An Argus Specialist Publication

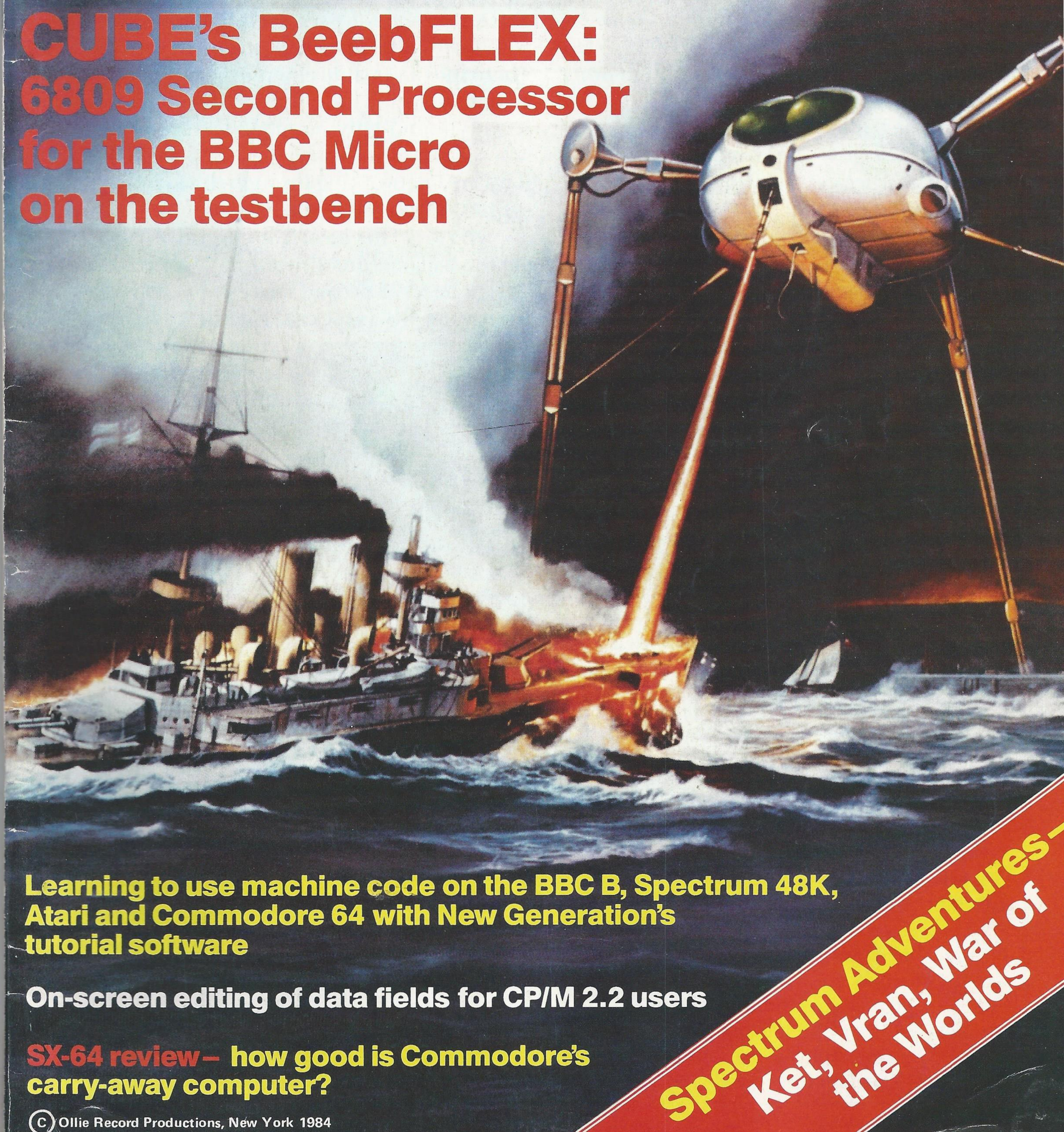
Computing today

SEPTEMBER 1984

90p

INCORPORATING MICRO COMPUTER PRINTOUT incorporating BUSINESS MICRO

CUBE's BeebFLEX: 6809 Second Processor for the BBC Micro on the testbench



Learning to use machine code on the BBC B, Spectrum 48K, Atari and Commodore 64 with New Generation's tutorial software

On-screen editing of data fields for CP/M 2.2 users

SX-64 review – how good is Commodore's carry-away computer?

Spectrum Adventures – Ket, Vran, War of the Worlds

You'd have to spend over £3000 to beat our £199 Second Processor.

The BBC Micro is already one of the fastest and most powerful micros around.

But with the addition of the 6502 Second Processor, it becomes the fastest micro in its price range.

(To be fair to the opposition, their £3000+ package includes a disc drive. But a similar BBC Micro set-up with the 6502 Second Processor will cost you less than a third!)

The 6502 greatly expands the Micro's usable memory. Its 64K of RAM combines with the BBC Micro's 32K, for a total of 96K.

It is supplied with its own special version of BBC BASIC, called Hi-BASIC, which allows the maximum amount of this memory to be used for BASIC programs and variables. Other languages allow some or all of this memory to be used for programs, and many will automatically adjust themselves to make maximum use of available space.

What's more, the 6502 uses the same microprocessor as the BBC Micro, but at a much higher speed. Which means programs can run up to 50% faster.

The 6502's extra power enables it to run more powerful software, such as that provided with the Acorn Bitstick, which turns the BBC Micro into a versatile computer graphics station. In fact, it has a variety of features usually found only on much larger systems.

It can also exploit the full potential of local area networking through the Econet system, with Level 2-File Serving.

So to get the most from your BBC Micro, get the 6502 Second Processor.

The 6502 Second Processor is available from your BBC stockist. For the address of your nearest supplier, ring 01-200 0200. If you wish to order by credit card, phone 0993 79300 during office hours.

<u>6502 Development Programs</u>
<u>(available seperately)</u>

MASM: A 6502 macro-assembler. A full range of macro facilities are provided, including looping recursive calls and conditional assembly.

XREF: A cross-referencer to be used in conjunction with MASM.

ViewEdit: A full screen editor based on the VIEW word processor.

TRACE: A 6502 trace package for de-bugging all types of program.

PRINT: A program to produce formatted assembly listings without using MASM.

The package is provided with a 250-page manual describing all the facilities provided by the system.

<u>Technical Specifications</u>

The Second Processor operates at a clock rate of 3MHz. A version 1.2 MOS will need to be fitted into the BBC Micro before operating the 6502.

Integral power supply

Measurements: 205mm x 345mm

Weight: 2.1 kg

Colour: BBC Computer cream

Construction: Moulded top and bottom to match BBC Computer profile. ABS injection moulded plastic.

Power in: 240v, 50Hz, 3w.

The BBC Microcomputer System.

Designed, produced and distributed by Acorn Computers Limited.

CONTENTS

VOL 6 NO 7 SEPTEMBER 1984

EDITORIAL & ADVERTISEMENT OFFICE
No. 1, Golden Square, London W1R 3AB.
Telephone 01-437 0626. Telex 8811896.

Computing today
SEPTEMBER 1984
CUBE's BeebFLEX:
6809 Second Processor
for the BBC Micro
on the testbench
Learning to use machine code on the BBC B, Spectrum 48K, Atari and Commodore 64 with New Generation's tutorial software
On-screen editing of data fields for CP/M 2.2 users
how good is Commodore's carry-away computer?
Spectrum Adventures — Ket, Vran, War of the Worlds

Editor:
Peter Green

Editorial Assistant:
Fiona Eldridge

Advertisement Manager:
Malcolm Wynd

Advertisement Copy Control:
Sue Couchman,
Ann McDermott

Managing Editor:
Ron Harris BSc

Chief Executive:
T.J. Connell

Origination by Design International.

ABC Member of the Audit Bureau of Circulation

ISSN 0142-7210

Computing Today is normally published on the second Friday in the month preceding cover date. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed by: Alabaster Passmore & Sons Ltd, Maidstone, Kent.

Subscription rates: UK £13.30 including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts HP1 1BB, England.

Computing Today is constantly on the look-out for well written articles and programs. If you think that your efforts meet our standards, please feel free to submit your work to us for consideration.

All material should be typed. Any programs submitted must be listed (cassette tapes and discs will not be accepted) and should be accompanied by sufficient documentation to enable their implementation. Please enclose an SAE if you want your manuscript returned, all submissions will be acknowledged. Any published work will be paid for.

All work for consideration should be sent to the Editor at our Golden Square address.

We regret the lack of photographs accompanying some of the articles dealing with aspects of graphics. This was due to a fire at our developing house which destroyed several films.

Aug/Sept 1984 NO.

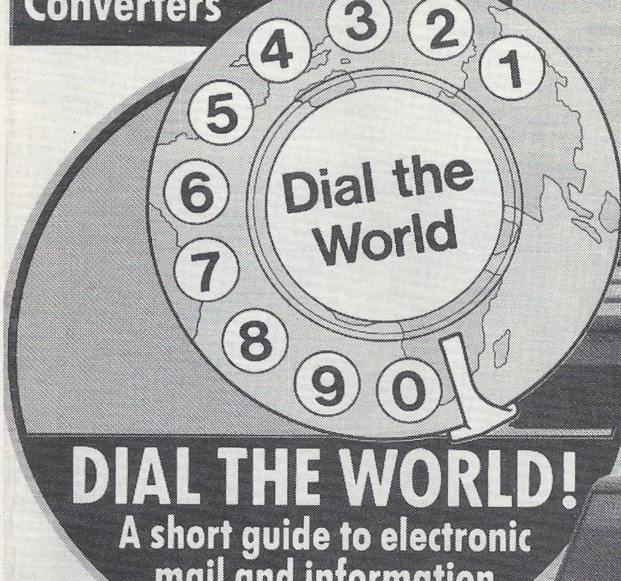
Digital & Micro ELECTRONICS

An Argus Specialist Publication

£1.95

Interfacing Memotech's MTX Computers

Data Conversion Projects Serial ↔ Parallel Converters



DIAL THE WORLD!
A short guide to electronic mail and information systems

Introducing The Amstrad CPC464

DIGITAL & MICRO PROJECTS
Datascope Logic Analyser
Frequency Meter Add-on
Enhanced Transistor Curve Tracer

ADD-ON A
THE BIGGE SURVEY
OF ADD-ON E COMPUTERS

ON SALE NOW!

NEWS

ZX TELETEXT

The TTX 2000 teletext adaptor for exclusive use with either the 16K or 48K Sinclair Spectrum home computer is now available to bring the teletext services to 1.4 million Spectrum users — at a fraction of the cost of a special teletext TV. No modification to the Spectrum is necessary — simply plug in, switch on and enjoy instant access to Ceefax 1, Ceefax 2, Oracle and 4-Tel, provided by BBC1, BBC2, ITV and Channel 4 respectively.

Priced at around £145 (including VAT and P&P) the TTX 2000, power adaptor, ZX interface cable and full detailed instructions are available in one package direct from the makers, O.E. Limited, the communications specialists of North Point, Gilwilly Industrial Estate, Penrith, Cumbria, CA11 9BN (phone 0768 66748). O.E. Ltd. are also the designers and manufacturers of the Prism VTX 5000 modem that recently won the British Microcomputing Awards 1984 'peripheral of the Year' category, and which is also for exclusive use with both Spectrum models.

