```

The program uses mode 2, and selects yellow (colour 3) as the drawing colour. Using two MOVE commands and one PLOT 85 command, a yellow filled triangle is then drawn using (200,100), (640,800) and (1080,100) as the three vertices.

Master and Compact

Users of these two machines have access to a much wider range of PLOT commands than model B users. Many of the additional variations provide more direct and efficient ways of achieving effects than are possible on the model B. Where appropriate, reference will be made to these alternatives in the text. All the routines as listed will work perfectly well, though, on these machines without any changes.

You will notice that the two sloping sides of the triangle have a stepped appearance. This appearance depends on the resolution being used (low resolution in mode 2). If you try the same program using mode 1 (medium resolution) or mode 0 (high resolution) you will be able to see the effect that this has. There is also a trade-off in that the higher the resolution, the fewer the number of colours available (16 in mode 2, 4 in mode 1, but only 2 in mode 0).

A filled triangle is the only shape that is immediately available to us (on a model B), and so any other filled shape is usually constructed from triangles. Consider, for example the task of drawing a filled rectangle.

Suppose we want a green rectangle, 400 units high by 600 long and with the bottom left hand corner in position (200, 100). It is worth just trying to work out the sequence of commands for yourself, and to see how few you really do need. A rectangle is, of course, made up from two triangles and the shortest way of achieving this is as follows:

```
100 MODE 2
110 GCOL 0,1
120 MOVE 200,100: MOVE 200,500
130 PLOT 85,800,100:PLOT 85,800,500
140 END
```

Notice how, by moving from one vertex to the next in the right order, we can draw our rectangle with just four commands. Basic always remembers the co-ordinates of the last two points visited. On a Master or Compact the same result could be achieved by writing:

```
100 MODE 2
110 GCOL 0,1
120 MOVE 200,100:PLOT 101,800,500
130 END
```

On the Master series, PLOT101 produces a filled rectangle using the last two points specified as the co-ordinates of two opposite corners.

In many graphics programs it is often convenient to parcel up various graphics functions as procedures. We can rewrite either of the routines above to produce a procedure which will display a square of any size, and in any colour, anywhere on the screen. The procedure, PROCfillsquare, may be defined as follows:

```
1000 DEF PROCfillsquare(colour,size,x,y)
1010 GCOL 0,colour
1020 MOVE x,y: MOVE x,y+size
1030 PLOT 85,x+size,y
1040 PLOT 85,x+size,y+size
1050 ENDPROC
```

Again, Master and Compact users can, if they wish, replace lines 1020 to 1040 by the one line:

```
1020 MOVE x,y: PLOT 101,x+size,y+size
```

By including the procedure in a loop as shown below, we can produce a random display of coloured squares on the screen:

```
100 MODE 2
110 REPEAT
120 colour=RND(7):size=RND(1000)
130 x=RND(1279):y=RND(1023)
140 PROCfillsquare(colour,size,x,y)
150 UNTIL FALSE
160 END
```

We can use a similar procedure to draw rectangles, or any four sided filled shape, and indeed shapes of more than four sides. Master and Compact users will find that they have alternative PLOT commands for directly drawing filled rectangles, parallelograms, circles and ellipses.

ALTERNATIVE TECHNIQUE

There is a quite different technique that can be used to produce coloured rectangles. This uses the VDU24 command to define a graphics window on the screen. The required colour is then set as the background colour by using GCOL0,128+C where C is the colour in the range 0-15 (for mode 2 at least). Simply executing the CLG command will then 'paint' the graphics window in the colour specified. The following short routine will draw the same yellow-coloured rectangle as before:

```
100 MODE 2
110 GCOL 0,129
120 VDU 24,200;100;800;500;
130 CLG
140 END
```

This technique is worth knowing, as it is both short and fast, but it is more limited as only rectangles (or squares) may be drawn, and the rectangle must be wholly within the normal screen area, or the VDU24 command has no effect. You may also need to cancel the VDU24 command after use with VDU26 to clear any windows.

FILLED CIRCLES

We will now look at how we can use the triangle filling facility to generate filled circles. Once we realise that a circle is just a polygon with a large number of sides, we can readily see our way to writing a routine to produce a filled circle, made up of many triangles, all touching at the centre. This time, the routine has been written directly as a procedure to display a filled circle in any colour and with any radius, with its centre in any screen position (x,y).