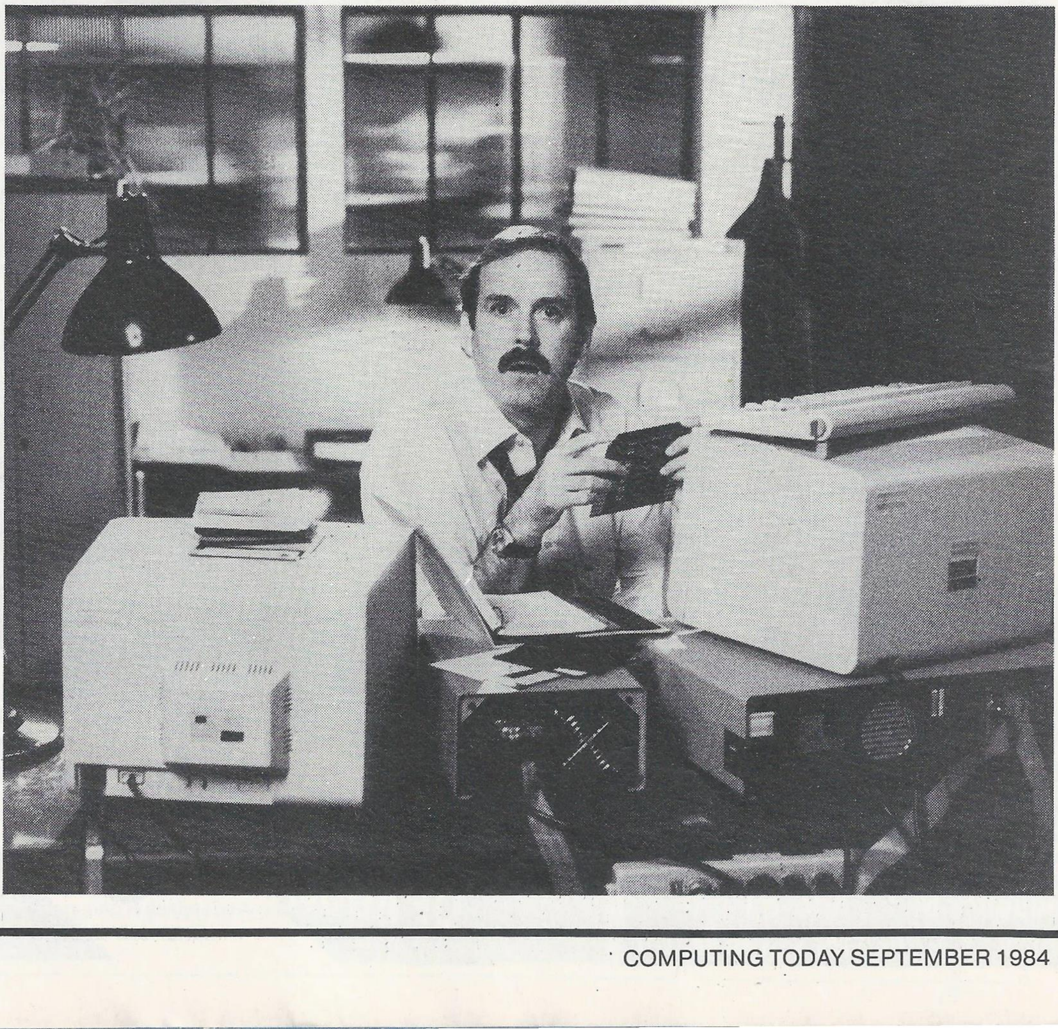
The TTX 2000 sits directly beneath the Spectrum to form a neat and compact unit, with ribbon connector to the micro and an auxiliary power supply lead to plug the adaptor into the mains socket. It will work with any standard black and white or colour TV, displaying full teletext page reproduction.

There are four channel preset controls and pages are called up by simply keying the appropriate number. The usual options such as Hold and Reveal are provided for. Teletext pages can be held on screen, stored on a Microdrive for later recall, or printed out for a permanent record using a Spectrum printer or any compatible unit. The TTX 2000 measures only 9" by 6" by 1.5" and weighs just 1.25 lbs.

O.E. Ltd. have announce plans for a telesoftware program to allow all TTX 2000 owners to receive and download specially broadcast software for use with the ZX Spectrum. The BBC has just started software broadcasting trials for the Spectrum via teletext on Ceefax, and O.E. Ltd. states that the downloader facility will be available shortly as an upgrade ROM.

FAWLTY COMPAQS

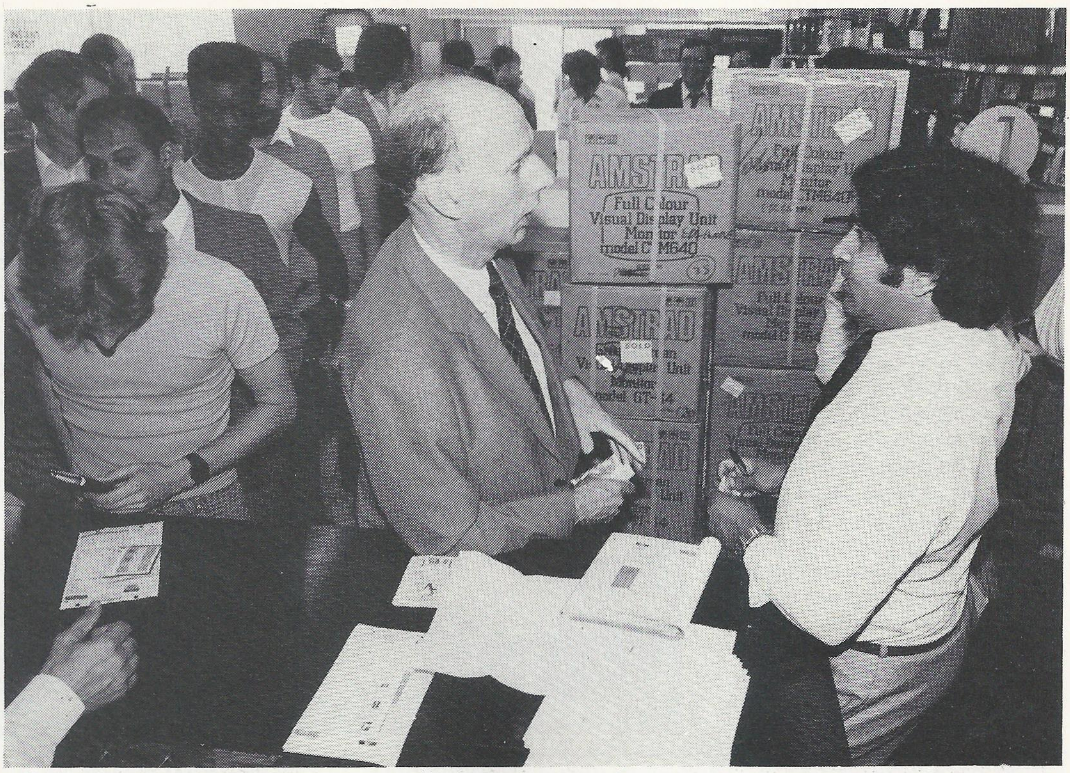
"Now look, Manuel, it's like this. This, computer. This, keyboard. This, smack on the head". Well, not exactly. What's happening is that John Cleese, as much a household name in the USA as he is in Britain, has signed a contract with Compaq to advertise their computer on television there. Cleese plays the role of a businessman who has bought the wrong computer — presumably, not a Compaq. Well, knowing the Cleese skill for rib-tickling, the ads are probably very funny, which is more than can be said of the IBM Charlie Chaplin ads (just what are they trying to get across? Answers on a postcard . . .)



AUTHOR! AUTHOR!

We aren't entirely sure how it happened, but the TRS-80 Screen Scroller published last month was credited to the wrong author. Naturally Mr. T. Ithell, who submitted the program, would like the record put straight, which we are pleased to do here. Sorry, Mr. Ithell!

THE AMSTRAD ARRIVES



The first supplies of the Amstrad CPC464 were on sale on June 21st — keeping Alan Sugar's promise made when he launched the home computer product range in mid-April. The Rumbelow's chain, one of the nominated stockists, has received its first consignment and the other retailers — Dixons, Boots, Comet and the major mail order houses — will be taking delivery of their supplies on a weekly basis, as the container ships reach the UK.

Sales director Dickie Mould says he has been overwhelmed by the intensity of the retailer demand: "The initial interest, at the time of the launch, was perhaps predictable, but this level has been maintained and I am called every day by retailers who want to be supplied. Our initial run of 200,000 units for 1984 is already underwritten by our nominated stockists and we are urgently reassessing our budgets for 1985, when we anticipate bringing in 600,000 units."

Rumbellows' Marketing Director, Peter Jackson, said "We are naturally delighted to be the first in the High Street with this important new introduction in the microcomputer market. Bearing in mind the enormous interest the public have shown in the new product I am also very pleased to see that Amstrad are keeping to their promised delivery schedule."

Meanwhile the independent software houses are dong their bit for the machine. Kuma Computers have announced the first program in their software library for the Amstrad CPC464, this being their ZEN Z80 Assembly Language Programming System consisting of a Editor/Assembler/Disassembler/Debugger.

ZEN is a complete system for the generation and analysis of Z80 assembly language programs. Included are a symbolic assembler and disassembler, a text editor, machine code monitor and a debugger. The complete source code of ZEN itself is supplied to every user; this provides a valuable introduction to Z80 programming for the beginner while allowing the expert to customise it if desired.

ZEN will be invaluable to users wishing to write and debug efficient assembly code programs. ZEN has been tried, tested and enhanced over a considerable period of time, having been implemented on practically all Z80-based home micros to date.

John Day, Sales Manager of Kuma said: "Although ZEN takes up only 6.4K of RAM, it is a complete assembly code system with the benefit of a greatly extended manual, setting an industry standard that will be very hard to beat."

ZEN for the Amstrad is available from Kuma Computers; dealer and distributor enquiries are welcome. ZEN is the first in what is planned to be a comprehensive range of Kuma products for the CPC464. The retail price of ZEN is £19.95 including VAT. For further details regarding Amstrad hardware and software contact Kuma Computers Ltd, 12 Horseshoe Park, Pangbourne, RG8 7JW (phone 070357 4335).

XCALIBUR UNSHEATH XMEM

XCalibur have announced an input/output backplane for the BBC Micro. This unit allows extra input and output cards to be used with the BBC, so increasing its usefulness and range of application. The performance of the machine is further enhanced with the extra 64K RAM expansion built on this expansion chassis.

A major reason for the success of the Apple II (2 million units worldwide) has been its backplane, into which it is possible to plug an incredible range of I/O and expansion facilities. Now XCalibur offers a backplane for the BBC which not only allows expansion, but allows expansion using standard Apple cards. This means that the user or dealer can take an Apple II card off the shelf, plut it into the backplane and run it directly from BBC BASIC or machine code. In use, the input/output addresses of the Apple are mapped into defined addresses on the BBC, which can then drive the card(s) through BASIC commands or Assembler. All XCalibur Apple I/O cards ae compatible and are documented in the XMEM Manual.

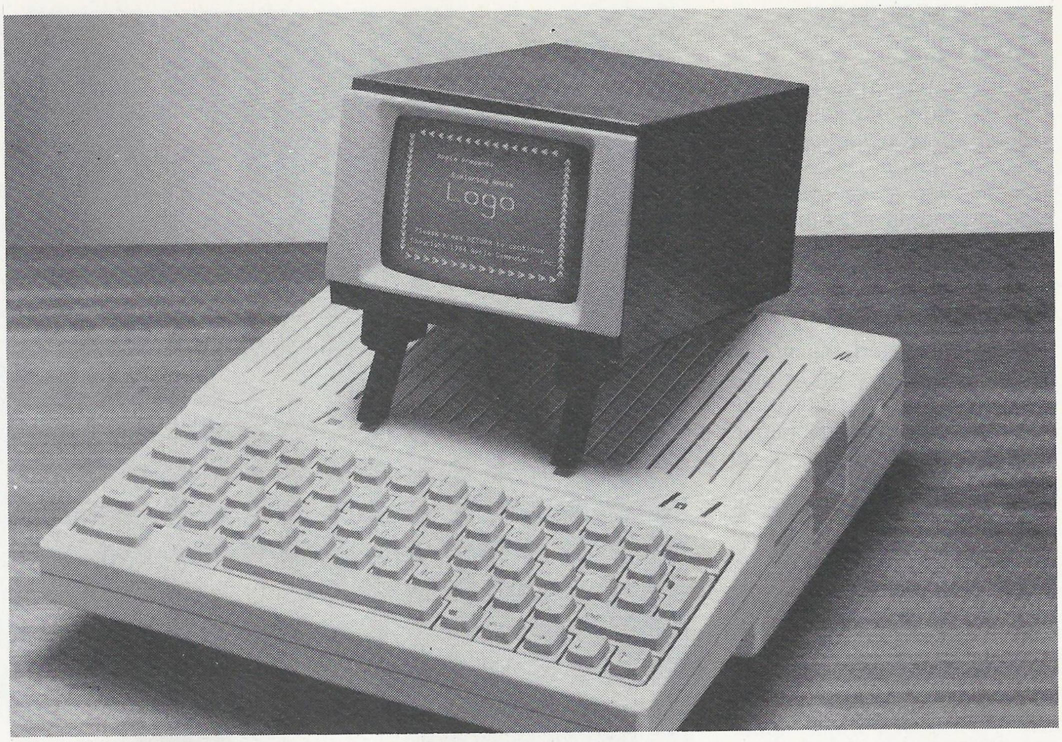
The unit has five slots and 64K of Random Access Memory (RAM). It plugs directly into the 1 MHz Bus expansion socket on the BBC and takes its power from the BBC auxilliary connector. Typical cards readily available are Analogue-To Digital Converters; Digital-to Analogue Converters; Multi-channel Counters; Multi-port Communications; Graphics Processors; Calendar/Clocks; Plotters, Digitizers, Bar Code Wands/Readers; Parallel interfaces and so on. For further information contact XCalibur Computers Ltd, Spencer House, 3 Spencer Parade, Northampton NN1 5AB (phone 0604 21051/4: telex 31612).

LIGHTWEIGHT MONITOR

XCalibur are now supplying a screen for the portable Apple IIc which doesn't give you a hernia when you try to carry it. While the Apple IIc is a lightweight computer suitable for transportation in a briefcase, the additional PSU, second drive and screen can add considerably to the load. By using the 5" screen from the XCalibur Portable, real transportability is now possible for a IIc system.

The screen is connected directly from the Apple IIc, taking both video and power through a single connector. Weight is 1 kg. The accompanying photograph shows that the display is fully legible in normal text and hi-res modes.

Xcalibur are at Spencer House, 3 Spencer Parade, Northampton NN1 5AB (phone 0604 21051/4: telex 31612).



QL SOFTWARE

Quest International Computers Limited and Digital Research Inc. have signed an agreement whereby Quest will implement CP/M-68K on Sinclair's QL microcomputer. Quest are the exclusive licensee worldwide for the product, which will retail for under £50, and will be offering a flexible CP/M implementation on the QL to allow for the wide range of hardware and software options which will be made available for the system. Quest themselves will announce a range of hardware products covering mass storage devices and memory in late September.

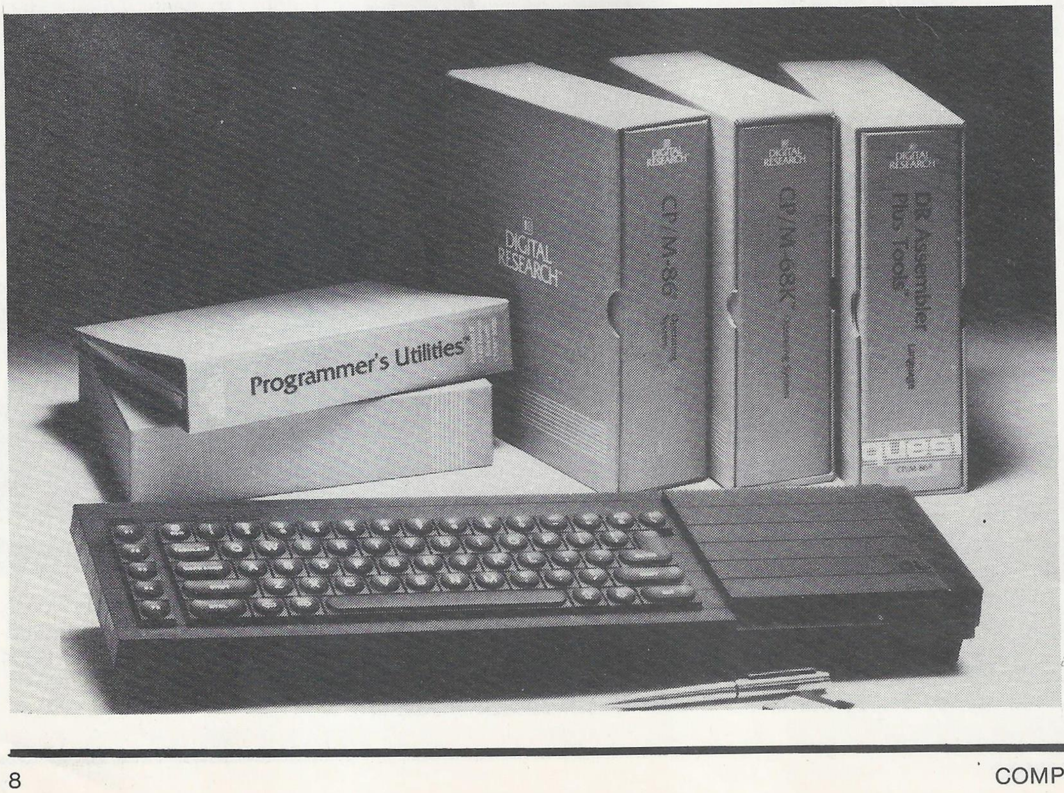
The impact on the QL market is likely to be significant as the combination of an advanced, low-cost machine backed by a full array of storage devices with access to the vast library of tried and tested applications programs under CP/M will inevitably speed the QL's acceptance by the business community (although working hardware would do even more to accomplish this goal).

The end-user price of the CP/M package in the UK is £49.50 (including VAT), a level which Quest believe will make CP/M readily accessible to all QL users. Quest will shortly be appointing distributors in the UK and will do so overseas as and when the Sinclair machine is made available. Of particular interest is the USA where a figure of 300,000 QL sales has recently been mentioned as a possibility within a short period.

Meanwhile Sagesoft Limited, the Newcastle-upon-Tyne software producer, has concluded an agreement with Sinclair Research to provide their Sage Accounts program for the QL. The program, which is being marketed by Sinclair, is being modified by Sagesoft to take advantage of the machine's features and graphics. It will be made compatible with the bundled software where appropriate.

The Sage accounts program is a truly integrated, three ledger book-keeping program which has many of the features of more expensive accounting programs. It provides sales, purchase and nominal ledger accounting with such facilities as VAT return information, aged debtor analysis for credit control, monthly and annual profit and loss statements, cash book, trial balance, and so on.

The entire program will be provided on a single microdrive cartridge with the accounting transactions being held on the second drive. Work has already started on this development and it is anticipated that the QL version of Sage will be launched in Spring 1984. Overseas versions of the system are also planned.



DRIVING UP THE M5

CGL have added to their software range for the CGL M5 Home Computer by introducing eight new game cassettes to their range. The latest games are packaged in the CGL corporate green and black colour scheme and are competitively priced. The single game cassettes have a recommended retail

HARD DISCS FOR BEEB

London based hard disc manufacturer, Intec, have just released a range of Winchester add ons for the BBC micro. Intec's Managing Director, John Groves says, "The range is aimed at the serious commercial and educational user by offering a mass storage with a high degree of sophistication and reliability at a very competitive price. We have deliberately taken an aggressive stance on pricing so that BBC Micro users can get the applied computing power of a true business system with a factor of 3 to 6 saving on hardware costs."

The range offers mass storage in 5, 10 and 20 Mbyte units in both fixed and removable cartridge configurations and, as well as the high speed accessing of files normally associated with Winchester discs, such features as logon, passwords and directory subvolumes are included. Also incorporated are on-board diagnostics to permit low level communication to distinguish and diagnose hardware problems, software bugs and user errors. From a cold start a system can be set up, installed and ready for use in minutes, says Intec. Prices start at £1399 for 5 Mbytes. Intec can be found at 41A - 45 Knight's Hill, West Norwood, London SE27 0HS (Phone 01-761 5999: telex 8813271 GECOMS G).

ACTIVATE YOUR COMMODORE 64

Activision, the independent manufacturer of software for video games machines and home computers, is now producing games cassettes for the Commodore 64 home computer. First titles off the production line are Beamrider, Decathlon, H.E.R.O., Zenji, Toy Bizarre, River Raid, Pitfall I and Pitfall II. Of these titles, Beamrider, Decathlon, River Raid and Pitfall I are new versions of Activision's top-selling 1983 titles, especially adapted and enhanced for the Commodore 64. Pitfall II and H.E.R.O. are Commodore 64 versions of Spring 1984 releases which are already high in the charts, and Zenji and Toy Bizarre are brand new Summer 1984 titles which are being launched simultaneously for the Commodore 64 and other systems.

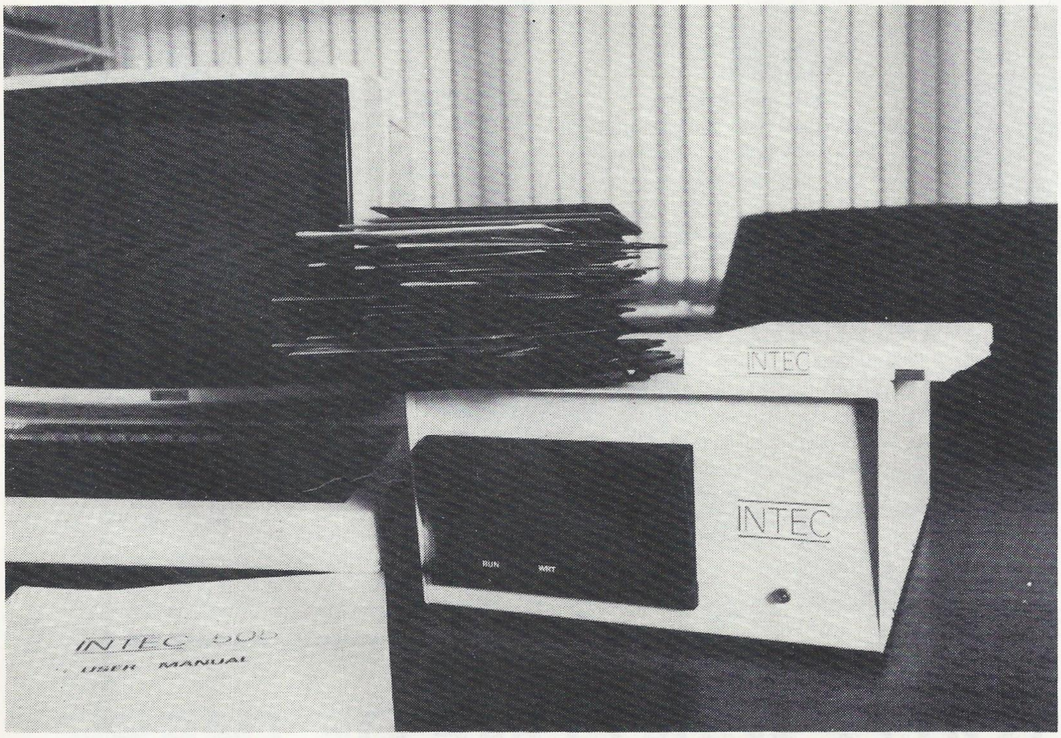
"We see our continuing role in the games software market as the major source for a regular supply of unique and top quality titles for all the top systems" says Activision's UK Managing Director, Geoff Heath. "We have developed versions of our best-selling games over the last eighteen months especially for Commodore 64 users — they are not just adaptations of existing progams — and in Zenji and Toy Bizarre we have two very hot properties indeed."