```

1100 DEF PROCfillcircle(colour,radius,
    x,y)
1110 LOCAL angle,X,Y
1120 VDU29,x,y;GCOL 0,colour
1130 MOVE radius,0
1140 FOR angle = 0 TO 2*PI STEP PI/16
1150 X=radius*COS(angle)
1160 Y=radius*SIN(angle)
1170 MOVE 0,0:PLOT 85,X,Y
1180 NEXT angle
1190 ENDPROC

```

Notice the use of the VDU29 command to move the origin to the centre of the circle - this nearly always makes circle drawing much easier. Each time the program moves round the circumference from one point to the next, it fills in a triangle from those two points to the centre of the circle. The order of visiting the three points of the triangle is important in producing an efficient routine.

Master and Compact users have the advantage of a PLOT command that completely replaces the FOR-NEXT loop in the program above. This is PLOT157. Just replace lines 1130 to 1180 by:

```

1130 MOVE 0,0
1140 PLOT 157,radius,0

```

Here, we move to the centre of the circle, and then plot the circle, specifying any point on its circumference.

The previous program to draw various sized squares randomly on the screen in different colours could be very easily adapted to do the same job for circles (this is included on the magazine cassette/disc for this month). This last program also shows very clearly that, just as with DRAW and MOVE, no error occurs if the shape being drawn goes off the visible screen area.

3D SHAPES

The final program for this month shows how the three commands of MOVE, DRAW and PLOT can be combined in a simple procedure that gives an appearance of depth to the screen display. Most of the work in this program is done by a procedure called PROCcube. The purpose of this procedure is to display a cube on the screen with a specified colour, size and position. This is really just an extension of our earlier filled square procedure.

A cube, as displayed on the screen, consists of a square (just as before)

together with two parallelograms, one adjacent to the top edge and one adjacent to the right-hand edge of the square. The furthest visible corner of the cube is chosen to be 0.6 of the size of the cube in distance from these two edges. This is quite arbitrary, the value really depending on the effect you want to achieve. This is not a true perspective, as opposite sides are parallel, not converging to a vanishing point, but at least it keeps the maths quite simple.

In order to emphasize the 3D effect, the three edges of the cube separating the visible faces are drawn over the top of the coloured faces using a colour logically one more than that used for the cube as a whole. Master and Compact users may like to incorporate the rectangle and parallelogram drawing functions available to them (PLOT101 and PLOT117) to produce simpler code, though, of course, the program as listed will run on all systems.

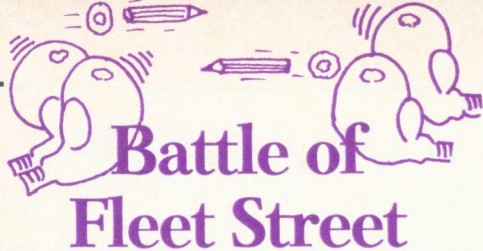
As with the other programs, this one repeatedly calls the procedure to place cubes randomly on the screen. Nevertheless the results are quite effective.