ENTER THE CHAMP

Dean Electronics have signed an exclusive agreement to distribute the Super Champ Joystick in the UK. This has been the top selling joystick in the United Sttes for the last two years and has several features not previoulsy available to the UK games player.

It has ten feet of cable that can be retracted into the base of the joystick when not in use. A special contoured handle provides maximum comfort for both left and right handed players, while the 360 degree swivel base with sunction cups allows single handed control. And, last but not least, it is robustly built to withstand the severest of physical abuse.

The Champ is compatible with Commodore, Atari, most Sinclair interfaces, Acorn Electron and with an adaptor, the Acorn BBC B. The joystick costs only £12.95 and is available from most High Street stores or by adding £1.00 for post and packing, direct from Dean Electronics Ltd, Glendale Park, Fernbank Road, Ascot, Berks (phone 0344 885661).



price of £5.95 and there are six titles to choose from: Bomber Run, Devil Bird, Intrigue, Slots, Stranded and Wheels.

The two new triple game cassettes have an RRP of £12.95 for three games on one cassette. Titles available at present are: Simon/Granny/Spiders and Squash/Lander/Raiders. If you're a software-starved M5 owner, CGL are at CGL House, Goldings Hill, Loughton, Essex IG10 2RR (phone 01-502 0133).

BROADWAY STEPPING OUT

Broadway Electronics have launched an Electron Owners Club giving members priority servicing, substantial discounts on accessories, and other valuable benefits. The move follows the success of their BBC Owners Club 18 months ago, which now has more than 1,000 members.

Members of the Broadway Electron Club will be able to extend their computer's guarantee for a full year. This covers all parts, labour and servicing by Broadway's own qualified engineers. Work will be completed "while you wait" whenever possible. But if Broadway have to keep the machine more than two days, they will loan a replacement.

Other benefits of the club include: 10% off hardware and accessories, apart from micros; 15% off software; 20% off blank tapes; a regular club newsletter; and exclusive special offers. Membership is £28.75 (including VAT) for Electronics purchased from Broadway's shop in Bedford. For computers bought elsewhere, membership costs £40.25.

Paul Vaughan, Managing Director of Broadway, said: "Many Acorn guarantees will be expiring soon and this is a very economical way to extend the cover. It can run either from the date the original warranty runs out, or from the date of membership. The discounts cover our existing new range of Mushroom add-ons."

Already available is the combined Printer and User Port Card made in Mushroom's own Bedford factory. Complete with manual and software, including a screen dump routine, it allows the use of printers, joysticks and other peripherals. On the way are an analogue port and an extension ROM card, opening the door to word processors and advanced graphics.

Broadway Electronics are at Aston Road, Bedford (phone 0234 58303).

IF YOU'VE GOT ONE OF THESE . . .

. . . THEN YOU'LL WANT ONE OF THESE

YOUR COMMODORE

EVERY MONTH
80p
COMING SOON

Where do 100,000 DP Managers obtain their computer supplies? –From Inmac

Better solutions for professionals-faster!

These days, there are dozens of companies offering to supply your computer needs. They come — and they go.

Yet more and more computer professionals are turning to Inmac for all their requirements.

You don't need a computer to work out why. Inmac are themselves truly professional in every aspect of their business. They don't quote silly prices on loss-leaders — and make it up elsewhere. They provide competitive prices and top-quality service across the board.

And everything carried must be quality. Inmac look after your bottom line — help you provide a more cost effective computer service over all.

Inmac carries many products for leading name equipment like Centronics, DEC, Data General, Diablo, Hewlett Packard, IBM, NEC, Qume, Systime and Texas Instruments.

In addition you get:

*** 30 day risk-free trial,**

*** 12 months' minimum replacement guarantee on most products.**

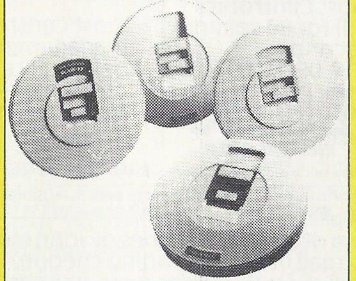
*** Next day delivery and same day in the London area.**

Protect it!

Your computer system is vital — and vulnerable. Inmac supply a whole range of safety and security devices from special fire extinguishers to clever security locks.

PRODUCT OF THE MONTH

Your 5 star DEC cartridge service



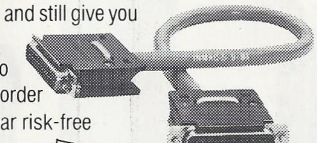
3 YEAR GUARANTEE

30 day risk free trial	**Free Advisory Service**
Next day delivery	**Original DEC quality**

We also supply most other major manufacturers disk cartridges. Just call us to check your requirements.

CABLES AND CONNECTORS

Just tell us what connections you need to make — we will supply your requirements from stock today. We can also custom build cables to your specification in only 5 days. Inmac prices can save you up to 20% and still give you next day delivery, no minimum order and a 3 year risk-free guarantee.

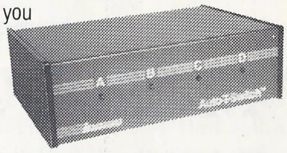


PRINT PERFECT

Whatever the make, whatever the type of printer or typewriter, Inmac have the right ribbon for your needs. Precision manufactured, with high yield and quality results. Print wheels? Ours are compatible with Diablo, Qume, Olivetti, etc.

'T' SWITCHES

You won't find a larger range than Inmac — we supply everything from a simple two device switch up to the super sophisticated Auto-T-Switch which "speaks" to you and cues you automatically.



Ring for our catalogue containing over 1,000 great ideas.

09285 67551 or **01 740 9540**

INMAC (UK) LIMITED
Dept 815 , Davy Road,
Astmoor,
Runcorn, Cheshire,
WA7 1PZ United Kingdom.
Telex: 629819

INMAC (UK) LIMITED
Dept 815 , 16 Silver Road,
London W12 7SG
United Kingdom.
Telex: 924313

Get a better deal all round

The products described on this page are just a tiny selection from the total Inmac range. If you see what you need here — phone us now and we'll deliver it tomorrow or sooner. Just phone us and talk to our free advisory service — we'll do our best to help meet your needs.

Name ____________________

Title ____________________

Company ____________________

Address ____________________

____

____

____________ Postcode ____________

Telephone No. ____________________

My computer system is ____________________

INMAC (UK) LIMITED 815

The Key to Spectrum Machine Code Success.

Picturesque's MACHINE CODE SYSTEM is used and recommended by professional software writers, yet the excellent documentation and the friendly, easy-to-use programs have been highly recommended for beginners.

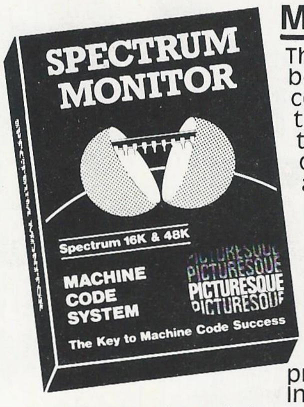
You will only buy one Machine Code System, so buy the best, the one the professionals use.

NEW IMPROVED PROGRAMS — MICRODRIVE COMPATIBLE

ASSEMBLER

Completely self-contained with its own line editor, giving an easy-to-read 40 column tabulated listing. Auto line numbering, line renumbering and auto tabulation make this one of the fastest and easiest Assemblers to use. 5-character label names. Decimal, Hex or ASCII constants. SAVE/LOAD/VERIFY both the listing and Machine Code to cassette/MICRODRIVE/NETWORK. Assemble from cassette or Microdrive into memory for very large programs. Customise to most Centronics printer Interfaces, or RS232 (with Interface 1) for 80 column printout. FAST ASSEMBLY — 1k of code in 7 seconds. Assembler Directives:— ORG, END, DEFB, DEFW, DEFM, EQU, DEFL. (Microdrive and Centronics facilities only operate with 48K machines.)

£8.50
INCL. VAT & P&P.



MONITOR

The ideal tool to help the beginner get started, yet it contains all the commands for the experienced programmer to run and de-bug machine code programs. Inspect and alter memory contents in Hex or ASCII. Breakpoints and full Register display. NOW WITH SINGLE STEPPING through RAM or ROM. Disassemble any part of memory, RAM or ROM. Dec-Hex-Dec number converter. Printer output to ZX printer or via RS232 (with Interface 1) or customise to most Centronics printer Interfaces. General memory management commands include Hex dump, Insert, Delete, Fill and Move. Can reside in memory with the Assembler (48K machines only) to give a complete Machine Code programming system.

£7.50
INCL. VAT & P&P.

Programs supplied on cassette with option to Save onto Microdrive (cartridge not supplied.)

Existing owners can obtain the new programs by returning the **cassette only** to Picturesque, along with a cheque/PO. for **£1.50 per program** (inc. VAT & P&P). New cassettes will be supplied by return of post.

Available from the "SPECTRUM" chain of stores, branches of John Menzies and all good computer shops, or by mail order by sending cheque/PO to:
PICTURESQUE, 6 Corkscrew Hill, West Wickham, Kent, BR4 9BB. Send SAE for details.

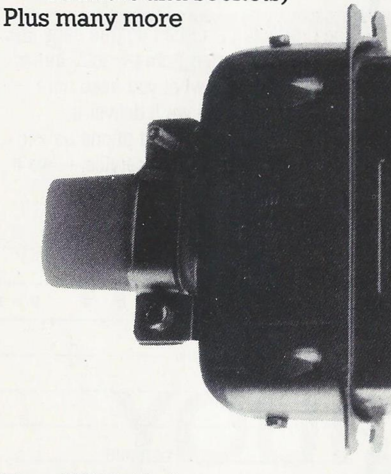
PICTURESQUE

MAKE THE RIGHT CONNECTIONS !

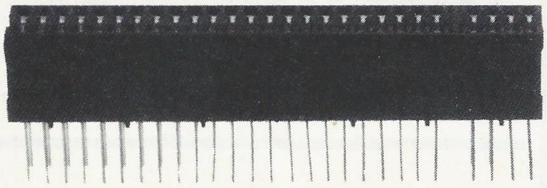
We have the answer whether you need:

- 0.1" or 0.156" edge connectors
- Two part Euro connectors
- Miniature 'D' connectors
- Ribbon cable connectors (including headers and sockets)

Plus many more



For further information on these products and local availability, plus a copy of the 'Hobby Herald', containing the full product range, telephone Chandlers Ford (04215) 62829 or write to :



BICC vero

BICC-VERO ELECTRONICS LIMITED
Retail Dept.,
Industrial Estate,
Chandlers Ford,
Hampshire SO5 3ZR.

The NEW Delta connector to IEEE 488 specification is now available for either ribbon or round cable connection in both 24 and 36 ways. Metal covers are also available.

NEXT MONTH

computing today

AMSTRAD CPC464

One of the more remarkable machines to appear on the home computer scene recently has been the CPC464 from Amstrad, previously known for hi-fi and other domestic electrical equipment. Coming from such a background, you may wonder how it rates against computers from other, more single-minded manufacturers in the field. The answer is, very well indeed. Amstrad have employed a music centre approach to their offering, which consists of a keyboard unit with built-in cassette deck plus a monochrome or colour monitor with a built-in power supply. All very neat, and only one mains plug to worry about.

The software shows just as much thought as the hardware, with an innovative BASIC (interrupts are supported) and CP/M compatibility. Next month the CPC464 will be on our testbench, with an in-depth review of the computer which is already selling fast. Don't miss the October issue of *Computing Today.*

ATARI GRAPHICS

Considering what is it is capable of, the Atari is an oft-neglected machien which is deserving of wider exposure. It gets some of that next month, when we begin a series dealing with the versatile graphics facilities of the beast. Player/missile programming will no longer mystify once you've read the next issue of *Computing Today.*

PSION'S FICTION?

Almost since the genre began, one of the staples of science fiction has been the pocket computer, the electronic slide rule, the database on your wrist. Fact and fiction moved a step closer together last month, when software company Psion launched their first hardware product, a portable, solid-state database/calculator that slips into a shirt pocket at a price that won't damage your trouser pocket. The datapacks are interchangeable, and by buying blank packs you can create your own information bank.

But that's not all. Psion's Organiser comes with a procedure-based programming language that allows you to write and store programs in the datapacks. Next month we hope to have the first review of this innovative product to include a hands-on test of POPL, the Psion Organiser Programming Language. Don't miss it.

Articles described here are in an advanced state of preparation but circumstances may dictate changes to the final contents.

ADVANCE AND BE RECOGNISED

W.H. Smiths have a long tradition in computing, starting with the stocking of magazines through to computer books, software and various pieces of hardware. Now they have launched their 'own brand' computer, the Advance 86, a dual-purpose machine designed, in its A-incarnation, for home use, and for business when expanded to the full-spec B-version. We'll be reviewing the Advance 86B next month and seeing how it will fare in the competitive business marketplace.

DRAGONS AND SPRITES

Dragon may have been down, but they don't seem to be out now that a Spanish company has bought them up. So there's certainly no reason to stop writing about the computer, and in the next issue we'll be looking at ways to perk up the rather dated graphics on this popular machine. Sprites are the name of the game, and we'll be checking out how well various commercial packages weave their magic spells on the screen.

ADVENTURES

Next month it's the turn of the Commodore 64 to receive the scrutiny of our adventurous reviewer. Among the offerings are Legend's version of Valhalla, originally programmed on the Spectrum, and Melbourne House's Classic Adventure, originally programmed on a mainframe by the men who started the whole field of Adventure games rolling.

OCTOBER ISSUE ON SALE SEPTEMBER 14th

GIVE THIS TO YOUR NEWSAGENT

Please reserve me a regular copy of Computing Today

NAME

ADDRESS

POSTCODE

BOOK PAGE

Garry Marshall

One of the most prolific areas in computer books is the 'How to . . .' tome. Our reviewer looks at some new additions in the field, and at a new book on Logo.

Ever since the first micros appeared with their terrible manuals, publishers and authors have been busy producing replacements. The publishing of introductory books on computing, as well as the publishing of manual replacements, has been further stimulated by a genuine demand from readers and computer users for information. Since all this has been going on for some years now, we might have expected that there would be a range of good books available to meet these needs, but it seems that this is not so. I say this not only after having read a number of them, but also in the light of the continuing appearance of books of this kind. Among the latest offerings is the **Screen Shot programming series** from Dorling Kindersley. If we try to see the reasons for the disappointing general standard of the existing books we may develop a viewpoint from which to judge the new ones.

Undoubtedly, a lot of books have been published just to cash in on the computer boom. A lot of these have been written and published very quickly, and contain all the signs of this that we might expect. Inaccuracies, programs that do not work, and so on are all to be found. Unfortunately, the demand for information on computers is so great that it is hard to detect that this has deterred consumers from purchasing the books.

On this score, the Screen Shot series comes out very well. The books are carefully designed and well presented with plenty of colour illustrations. The programs all work (as far as I have examined them, anyway, and have obviously been thoroughly checked. It is indicative of the care that has gone into the production of the books that each one has exactly 64 completely full pages. (Powers of two are good for publishers, as well as computers, because folding a large sheet of paper in half, and folding it again and again will give a number of pages that is a power of two, with the power depending on the number of folds that has been made.)

Another problem with introductory books is whether to write a general introduction or an introduction that is

geared to a particular computer. The attraction of a general book is that it can deal with the underlying principles of computing, which is surely what we need to understand. A treatment that is geared to a particular machine cannot, by its very nature, take a broad view; it can also become little more than an account of the quirks of the computer. But the demand from the market is for books linked to a particular machine. The manual replacements obviously must be linked, but this is also the preference for introductory books. This leads to the production of series of 'cloned' books with one for each of a number of computers. For a series of this kind to be successful, the presentation must follow rather similar lines, while bringing out the strengths of each individual machine.

The Screen Shot books form such a series. The first books to be issued are for the BBC Micro and the Spectrum They start in a fairly staid and familiar way with their introductions to BASIC, but as soon as they move on to sound and, in particular, graphics, the treatments are much more expansive and well judged to show off the capabilities of the different micros to the full.

The selection of the micros that should be covered in these series is interesting. Even before the misfortunes of the Dragon and Lynx, most of the major publishers seem to have decided that the computers to go for are the Spectrum, the BBC Micro, The Electron and the Commodore 64. (I am talking exclusively about the lower end of the personal computer market, of course.) The reasons for this are not hard to find. Perhaps I can encapsulate them by quoting the results of a survey of sales in the UK personal computer market for the first quarter of this year, which gives 43% of unit sales to Sinclair, 28% to Commodore and 10% to Acorn. Besides the present books on the Spectrum and BBC Micro, the Screen Shot series promises books on the Commodore 64, the Electron and, interestingly, the IBM PC Junior.

So the Screen Shot series follows the conventional wisdom for such series, and certainly avoids, at the least, the worst of the pitfalls. Its innovation is to use throughout, pictures taken directly from the screen of the displays produced by the programs that are presented in the books. This definitely enhances one's confidence in the programs. It is rather gimmicky when used to illustrate simple BASIC programs, but comes into its own for graphics programs. The author, Ian Graham, has devised his programs, after the initial ones, to take advantage of the screen shot format, and the books are enhanced as a result. In the second volumes, for both the Spectrum and the BBC Micro, the presentations of the games programs and simulations are better and clearer than they would be otherwise.

Either as introductions to computing or as manual replacements, these books are far above average. I think that they provide treatments that are as good as anything else that is available, and that are far better than most.

I would make a few further comments, which are not meant as criticisms. First, it is difficult to dig out the general principles of, say, graphics when grappling with the way that they are generated by the Spectrum on the BBC Micro. But, as I have already said, this is true of any machine-dependent book. Secondly, the books do provide a general introduction to BASIC, as do almost all such books, but I am becoming increasingly dubious that the route to computer literacy is through BASIC or, indeed, through learning to program in any language. Not many people other than professional programmers will be able to write really useful programs. And the way to make a computer really useful is to load a database or a spreadsheet or whatever program meets your needs, just as the way to turn the computer into a source of entertainment is to load a games program. Information about what spreadsheets and other programs are and what they can do may turn out to be far more useful than learning BASIC. An awareness of program generators might also prove more valuable.

This month's books are:
Step-by-Step Programming for the BBC Micro Books 1 and 2 by Ian Graham (Dorling Kindersley), 64 pages each, £5.95 each
Step-by-Step Programming for the ZX Spectrum Books 1 and 2 by Ian Graham (Dorling Kindersley), 64 pages each, £5.95 each
Introducing Logo by Boris Allan (Granada Publishing), 112 pages, £5.95

Reading Dr Bronowski's **Ascent of Man** again, I came to his point that the human brain is not programmed for speech, vision and so on, but for learning speech and learning to interpret visual patterns on the retina. This shows us one kind of software we badly need for our computers — software that learns to do things (the things that we want it to do) and not software just for doing things.

These thoughts bring me to **Introducing Logo** by Boris Allan, for Logo is the best approach available to us so far for harnessing the computer to allow us to learn with its use. Allan quotes Seymour Papert (the inventor of Logo) quite liberally, and includes remarks of his about how Logo can not only help us to learn, but can also enhance our learning capabilities. Although the book provides a gradual and easy-to-follow introduction to Logo, it does not really show how it can be used as an aid to learning in the way that Papert intends. Some of the illustrations in the book look remarkably similar to the illustrations in the author's QL book, which is rumoured to have given a turtle graphics capability to the QL even before Sinclair knew about it.

ELECTRON ART

Peter Green

Huddled in a tiny garret, your editor has been trying to satisfy his artistic urges, armed only with an Electron and three commercial graphics packages. Some of the results are quite impressive, even though he refused to cut off his ear.

The BBC Micro, and consequently its younger relative the Electron, has just about the most extensive range of graphics modes on any home computer (so far — the Enterprise is still lurking in the wings). With these modes come a vast repertoire of graphics commands for plotting, drawing, filling and generally manipulating the appearance of the TV image. Naturally, these are all of a fairly low-level form: to produce even rudimentary computer graphics requires a bit of programming skill and a lot of patience.

Of course, it was only a matter of time before someone produced a utility package that takes the hard work out of using graphics, by writing useful routines to perform the various 'high-level' drawing functions at a keystroke (or two), leaving the user free to exercise his visual imagination. In this feature we'll be looking at three such programs: Electro-Art from Quicksilva, the Electron Graphics System from Salamander Software, and Sketch Pad from Goldstar (Dorling Kindersley Software Ltd). I tried all three out on an Electron, though the Quicksilva program is a version of their Beeb-Art and Sketch Pad runs on the BBC too.

THE QUICK AND THE DEAD

The first package I loaded was Quicksilva's, and very rudimentary it is too. The program loads quite quickly in two parts, a small machine code block (presumably for the colour fill routine) and the main program which is in BASIC. Once the main program is in, you are asked to decide your background colour and first foreground colour. Here we come to the first black mark against the program: once you have chosen your background colour you are stuck with it, as the only way to change it is to QUIT what you are doing, which takes you back to the start of the program and consequently wipes out your drawing.

No access to the palette rotation facility of the Electron is permitted, which would allow you to change background colour by redefining the logical colour. You can change your foreground colour to any of the eight available (16 if you take Acorn's line that flashing combinations are a different 'colour'), but once it's on the screen you can't use VDU19 to see if it looks better in a different colour. The foreground 'inks' are chosen by pressing the R, G, Y, B, M, C, W or K keys (K is for black which would otherwise clash with B for blue), while flashing is toggled with keys F and O. There is a text prompt on the bottom line of the screen that tells you what drawing mode you are currently in, and as a reminder this too is displayed in the current foreground colour.