```

100 MODE 2
110 REPEAT
120 x=RND(1024):y=RND(1024)
130 size=RND(500):colour=RND(7)
140 PROCcube(colour,size,x,y)
160 UNTIL FALSE
170 END
180 :
1200 DEF PROCcube(colour,size,x,y)
1210 GCOL 0,colour
1220 MOVE x,y:MOVE x,y+size
1230 PLOT 85,x+size,y:PLOT 85,x+size,
    y+size
1240 PLOT 85,x+1.6*size,y+0.6*size
1250 PLOT 85,x+1.6*size,y+1.6*size
1260 MOVE x+size,y+size
1270 PLOT 85,x+0.6*size,y+1.6*size
1280 PLOT 85,x,y+size
1290 GCOL 0,(colour+1)MOD8
1300 DRAW x+size,y+size:DRAW x+size,y
1310 MOVE x+size,y+size
1320 DRAW x+1.6*size,y+1.6*size
1330 ENDPROC

```

Next month we will look at ways of filling more irregularly shaped areas on the screen, together with some of the other applications of the PLOT command.



Battle of Fleet Street

If you're interested in desk top publishing on the Beeb then you must have looked at Fleet Street Editor from Mirrorsoft and Pagemaker from AMS. Simon Williams examines the latest enhancements to these packages as the circulation battle heats up.

Product : Fleet Street Admin Xtra
Walt Disney Graphics Library
Supplier : Mirrorsoft Ltd,
 Maxwell House,
 74 Worship Street,
 London EC2A 2EN.
 Tel. 01-377-4837
Price : £14.95 each inc. discs and
 manual supplement.

Mirrorsoft's first venture into desktop publishing on the BBC micro, the highly successful Fleet Street Editor, is a very capable program, but as with all good things there are always improvements that can be made.

Two new packages which help the budding magnate to put together a smart paper are Admin Xtra and the Walt Disney graphics library. Both are supplied as a flippable 40 or 80 track disc in an A5 folder, also housing appendix sheets pre-punched for filing in the main Fleet Street binder. The documentation is up to Mirrorsoft's usual high standard, with plenty of illustrations.

Admin Xtra, programmed by those doyens of useful utilities, Clares, provides four main utilities: mode converter, display, poster maker and disc index. They can be used independently of the main FSE disc, and are called from their own menu, using Space and Return as normal.

The mode converter takes screens from modes 0,1,2,5 and 7 and converts them into mode 4 screens, suitable for incorporating into Fleet Street panels. You can now take graphics screens from many art packages



(including, cheekily, the AMX range), convert them to mode 4 and use them in your newsletter or poster.

Display is a utility which should really have been provided with the original package. It allows you to scan through panels that you've made up and stored on disc. Handy for copying panels from disc to disc, or checking their order before making up a page.

Poster Maker is a useful utility which broadens the application of FSE. The program allows you to print any panel in a number of large scale formats, including 2x2, 3x3 and 2x4. These options open the possibility of handouts and flyers, or of producing colouring books for your children! It's a pity the options are only available for a single panel at a time, but it does allow you to produce A4 landscape proportions.

The disc index can be used on any set of discs, not just those used for FSE files. It automatically reads all the file names on a disc, together with 'descriptors' of up to 30 characters each, and stores them on a master index disc. It can then search the index disc for a word or phrase within a file name or descriptor, and will display a disc number so that you can go straight to your properly catalogued (they are, aren't they?) disc of that number. This utility is only as good as is your own organisation.

The utility which still seems to be missing, and why I still favour Pagemaker

to FSE, is the ability to view a full page before printing. I don't expect to be able to see it at full size, but it's very difficult to gauge page layout by looking at individual panels or a page make-up screen consisting of file names.

The Walt Disney graphics library contains 22 pages of Mickey, Donald, Goofy and the rest. Most of the key characters are there, though in most cases only one sketch of each. They're all outline drawings, and very close indeed to the originals.

My only hesitation is that the images are so well known that it's difficult to use them in anything other than a Walt Disney-based application. If that's what you want, then this library has a good cross section of favourites.

Product : AMX Extra! Extra!
Supplier : AMS Ltd,
166/170 Wilderspool Causeway,
Warrington WA4 6QA.
Price : £24.95 inc two discs and manual.

If Fleet Street Editor deserves extra bits and pieces, then so also does its main rival, the AMX Pagemaker. The package breaks down into utilities on the one hand and fonts and clip-art on the other. Unlike FSE, Pagemaker wasn't originally supplied with a graphics library, and it's taken AMX a while to get one out.

The utilities are run from a main menu, and are selected with the mouse. They are: icon stripper, font converter, page copier and customiser.

The icon stripper removes the border icons from saved Pagemaker screens, so you can use them as title screens, or dump them cleanly to paper.

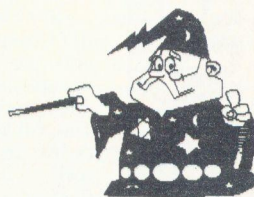
The font converter allows you to use the fonts from AMX Super Art by converting them to Pagemaker's mode 0 format. This includes the icon files, which make useful clip-art in their own right. The combination is great for producing technical sheets with circuit diagrams in them.

The page copier is a neatly designed utility for copying pages directly from one page format disc to another - again a function which should have been provided with the original program.

The customiser is probably the most useful utility provided. It allows you to select screen colour and position, and save the configuration to disc. You can also select a customised print dump. Unlike the original version of the printer utility, the dumps in Extra Extra aren't all the same! A Canon laser printer dump is even included in the list for the wealthy.

A SAMPLE OF SOLO A SLICE OF SAUSAGE

A book of HiTeo 7B which is a very readable font.
 A clump of COWBOY, looking good.



The clip art provided on the second disc in the pack is compressed to fit. Once expanded again, there are a lot of useful images categorised under headings such as sport, music, buildings and fancy borders. There is a good selection of cartoons, and all material is original, so it won't be immediately recognised by your readers.

28 new fonts are provided, which complement the 16 originally included. There are plenty suitable for display as well as new 70, 80 and 90 column fonts.

All in all the package is very welcome, and should further the reputations of its authors, Alex Blok and Neil Lee, as well as that of AMX.

Surac explains what Turtle graphics is all about, and provides some routines to implement your own Turtle system.

Those of you who have played with languages such as Logo will be familiar with the idea of the "Turtle". For those who haven't, it's a different approach to drawing graphics from the usual sort on the Beeb, and most other computers.

Usually, graphics displays use the screen as a sheet of squared paper, and lines are drawn by saying "Go from this point to this point". The Beeb also has a relative graphics system (e.g. PLOT 1,X,Y) which effectively says "Go up so far, and sideways so far".

Turtle graphics is different and uses the concept of lines being drawn by a turtle crawling around the screen. The turtle can be made to point in any direction, and go forward any distance in the direction in which it is pointing. Put it in the middle of the screen facing upwards, turn it right 45 degrees, and go forward 500 units. The result is a diagonal line from the centre of the screen to near the top right-hand corner; the turtle is now at the end of the line, still pointing at 45 degrees to top-right.

If you like, it's a graphical system using relative polar co-ordinates. The approach has two big advantages. Once you have worked out a routine to draw the shape you want, it's easy to put that shape at any angle where you want it on the screen, simply by

putting the turtle in the correct starting position. Also, once the shape is right, it's easy to draw the same pattern in any different size, just by putting a suitable scaling factor in the drawing routine. Put the two together, and you have the basis of a powerful system.

BBC TURTLE ROUTINES

That's all very well; but how do you do it? Here are a set of procedures to give the fundamental turtle tools:

```
10000 DEF PROCTMove(dist)
10010 GCOL 0,TColour%
10020 MOVE TXPos,TYPoS
10030 TXPos=TXPos+dist*SIN(RAD(TAng))
10040 TYPoS=TYPoS+dist*COS(RAD(TAng))
10050 PLOT 4-TPenDown%,TXPos,TYPoS
10060 ENDPROC
10990 :
11000 DEF PROCTTurn(angle)
11010 TAng=FNNorm(TAng+angle)
11020 ENDPROC
11990 :
12000 DEF PROCTInit(col%)
12010 TPenDown%=TRUE
12020 TColour%=col%
12030 PROCTPos(640,512,0)
12040 ENDPROC
12990 :
13000 DEF PROCTPos(x,y,a)
13010 TXPos=x
13020 TYPoS=y
13030 TAng=FNNorm(a)
13040 ENDPROC
13990 :
14000 DEF FNNorm(angle)
14010 angle=angle MOD 360
14020 IF angle>180
      THEN angle=angle-360
14030 IF angle<-180
      THEN angle=angle+360
14040 =angle
```

This group of routines uses a shared set of variables to keep track of what's going on. "TXPos" and "TYPoS" track the position of the turtle in the Beeb's usual absolute screen co-ordinates. "TAng" holds the angle, in degrees, it's pointing in; straight up is zero and positive angles turn it to the right. TColour% holds the colour of the turtle's "ink", using the normal BBC colour codes. Finally, TPenDown% is a logical variable which, when TRUE, makes the turtle draw a line;

set it to FALSE and the beast moves without drawing anything. Note: this is not the same thing as setting the colour to black (why not?).

PROCTMove moves the turtle forward "dist" graphics units. A negative value moves it back. Note that lines 10010 and 10020 re-position the turtle and set it to its colour - this lets you carry out conventional graphics work between turtle actions. Lines 10030 and 10040 work out the new position and line 10050 actually draws the line; it uses the fact that a TRUE variable has the value "-1" to control whether or not lines are drawn.

PROCTurn turns the beast "angle" degrees to the right. It uses FNNorm to actually keep the value of the angle between -180 and +180 so that the sine and cosine calculations work sensibly. Negative angles turn left.

There are also two other useful procedures. PROCTInit puts the turtle in the middle of the screen, pointing up, and ready to draw in colour "col%". PROCTPos positions the turtle at (x,y), pointing at angle "ang".

USING TURTLES

Of course, those routines are only the starting point but, to play with them a bit, lead in with this simple code:

```
100 MODE4:PROCTInit(1)
110 *KEY0 CO.0|MC0.7:
    INPUT"Colour:"C%:PROCTInit(C%)|M
120 *KEY1 CO.0|MC0.7:
    INPUT"Distance:"A%:PROCTMove(A%)|M
130 *KEY2 CO.0|MC0.7:
    INPUT"Angle:"B%:PROCTTurn(B%)|M
140 VDU 24,0;300;1279;1023;
150 VDU 28,0,31,39,25
160 MOVE 0,304:DRAW 1280,304
170 END
```

It splits the screen into separate graphics and text screens and sets up function keys 0, 1 and 2 to call up the turtle commands. Use the keys to call the procedures, inputting the right values for the parameters when prompted, and moving the turtle around the screen.

That's not really a lot of use! However, once the basic turtle routines are available, they can be used to draw more complex shapes. For example, here's a

procedure to draw a regular polygon with "nsides" sides, each of length "length":

```
5000 DEF PROCPolygon(nsides,length)
5010 LOCAL I%,turn
5020 turn=360/nsides
5030 FOR I%=1 TO nsides
5040   PROCTMove(length)
5050   PROCTurn(turn)
5060 NEXT
5070 ENDPROC
```

It works out (line 5020) just how much the turtle must turn at each corner of the polygon and then goes ahead and draws it. Note that it draws it from the turtle's starting point and position; so, to draw a diamond in the upper left quadrant of the screen:

```
400 PROCTPos(100,800,45)
410 PROCPolygon(4,200)
```

The procedure always leaves the turtle where it started. Remember, that once a basic pattern can be drawn by a set of turtle routines, it can be repeated anywhere, any angle, any size. To draw a shell-like pattern on the screen:

```
100 MODE 4
110 PROCPattern
120 REPEAT UNTIL INKEY=99
500 END
990 :
1000 DEF PROCPattern
1010 PROCTInit(1)
1020 CLG
1030 L%=20
1040 R%=20
1050 FOR I%=1 TO R%
1060   PROCPolygon(24,L%)
1070   PROCTurn(360/R%)
1080   L%=L%+2
1090 NEXT
1100 ENDPROC
```

It runs remarkably fast but it does use a lot of trig calculations. However, if you combine it with the look-up table approach (see the Workshop in BEEBUG Vol.5 No.4), the results are spectacular.

Remember, turtle graphics can make it very easy to build on basic routines to get many different functions. For instance, we can use PROCPolygon above to draw circles, any size, anywhere on the screen in the next procedure.

NEW BOOKS for VIEWERS

Until recently, users of Acorn's View family have had to rely solely on the Acorn manuals for help and assistance. With the publication of these two books all this has changed, as Mike Williams reports.

I have long believed that View is by far the best word processor for the Beeb (see BEEBUG Vol.2 No.2), but I have often felt myself to be in the minority. Now that Acorn have sold over a 100,000 Masters and Compacts, each with a copy of View included, I no longer feel so alone, and the arrival of two brand new books to support the View family is added comfort. Despite popular opinion to the contrary, I do not believe that View is difficult to learn, but in common with most of the better things in life it does demand more effort from the would-be user. Even so, a little help can go a long way, and both the books reviewed here will do that and a whole lot more.

Mastering View, ViewSheet and ViewStore by Clive Williamson, published by Sigma Press at £12.95.

Clive Williamson's book is devoted not just to View, but also covers ViewSheet and ViewStore (the spreadsheet and database applications in the View family). Perhaps because of this, the book starts with a wealth of general information - setting up a system, differences between the B, B+, Master and Compact, choice of filing systems, etc. I am sure that much of this is useful, but some 30 pages have passed before we hear much of View itself.

The initial introduction to editing text in View I found to be excellent, with some very clear diagrams showing the range

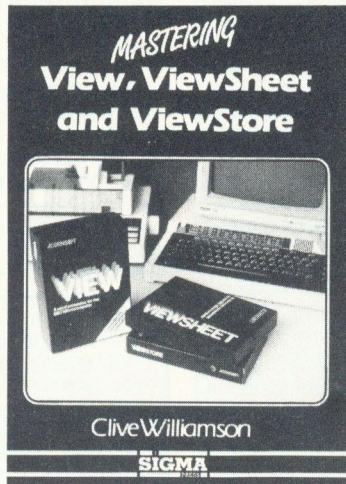
of cursor movements possible. The simpler function key operations are also well described. There IS one major hiccup - the illustration of the View function key strip has disappeared from page 37 leaving just the function keys themselves - not too disastrous though; all View users will have their own function key strips anyway.

The book moves on to the loading, saving and printing of View files, followed by the use of rulers, a fundamental aspect of View, and stored commands. However, two further chapters entitled 'Common View Problems' and 'Advanced Techniques' fail to provide more than cursory descriptions of commonplace and important editing operations. Markers get very little attention, and the whole business of moving, copying or deleting complete blocks of text is poorly covered if at all. Page 83 even includes a reference to a non-existent View command, COPY!

The remainder of the book provides a similar introduction to ViewSheet and ViewStore, but at about 50 pages each, no more than that. This is more than adequate for the beginner to these applications, but the more experienced user will find little of which he/she is not already aware. There are many illustrations that could have added much to the text, but the frequent poor quality of reproduction renders many (page 169 for example) less than useful. The attempt to reproduce the screen appearance of white on black is far less successful than the less accurately representative black on white of Bruce Smith's book.

View: A Dabhand Guide by Bruce Smith, published by DABS Press at £12.95

Bruce Smith has produced a book that is devoted entirely to View, and as such has much more to offer the already competent View user, as well as the comparative novice. The style is generally 'chatty' and Bruce Smith's now considerable experience as writer and author (much of it using View) is often apparent, and adds convincing support to what he says.



Even so, I felt that some of the introductory material, covering simple editing using the cursor keys, was not as clearly presented as many beginners to View might wish. You have to veritably 'mine' the relevant information out of some copious text. Once into his stride however, Bruce Smith becomes much more readable. Rulers, markers, formatting and all the rest of View is described clearly and with many clear well chosen examples. In fact, I thought the way in which example View screens are presented a particularly strong feature of this book.

One of the other attractions here is the inclusion of listings for several utility programs. As might be expected there is a Printer Driver Manager, Extended Disc Catalogue, and five utility programs. The listings are, however, very wasteful of page space, and 'appear' all to be wrongly titled. As might be expected, all these programs and some others are available on disc at £7.95 (5.25" DFS), £9.95 (3.5" ADFS).

CONCLUSIONS

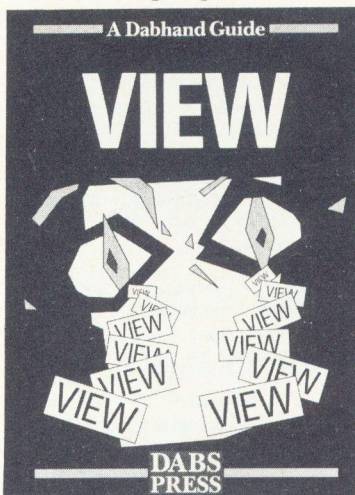
It is always interesting to undertake a comparative review, and this occasion is no exception. Clive Williamson's book provides a most useful introduction to the

three main components of the View family, and his initial coverage of editing text with View will find favour with many who have perhaps 'struggled' with View in the past. However, several important features of View are poorly described, and as already mentioned, the illustrations (of screens) let the book down badly.

Bruce Smith's book concentrates solely on View, and as a result has much more to say on that subject, most of it being practical and down-to-earth in style. The early pages are not as clear as the rest, but the thorough coverage, additional programs, and reference appendices will allow the more demanding user to make the most of View.

In summary, anyone just starting with View, and those who want to know about ViewSheet and ViewStore too, will probably prefer Williamson, at least to start with, while for those who want a complete, thorough and readable guide to View then Bruce Smith is your man.

And if you wonder whether, after more than four years of using View, either book was able to teach me anything new about View I can only say both did, but professional pride prevents me from revealing what!



WORKSHOP