Note that I said 16 colours are available: Electro-Art runs in Mode 2. If you want to produce graphics in some other Mode then you're out of luck, because this is the only one supported by Electro-Art.

The initial drawing mode is called 'using cursor', and as you move the cursor around the screen (which is done with the Z and X keys for left and right, the : and / keys for up and down), a trail is left behind in the foreground colour. In other words, this is the freehand drawing mode. You can change colours at any time, and if you want to move elsewhere without leaving a line, pressing I renders the cursor's path invisible. Pressing any of the colour keys, or F or O, puts you back into visible mode.

The mode you have been in all this time is P (for Paint) mode. D (for Draw mode) is the other alternative. The difference between the two is that, when using the predefined shape facility P gives filled-in shapes while D gives outlines. Not surprisingly, you toggle between Paint and Draw using the P and D keys.

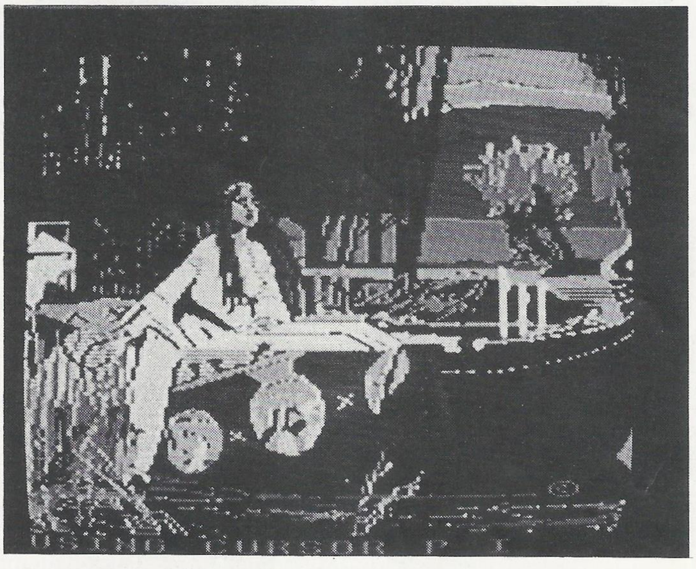
The predefined shapes are drawn onto the screen by selecting the shape number (2 to 9) followed by Shift to draw it. The shapes are square, rectangle, four types of right-angled triangle (the four 'corners' of a diamond), and lines running vertically over the screen height or horizontally across the width. Shape 1 selects the colour fill routine (in theory! See later).

The problem with these shapes is that you cannot alter their physical size. Nor can you alter their orientation. This is, not to put too fine a point on it, daft.

Circles can be drawn using the 0 (zero) and — keys. Move the cursor to the desired centre of the circle and press 0. The text prompt asks you to 'Go to radius length' (by moving the cursor again), at which point pressing — draws the circle to the selected radius. There are two forms of this function; true and abstract (you guessed it, press T or A). The former draws a normal circle as you would expect. Abstract mode draws the circle as a set of spokes, which Quicksilva's manual describes as "even more exciting". No comment.

It is possible to produce your own outline shapes of any size or form using the line-drawing facility. This uses the <and> keys to select the start and end points of the line you wish to draw.

The only saving grace of this package would be the colour fill routine. Unfortunately it doesn't work. No



A reproduction of Quicksilva's demo pic.

matter where I positioned the cursor on the screen or what shape I tried to fill, selecting fill dropped the machine back into BASIC (without an error message). Examining ERR gave 0, which is the default at turn-on and corresponds to 'no room', while ERL gave the line which calls the machine code routine. There are no statements in this line that could cause a No Room error, and indeed typing REPORT gave a blank output, so a genuine BASIC error had not occurred. I can only assume that there is a bug in the machine code that causes a jump into BASIC's error handling routines, and thence back into command mode. The fault was not limited to my own copy — I tried others that we had, and they bombed too. I am not impressed.

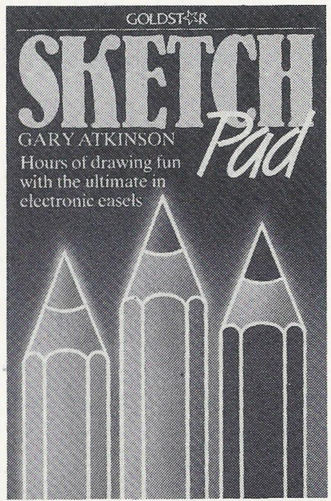
Once you've finished drawing a picture, the whole screen can be SAVEd on tape and LOADed back later. The tape actually contains three example pictures, one of which is reproduced here; but I wasn't able to draw anything worth showing you with this package. I don't see how the example, which are very good, could have been produced, even by a professional artist, unless he spent days painting in the pixels one by one. Not recommended.

SKETCHING WITH GOLDSTAR

The second package comes from the range of software from Goldstar, the brand name for Dorling Kindersley Software. Like the previous offering, the program only works in Mode 2, but the presentation and use is much more professional.

The major complaint I have against this package is the same as the Quicksilva one: no function to alter the physical colours that correspond to the given logical colours. But the total number of colours at your disposal is nine. Yes, nine. Goldstar have not provided an option to select flashing colours (Well, Rembrandt and Van Gogh made do without them!), but you can mix a ninth colour from any two of the other eight. The mixed colour can be in the form of a checkerboard pattern, or vertical or horizontal stripes. The eight 'primary' colours, the two colours to be mixed, and the resultant ninth colour are displayed as a row of boxes along the bottom of the screen, and the current pen colour is selected by pressing Ctrl-C (the Control key is used to access most of the Sketch pad functions).

The cursor keys are used to move the 'drawing head' about the screen, which is more sensible than the Quicksilva system, but I do have one major criticism. On the Electron, the cursor keys are clustered to one side of the Break key. The program does not disable the Break key; and once, after painstakingly building up a picture, my finger slipped and I lost the lot.



Freehand drawing is quite easy using the cursor keys: you use Ctrl-P to raise and lower the 'pen' from the 'paper', and to remind you what the current state is, there's a cute little graphic image of a pen nib displayed at the far right of the prompt box. There is a range of different 'nib' shapes available which are selected by Ctrl-N but I found most of the abstract ones rather useless.

Several drawing utilities can be selected. Line drawing between two selected endpoints (Ctrl-L) is the usual rubber-banding type of utility with a guideline being constantly drawn and erased, but when you are satisfied with its position you have the option of drawing it in solid or dotted form or cancelling it. An extension to this routine, accessed by pressing T (no Ctrl) after drawing in one side using Ctrl-L, allows triangles to be drawn in any size, shape and position. Again, you have the option of drawing it as an outline, filling it in, or erasing it.

Ctrl-R results in rectangles, where you specify the coordinates of the diagonally-opposite corners. The same three options are available as for triangles. Circles are generated by Ctrl-O, which fixes the centre as being the current position of the cursor, and the left and right cursor keys expand and contract the radius. One point here — if you want a big circle things get a bit tedious as the software keeps drawing and erasing the 'guidelines'. Circle drawing on home micros is generally a very slow process, even in machine code, because of the trig calculations required, and this continual redrawing makes things even worse.

The Ctrl-F command, for filling, works. In fact it works very well. It's quite slow, filling a shape point-by-point with the filled area expanding along a diamond shaped wavefront, rather than using the horizontal line-fill function built into the Electron's operating system. Filling is even slower when using the ninth 'composite colour', of course, because of the need to calculate which pixels have to be in each of the two mixed colours. However, the big plus point is that I was unable to find any shape, however complex, that the routine couldn't fill in one go. This contrasts with the final program that we'll be considering, and the previous one which wouldn't fill anything.

Another excellent feature is the enlarge function, which expands a small section of the screen to four times its normal size. This is done by moving the cursor over the area of interest and pressing Ctrl-E. It is now very simple to make accurate pixel-by-pixel alterations to the enlarged area, then cancel the enlargement to redraw that section of the screen back, together with the alterations. This is invaluable for tidying up mistakes or working on fine detail which is difficult to see on a domestic TV.

Text can be added to the picture by selecting Ctrl-T, but take care not to make any mistakes because there isn't any delete facility. Errors have to be corrected by painting over the letters, carefully, and rewriting.

Finally, the finished picture can be saved onto tape using Ctrl-S and loaded back in using Ctrl-G (for Get).

I think my greatest criticism involves, not the graphics program itself, but the loader program that precedes it on the tape. This is a ridiculously self-indulgent piece of code that 'zaps' the name Goldstar onto the screen in giant letters pixel by pixel. I think you can imagine just how long that takes. The only other function that it performs is a small piece of initialisation, so I suggest that you wind the tape past the first program and type in the following in direct mode:

```
IF PAGE=&E00 ?0=1 ELSE
?0=0:*RUN SP
```

This will save you a lot of time.

SUPER SALAMANDER

Finally we come to the Rolls-Royce of the three packages. Salamander's Electron Graphics System is based on a totally different type of system to the other two in the way that it works, and is much the better for it. It also offers many more facilities and using it is a real pleasure.

The difference is that the first two programs are 'what you see is what you get'. As you build up your picture on the screen, the screen itself is the only place that holds the picture information and if you make a mistake, you can't edit it out of the sequence: you simply have to try to repair the damage by overpainting. Imagine what it would be like to slip with the brush on an oil painting, or spill paint on it.

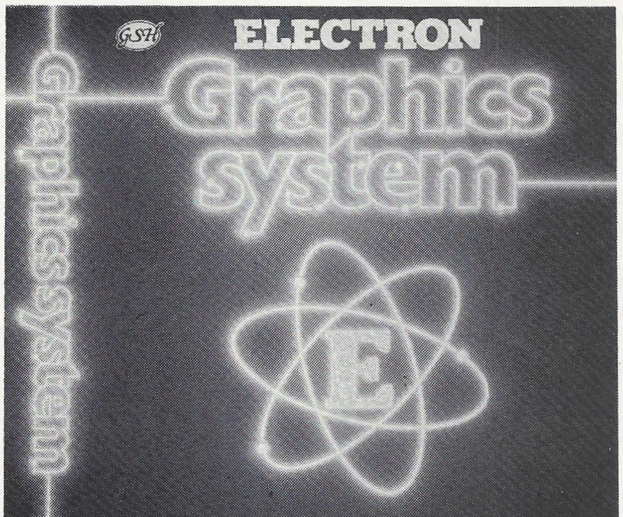
Salamander's software, on the other hand, produces pictures in such a way that mistakes can be 'edited out'. Anything you ask the system to do is reproduced on the screen, but it is also stored in 'Picture Memory': a section of memory that contains a list of the commands used, the colours employed and the screen positions involved for everything you've done. Make a mistake, and you can delete the last item; the screen is cleared and the Picture Memory is 'played back', so the picture you drew reappears exactly as before, minus the last item of course. This makes Salamander's program a similar beast to Acorn's own Bitstik software.

Dorling Kindersley Software have asked us to point out that if anyone purchasing their Sketch Pad tape has a *grey* cassette, then it contains a bug and will be replaced free of charge.

This really frees the novice artist. Although the redrawing takes a bit of time, it's great for your peace of mind to know that whatever you do wrong, you can get back to the previous position.

There are limitations, of course. There's a Picture Memory percentage counter displayed in the text prompt window at the bottom of the screen, and it clocks up towards 100% fairly quickly. The beauty of the system is that, once it's full (or nearly full), you can save it to tape without corrupting the screen. Clear out the Picture Memory and you're free to continue adding to the existing picture, and when you reach 100% again, save the data a second time and carry on once more. Eventually you get a whole set of data blocks, one after the other on the tape, which when loaded into the computer in sequence and executed, will reproduce the original picture exactly.