```
6000 DEF PROCTCirc(x,y,r)
6010 PROCTPos(x,y+r,90)
6020 PROCPolygon(24,PI*r/12)
6030 ENDPROC
```

That puts a circle (actually a 24-sided polygon, but that's close enough), radius "r" graphics units, on the screen with its centre at (x,y). How would you draw a circle at the current turtle position?

The routines I've outlined here give a totally different way of producing graphics from the Beeb's standard facilities. Each method has its own advantages and disadvantages. Turtles, though, are particularly useful when a complex pattern is being repeated in several places on the screen, especially if the size or angle has to alter as well.

MARKS ANALYSIS

```
2430 :
2440 DEFFNpercent:Z%=Z%+C%(X%)
2450 =INT(((Z%*100)/N%)+.5)
2460 :
2470 DEFFNstandard=INT((newmean+(X%-me
an)*newdev)/dev)+.5)
2480 :
```

```
2490 IF ERR=17 THEN CLOSE#0:GOTO170
2500 ONERROR:OFF:VDU15:*FX3,0
2510 CLOSE#0:IF ERR<6 THEN VDU26:CLS:R
EPORT:PRINT ERL:END
2520 VDU7:PRINT"The file you have trie
d to open is not adata file . . . .":PRO
Cwait:GOTO 170
```




POSTBAG



POSTBAG

Improved ROM Controller

I have frequently used the ROM Controller program (published in BEEBUG Vol.5 No.3), but I have noticed that sometimes the program saves the last 50 bytes or so of a full 16K ROM incorrectly. Lines 900 and 950 seem to be the culprits. The ROM is copied to RAM starting at &3C00. Consequently a 16K ROM will finish at &7BFF, one byte below the mode 7 value of HIMEM. However, this ignores the fact that the Basic stack begins at &7BFF, so any access to the stack after copying a 16K ROM will corrupt the last few bytes of the ROM image.

The solution is simple: just copy the ROM into RAM at &3A00 onwards. This is achieved by replacing &3C by &3A in line 900, and 3C00 by 3A00 in line 950.

M.R.Chambers

This is a most useful piece of information which has already helped some members who had encountered this problem.

Helping Beginners

I purchased a BBC Master two months ago. A complete beginner, I have spent many hours with the Welcome disc and manual.

I have a Utility Disc with the disc drive that appears not to work on the Master, a ViewStore ROM which is beyond my capabilities to adapt to the ADFS, a Printer Generator disc which only works on DFS, a Panasonic Printer manual which is incomprehensible, as are

the instructions with the Dumpmaster ROM (What is DUMPING?). As for the ADFS, I am beginning to think I shall need a degree to understand that! No doubt all these problems can be solved with time and patience. I have plenty of the latter, but little of the former.

May I suggest a short training session for which I am sure most people would be prepared to pay.

M.Costa

Those of us who have built up several years experience of using computers tend easily to forget how daunting the whole subject can be to newcomers. In the first instance, we would suggest that a beginner to computing takes time in discussing their needs with a good shop or dealer - our own showroom and staff are ideal for this. ViewStore cited above is reckoned to be difficult to use even by experienced computer users, as is the ADFS.

Secondly, make use of your supplier in getting any extra help you need. In addition, BEEBUG has a group of people set up to answer all manner of technical questions from beginners upwards. You may also find it worthwhile to join a local User Group whose members will be able to answer many of your questions for you - we have a list of such groups known to us - or try your local library for information.

The idea of running courses is an interesting

one and we shall be giving this some serious thought.

Puzzled

The American Cryptogram Association is publishing a Computer Supplement for people who would like to apply their computers to the solution of ciphers. We believe that there will be some interest amongst your readers in this publication which appears twice a year. The supplement costs \$2 per issue, and a subscription to the ACA costs \$18.75 from the UK. For further information contact the ACA Treasurer at 12317 Dalewood Drive, Wheaton, Maryland, 20902 USA.

Mike Barlow

Computer Supplement Editor

A Highland Welcome

May I draw your attention to the Grampian Amateur Computer Society. Formed around ten years ago, our members come from all walks of life, and with a variety of machines and interests. We meet regularly on Monday evenings, and there is a bi-monthly newsletter. We would like to make our presence known to BEEBUG readers in the North East of Scotland, or indeed any readers who might be visiting the area and would like to drop in. Anyone interested may phone myself (0467 24030) or Bruce Edelsten (0224 639911); they will be made very welcome.

Paul Cuthbertson

This obviously thriving group is based in Aberdeen, and we are sure any new members or visitors will be made most welcome.

HINTS HINTS HINTS HINTS HINTS

and tips

and tips

and tips

and tips

and tips

Flashy REM Statements

It is useful to be able to highlight certain areas in a program listing. A flashing REMark in mode 7 is achieved by following the REM with a quote followed by Shift-f8 (denoted by <Sh-f8> below). Other effects can be achieved and are conveniently listed in the BEEBUG Giant Reference Card. As an example try:

```
850 REM"<Sh-f8> Flashy eh!
```

Clive Perrott

Printed Output Only

The User Guide seems vague on how to obtain output on the printer only and not the screen. The Advanced User Guide is more enlightening and shows that the solution is *FX3,10. Interestingly, this sends output to the printer whether or not VDU2 has been issued.

Mike Williams

Return Battering

The Return key takes quite a battering during many arcade games, and this short routine will transfer its burden to the much less utilised Shift key. If the game already uses the Shift key for another purpose, this routine is not appropriate:

```
10 osbyte=!&20A AND &FFFF
20 FOR I%=0 TO 3 STEP 3
25 p%=&900
30 [OPT I%
40 CMP #&81:BNE not
45 CPY #&FF:BNE not
50 CPX #(256-74):BNE not
```

```
60 LDX #(256-1)
70 .not JMP osbyte:]
80 NEXT I%
90 !&20A=!&20A AND
&FFFF0000 OR &900
```

This routine diverts OSBYTE call 129 and replaces a test for the Return key with a test for the Shift key.

David James

Watford's Diagnostic Disc & Shadow RAM Board

With Watford's shadow RAM board nothing happens on booting up their Diagnostic disc. The solution is to issue the *RAMOFF command followed by Ctrl-Break. PAGE then changes to &1900 and the diagnostic disc now works perfectly.

D. M. Wells

Editing Basic with Wordwise

Using Wordwise to edit Basic or Assembler listings can be very useful. However, entering new programs from scratch requires a file of line numbers. A Wordwise file can be created from Basic in two ways:

Method 1:
>*SPOOL W.auto
>AUTO
Now press Return up to the required number of lines
>press Escape
>*SPOOL

Method 2:
10 C=OPENOUT"W.auto"
20 FOR L=10 TO 100 STEP 10
30 s\$=STR\$L
40 FOR i=1 TO LENS\$

```
50 BPUT#C,ASC(MID$(s$,i,1))
60 NEXT:BPUT#C,13
70 NEXT:CLOSE#C
```

Substitute your own file name in line 10 and alter the values in line 20 according to your line number requirements.

John C. Thomas

Ibico LTR-1 & Pound Signs

I have found a simple way to print blank lines and pound signs on my Ibico LTR-1 printer.

VIEW - pound sign:

Use the curly left bracket on the key to the left of the pound key on the keyboard.

VIEW - Blank line:

<Return><Highlightl>
<Space><Highlightl><Return>

WORDWISE - Blank line:

<Return><Space><Return>

WORDWISE - pound sign:

Start text with:
<green> DP123 <white>
now use the keyboard pound sign as required.

M. J. Ansell

Printing with Tilde

When a PRINT statement starts with a tilde, and commas between items are omitted, the tilde applies to all the following items, compare the output between:

```
PRINT ~100,200,300
giving 64 200 300
```

```
and: PRINT ~100 200 300
giving 64 C8 12C
```

C. J. Collins

BEEBUG MEMBERSHIP

Send applications for membership, membership renewals, membership queries and orders for back issues to the address

shown. All membership fees, including overseas, should be in pounds sterling drawn (for cheques) on a UK bank.

MEMBERSHIP SUBSCRIPTION RATES

£6.90 - 6 months (5 issues) UK ONLY

£12.90 - 1 year (10 issues) UK, B.F.P.O., Channel Islands

£19.00 - Rest of Europe

£26.00 - Americas & Africa

£23.50 - Middle East

£28.00 - Elsewhere

BACK ISSUES (Members only)

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

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

Please add the cost of post and packing as shown:

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

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

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

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

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

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

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

BEEBUG MAGAZINE is produced by BEEBUG Ltd.

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

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

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

BEEBUG Ltd © 1987.

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £40 per page, but please give us warning of anything substantial that you intend to write. A leaflet, 'Notes of Guidance for Contributors', is available on receipt of an A5 (or larger) SAE.

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

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

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

Magazine Cassette/Disc

CONTENTS CASSETTE/DISC JUNE 1987

THE WORLD BY NIGHT AND DAY — see a fascinating view of the earth with this version of the program designed for both BBC micro and Master series.

EXPLORING ASSEMBLER — a first example in our new series for beginners to machine code programming.

THE MASTER SERIES

WORK DISC GENERATOR — complete EXEC file for you to adapt and use to create your own work discs.

AUTO-BOOTING ADFS MENUS — a selection of menu files to pick and mix as you wish.

HOW TO WRITE AN ADVENTURE GAME — part one of this program provides a complete 'skeleton' for writing your own adventure games.

MARKS ANALYSIS FOR TEACHERS — take the hard work out of exam administration with this program for analysing and standardising marks.

BEEGUB WORKSHOP — two complete demonstration programs containing the procedures from this month's Workshop on Turtle Graphics.

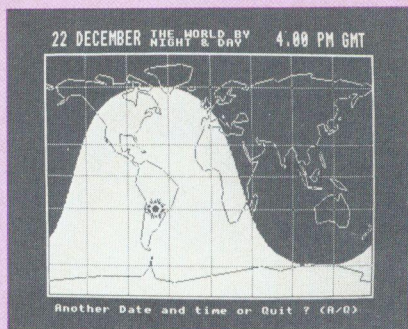
FIRST COURSE — three complete demonstrations of the plotting routines described in this month's article.

AN INTELLIGENT AUTO-SAVE UTILITY — a useful tool for programmers for automatically updating filename and version numbers as you develop a program.

EXTRA FEATURES THIS MONTH

MAGSCAN — data for this issue of BEEBUB (Vol.6 No. 2).

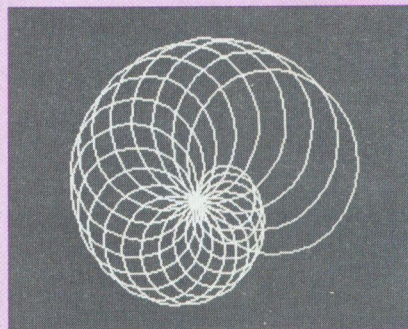
Night and Day



| | | | | | |
|---------------------------------|------------------|---|----|----|----|
| Total no. of items | 202 | | | | |
| Maximum possible mark | 80 | | | | |
| Spread of marks | 70 to 9 | | | | |
| Mean | 37.71 | | | | |
| Standard deviation | 12.09 | | | | |
| New mean ? | 40 | | | | |
| New deviation ? | 12 | | | | |
| STANDARDISED MARKS in column E: | | | | | |
| New Mean | 40.00 | | | | |
| New deviation | 12.00 | | | | |
| A = Mark | C = Frequency | | | | |
| B = Position | D = Cumulative % | | | | |
| A | B | C | D | E | |
| 70 | 1 | 1 | 14 | 72 | * |
| 69 | | | | | |
| 68 | | | | | |
| 67 | 2 | 1 | 24 | 69 | * |
| 66 | | | | | |
| 65 | | | | | |
| 64 | 3 | 1 | 40 | 66 | *_ |

Marks Analysis

Turtle Graphics



All this for £3.00 (cass) £4.75 (5¼" disc) £5.75 (3½" disc) + 50p p&p.

Back issues (5¼" disc since Vol.3 No.1, cass since Vol.1 No.10) available at the same prices.

| Subscription rates (UK only) | 5¼" Disc | 3½" Disc | Cass | Subscription rates (Overseas) | 5¼" Disc | 3½" Disc | Cass |
|------------------------------|----------|----------|--------|-------------------------------|----------|----------|--------|
| 6 Months (5 issues) | £25.50 | £29.50 | £17.00 | 6 months (5 issues) | £30.00 | £34.00 | £20.00 |
| 12 Months (10 issues) | £50.00 | £58.00 | £33.00 | 12 months (10 issues) | £56.00 | £64.00 | £39.00 |

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

Cassette subscriptions can be commuted to 5¼" disc subscription on receipt of £1.70 per issue of the subscription left to run. (3½" disc £2 per issue)

All subscriptions and individual orders to:

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

Telephone: 0727 40303