Naturally the more blocks of Picture Memory make up a picture, the more careful you have to be about making errors since the recovery redrawing procedure can only work with the data in the Picture Memory, and so if you make a mistake in the last block of a picture that's taken 20 blocks to create, you have to go back and re-load all 20 blocks to edit out the error as far as the screen is concerned. Nevertheless, a system in which no mistake is irredeemable has a lot going for it. (Even if you clear picture memory by mistake, it can still be recovered because only a pointer is moved: all the data is still 'there' until you overwrite it with the next drawing command. So if you decide the 'mistake' wasn't, you can un-edit it back into the Picture Memory).



Another neat aspect of this 'picture manipulation' technique is that you can delete an item from the picture memory but not perform a redraw — the screen is left unchanged. This allows you to add temporary guidelines to your work, such as lines to a vanishing point for true perspective diagrams. When you've finished, you obtain the final picture by reloading all the data blocks as already described: the data for the guidelines was removed from memory, so the second time round they don't appear. Clever eh?

It is also possible to draw picture elements whose data blocks are stored on tape to be re-loaded later in a different order, so that new types of picture can be created. However, care must be taken when using this technique: mixing or omitting blocks of data to edit the picture can lead to unexpected effects, such as trying to fill to a non-existent boundary or a boundary that wasn't previously there.

As if the picture manipulation facilities weren't enough, this package has a lot of other goodies to make life easier. Some of the facilities are similar to those discussed in the previous cases: the cursor can be moved around the screen using the cursor keys, and used in conjunction with the pre-defined drawing functions to produce boxes, circles, lines, triangles and filled-in areas. But, for example, the box function requires you to specify the length and relative angle of two adjacent sides, rather than diagonally opposite corners: hence it is possible to draw any parallelogram in any orientation. The arc function draws a curve through three specified points (the curve produced is actually that segment of a circle which can be drawn through the points). Having fixed the end-points of the arc, the third point is constrained to lie on the perpendicular bisector of the end-points, but if anything this makes judging the final effect you will get easier, rather than being a limitation.

Any box, circle or sequence of straight line segments can be re-drawn anywhere else on the screen using the copy function, making the production of regular picture elements a piece of cake. And the positioning on the screen of anything we've discussed so far is made much easier by the inclusion of the actual cursor coordinates in the text window, giving the X, Y positions, the angle the cursor is making with the previous significant line and the distance of the cursor from the previous significant point. No squinting at the screen trying to get accurate positioning here. Moreover, a background grid of dots may be switched on or off at any time as a guide to cursor positioning, and the cursor may be automatically 'homed' to the screen centre, the last significant cursor position, or one of three 'cursor position memories' that can be set by the user.

Pictures may be drawn in any of Modes 0, 1 and 2, and in all modes the full palette is not only available, but can have the physical colours rotated through the logical ones. If you get into too much of a mess, colours can be set back to normal with a single keystroke. At the end of a session, you have the opportunity to save the actual screen memory to tape rather than the Picture Memory data. A picture saved like this cannot be reloaded by the Graphics System the way data blocks can, but you can use the screen in other applications, games for example, by *LOADing it into memory.

The worst thing about the Salamander software is the fill function. This is obviously based on a very simple algorithm, and it can get confused very easily by jagged lines and sharp corners, which cause it to stop short of filling an entire area in one go. Complex shapes thus need several fill commands, and it gets a bit annoying trying to paint in all the little patches that get left out.

CONCLUSIONS

I think you can probably guess what they are before I tell you. Far and away the easiest and most versatile program is the Electron Graphics System from Salamander Software. I was able to draw a very detailed picture of a Japanese Samurai sword, with fully ornamental handle, lying across a monad (yin-yang) symbol, with only an hour's practice. In the same amount of time I was only able to produce a simpler, geometric pattern on the Goldstar system, and very little of consequence on Quicksilva's Electro-Art.

I suppose my ideal drafting package would consist of the Salamander program with the Sketch Pad fill and enlarge routines grafted on. That being impossible, the Graphics System is the best buy with Sketch Pad a creditable second. As for Quicksilva — well, I think the 'Games Lords' had better stick to games.

CUBE'S BEEBFLEX

Narendra Vekaria

At long last, both the promised Second Processors have arrived from Acorn. Meanwhile, other companies have been toiling behind the scenes and one of them, Control Universal, has produced a 6809 Second Processor. We FLEXed its muscles.

About the same time that the 6502B and Z80A second processor were released for the BBC Microcomputer by Acorn, a third eight-bit second processor became available. The CUBE BeebFLEX is a 6809-based second processor for the BBC Microcomputer, and is manufactured by another Cambridge-based computer company, Control Universal Limited. It is available now at around £412.00 (£358.80 excluding VAT).

(Note: to save confusion, products made by Control Universal Ltd are prefixed with the letters CU. TSC stands for 'Technical Systems Consultants, Inc).

When second processors were first considered for the BBC Micro, it was envisaged that they would either provide the user with extra memory to enable the development of much more complex software, or to give increased software compatibility with other systems. The arrival of the 6502 second processor relieved us of the memory gluttons of the Beeb, namely the Disc Filing System, Econet and Telesoftware utilities. The Z80A second processor became available soon after, arming the BBC Micro with CP/M. The decision was left to the user to opt for the system which best suited him or her.

Having said this, you may rightly question the purpose of the 6809 second processor. You would be advised to note that the 6809-based CUBE BeebFLEX is *not* designed to provide the user with extra memory which he or she can use in conjunction with BBC BASIC, so it could not possibly be a viable substitute for the 6502 Second Processor. At best, the CUBE BeebFLEX can offer an alternative form of CP/M as it is capable of running the FLEX operating system which is comparable to CP/M. It is also a very nice system to use for control applications, as the languages and hardware for it are designed for use mainly in this field.

THE CUBE BeebFLEX PACKAGE

The CUBE BeebFLEX package consists of the following items:

● **A CU-NINE 6809 central processor card.** This holds the 6809 processor and a monitor/operating system ROM for use with the FLEX operating system *(not supplied with the package).* The front panel of this card (the side which is exposed to the user) has a 26-way connector socket on it through which the on-board 6809 can communi cate with the Beeb. The on-board ROM contains software which enables industry-standard 40 or 80 track FLEX format 5¼" diskettes to be read and run by the BBC Micro.

● **A CU-DRAM memory card.** This card contains 64K of Dynamic Random Access Memory and a 4K monitor ROM for general RAM support.

● **A metal four-slot mini-rack** with an internal power supply. The case has a beige finish with a black back, which blends very neatly with most BBC Microcomputer systems and their associated disc drives. Two of the four slots are occupied by the above described cards, leaving two more to house the hardware for other applications: eg ROM emulation, In-Circuit emulation, extra analogue or digital I/O, high resolution colour video, or EPROM programming. Further details of these applications are available from Control Universal Ltd.

● **A Tube interface card** for linking the Beeb's Tube to the 26-way connector socket on the front panel of the CU-NINE 6809 card. The card contains a solitary integrated circuit which enables communication between the Beeb and the CUBE.

● **A CUBE BeebFLEX 'sideways' EPROM** which contains the software enabling communciation across the Tube. It also contains a formatting routine which formats discs in order that they may be compatible with the industry-standard FLEX DOS.

- **A six page booklet,** which describes the system hardware and software supplied and also outlines extension possibilities. Additional software support is also illustrated in the booklet.

The operating system ROM on the CU-NINE card enables the BBC Micro to communicate with FLEX. Without it, the CUBE could not access the disc interface on the Beeb's PCB and thus could not control the attached disc drives. Its presence means that the CUBE BeebFLEX can transform the BBC Micro into an intelligent terminal and file server controlling the 6809 unit. When operating, the CUBE does not impair any of the facilities offered by the Beeb and the user can return to BBC BASIC at any time.

Although the second processor has 64K of RAM, only about 36K of it is available to the user when the FLEX operating system is being used. The documentation and the CUBE's memory map suggests that on power-up, the two 4K operating system and monitor ROMs present on the CU-NINE and CU-DRAM cards are copied into the second processor RAM starting at location &E000 (56K along the memory map). Additional memory is gobbled up when the FLEX operating system is used. FLEX resides between locations &C000 (48K along the memory map) and &DFFF (56K), thus taking up an 8K block of memory. Also, when operating, FLEX allocates the first 12K of RAM for its own utility workspace. Remember that no languages have yet been implemented and when they are, the user will be left with considerably less memory.

FIRST IMPRESSIONS

I was rather disappointed with the design of the four-slot mini-rack. The front had a large gaping hole on the right where the additional cards can be inserted. The opening provided a good view of the CU-DRAM PCB, but it also provides a perfect entry site through which dust particles may enter and cause damage to the exposed circuitry. The designers could easily have rectified this problem by inserting a pice of metal or plastic just as Acorn did with the Beeb's ROM cartridge slot.

The 26-way connector cable gives the CUBE an unprofessional look since it leads away from the front panel of the CU-NINE card rather than the back of the rack. It would have been very helpful if the designers had provided all cable connections at the rear as this not only hides the cables but prevents them from interfering with the user.

There are also two large, ominous heatsinks at the rear of the CUBE. These, however, are very necessary due to the fact that the power supply which is contained near the back generates an awful lot of heat. This is because DRAM chips require a lot of energy to function efficiently and within about half an hour of use, the heatsinks show a considerable increase in temperature. The power on/off switch is located on the rear panel. There is no LED or lettering to indicate which state the switch is in, but the accompanying booklet clarifies that down is on, and also the CUBE emits an audible hum when operating.

INTERNAL TROUBLE

The Tube interface card is the most awkward piece of hardware supplied with the package. The card hosts just one 6522 Versatile Interface Adaptor IC and one small diode. Data lines branch away from some of the IC's pins to 26- and 40-way connector cables on either side of the card. The 26-way connector plugs into the socket on the CU-NINE's front panel and the 40-way connector fits snugly into the socket marked 'tube' on the underside of the BBC Micro. I refer to it as a pest because, although instructions are provided to insert the card within the Beeb's case, I cannot help wondering why the CUBE's designers did not install it within the CUBE itself. It would have greatly decreased the amount of work that the user has to put into the system to set it up and running.

If internal installation doesn't bother you, let me tell you that this is not an easy task, especially if you have other peripherals like a sideways RAM card or an extra ROM board already fitted. The space required for the card will undoubtedly conflict with that required for the other peripherals.

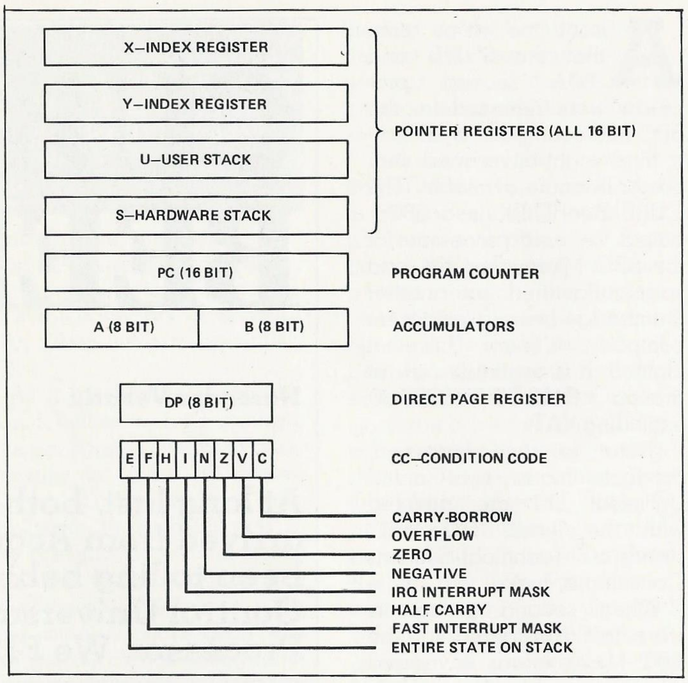


Fig. 1 A programming model for the 6809.

I must mention one last point while we are still on the subject of the Tube interface. You may be surprised to find that when the card is attached to the tube and the BBC Computer is on but the CUBE is switched off, the 6522 VIA on the interface card tends to get very hot. This is because the BBC Micro's own operating system is continuously interrogating its plethora of hardware ports, gathering information from them and adjusting various memory locations and output accordingly. The 6522 on the Tube interface card is thus repeatedly accessed by the micro causing it to heat up. A heatsink should have been attached to the IC because it does get extremely hot and could damage its own circuitry.

Fitting the 'sideways' EPROM is an easy matter and most users may have already gained enough experience to do this without reading the instructions. Clear instructions are provided in the booklet, however, for those who need them. The only important point that the instructions provided forget to mention is that the notch on the EPROM should coincide with the notch on the sideways ROM socket when inserting it. This is the only way you can determine whether the chip is correctly inserted. If other EPROMs/ROMs already occupy all of the available slots on the printed circuit board, then you will have to either sacrifice one of the installed ROMs and insert the BeebFLEX EPRPOM instead or buy a sideways ROM extension board.

DOCUMENTATION

The accompanying six-page booklet was found to be inadequate since it describes the CUBE's facilities superficially. For example, under the section headed SOFTWARE SUPPORT, the documentaion simply hints that the operating system (TSC FLEX) is capable of supporting 'convenient file specification defaults,' but fails to elaborate further. One page only of the booklet is devoted to the RAM monitor on the CU-DRAM card and its commands. The monitor, which is a very powerful piece of software, does not have a commensurate amount of information in the booklet conceming it. Just giving abbreviations, addresses and simple descriptions of the monitor's commands is, in my view, completely unsatisfactory for a product as complex as this.

Some errors crop up elsewhere in the documentation. To actually enter the monitor, the command you type is *FLEX and not *BEEBFLEX which is what the booklet tells you. Errors like this may be trivial but it almost had me pulling my hair out during the half hour I spent trying to find and correct the fault. The information and the way in which it is presented may not bother experienced users of the BBC Micro but it is definitely insufficient for anyone who has only gained about a year's experience of using computers.

All in all, the package could only be described as fair. The CUBE BeebFLEX system however, is an excellent concept and with appropriate software, could become a very powerful tool, for both programming and control applications. The extra extension slot provided on the mini-rack means that, once development is complete and a keyboard attached, the CUBE can be detached from the Beeb and used as a separate eight-bit computer operating under FLEX.

THE PROCESSOR

To fully appreciate the wider range of facilities that the 6809-based CUBE BeebFLEX can offer, it is necessary to have some working knowledge of the processor. In the next couple of paragraphs, I will attempt to outline some of the advantages the 6809 offers over the 6502 and Z80 CPUs.

The 6809 is a fairly recent introduction to home micros, having been implemented only in the Dragon and Tandy Colour computers to date. Made by Motorola, the eight-bit 6809 is capable of addressing the usual 64K of memory as well as being able to support paged RAM/ROM. It is credited as being one of the most powerful eight-bit designs, having two accumulators and being faster than a 6502 or Z80 running at the same frequency. It outperforms the 6502 in that it has more instructions, registers and addressing modes. It also has one extremely useful feature which is very difficult to achieve on the Z80A and the 6502B — it supports relocatable code. The 6809 central processor is therefore a very complex and very powerful eight-bit central processing unit and is not comparable to the Z80 and 6502.

As was mentioned in the previous paragraph, 6809 assembly language can be written such a way that it will run properly no matter where it is placed in memory. This is definitely a boon for machine code programmers as I myself know, when I use the BBC's BASIC Assembler and find that my programs will not run from the addresses at which they are assembled if they attempt to use any of the lower screen modes. This facility may be a major point of consideration for the serious programmer thinking of investing in a second processor.

It must be pointed out that the 6809 supplied with the CUBE BeebFLEX system operates at a clock rate of 1 MHz, and is therefore slower than the 2 MHz 6502 in the Beeb. For those of you who would like to know more about the processors innards, Fig. 1 shows a diagram of the 6809's registers.

USING THE MONITOR

The RAM support monitor installed on the CU-DRAM card is a simple, easy-to-use piece of software which is situated between addresses &F000 and &FFFF. Written in 6809 machine code, it is therefore not compatible with the Beeb's operating system and thus has to be entered indirectly through a 'sideways' ROM routine. Once you have made all the appropriate connections, and switched everything on, the Beeb should display the following message:

```
BBC Computer 32K
Euro-beeb/BeebFLEX
Acorn DFS
BASIC
```

Typing *FLEX followed by RETURN immediately enters the 6809 monitor. The screen changes to Mode 0 with white text on a blue background. The title 'BeebFLEX' is printed in the extreme top left hand corner of the screen and the '+>' monitor prompt is printed two lines underneath. All screen operations from now on are performed in Mode 0, so it would be a great asset if you were using a monitor or a television/monitor. A domestic television is really not suitable as long periods of staring at text on a Mode 0 screen can lead to eyestrain. The characters are also difficult to distinguish on a television, especially if there are interference bands moving across the screen.

An 80-column mode is necessary because only then is the screen capable of displaying a lot of information at the same time, and as the Beeb's resident RAM is not used for anything other than handling the screen, there is no excuse for not taking advantage of this facility. The software in the monitor ROM, therefore, makes extensive use of the 80 columns available.

Pressing the f1 function key on the BBC Micro instantly throws up a list of the monitor commands with extremely abbreviated descriptions of their parameters (See Table 1), similar to the screenful you get when you type *HELP DFS when using BBC BASIC to get a list of the DFS commands. The accompanying booklet gives more information relating to the use of these commands, but even this is a bit sketchy. Having said that, I must also say that some of the monitor commands are very easy to use and I was quite pleased with their operation.

I was also pleased with the screen handling of the monitor routines which output data to the screen. Other routines supplied within the monitor can transfer data to and from the 6809 RAM to the Beeb's 6502 addressed RAM, execute 6809 assembly code at a given address, examine and change memory locations, and display the contents of all the 6809's registers. Some of the commands (namely 'U' and 'F') are not usable unless you have a FLEX operating system disc.

SOFTWARE

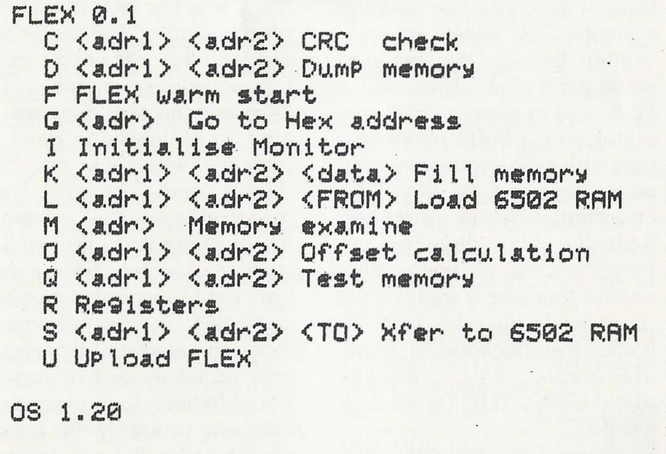
One infuriating aspect about the CUBE BeebFLEX system is that it doesn't come with any 'proper' supporting software. The code contained on the sideways EPROM had some bugs which will no doubt be ironed out given time. As examples of these bugs, I found that once a disc had been formatted to standard FLEX configuration by a routine in the EPROM (using *FLEX FORMAT), the software simply returned control to BASIC instead of entering the monitor as the booklet promised. Also, after entering *EURO (this command is undocumented), the screen switched to Mode 7 and displayed a message giving brief details about the use of the f0 and f9 function keys. This was followed by a complete keyboard disable, the only way out of which was to peform a hard reset by typing CTRL-BREAK.

The ROM-based monitor software resident on the CU-DRAM card offers some interest as it literally invites you to use it to explore the 6809's memory map. However, once using the monitor has become second nature, obtaining a copy of the FLEX operating system will become your primary concern.

FLEX IN GENERAL

The FLEX operating system provides the user with a powerful set of system commands to control all disc

TABLE 1



```
FLEX 0.1
   C <adr1> <adr2> CRC  check
   D <adr1> <adr2> Dump memory
   F FLEX warm start
   G <adr>   Go to Hex address
   I Initialise Monitor
   K <adr1> <adr2> <data> Fill memory
   L <adr1> <adr2> <FROM> Load 6502 RAM
   M <adr>   Memory examine
   O <adr1> <adr2> Offset calculation
   Q <adr1> <adr2> Test memory
   R Registers
   S <adr1> <adr2> <TO> Xfer to 6502 RAM
   U Upload FLEX

OS 1.20
```

operations directly from his/her terminal. FLEX can be divided into three parts; the File Management System (FMS), Disc Operating System (DOS) and the Utility Command Set (UCS). The FMS is actually the code that is uploaded from the FLEX system disc. The code controls such things as command entry and error detection. The DOS routines are supplied on the ROM on the CU-NINE card and the UCS routines come on the FLEX disc.

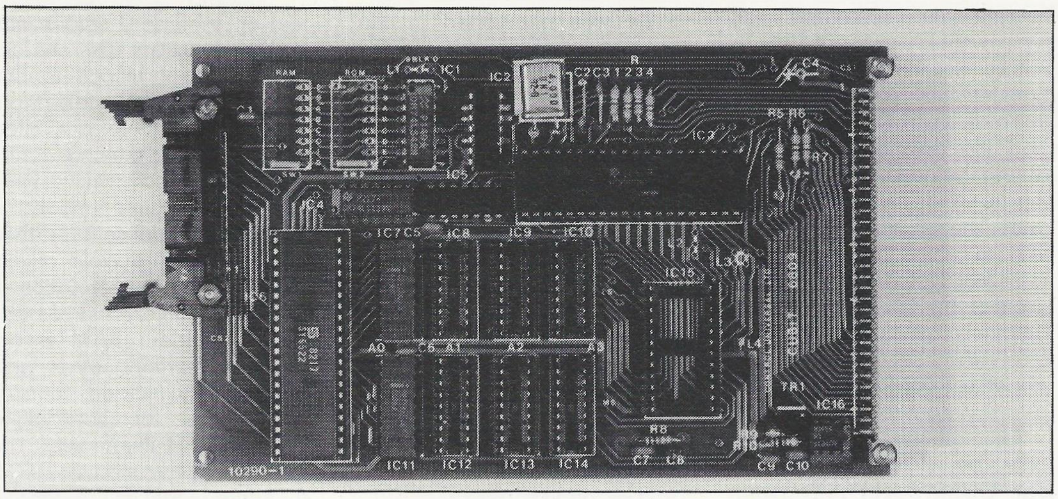
As was explained earlier, FLEX sits in 8K of RAM and needs another 12K at the beginning of memory for general purpose and utility use. The system also expects twin single-sided disc drives (configured as drives 0 and 1) to be interfaced to it. The utility which copies programs is on the FLEX system disc. Therefore, to make backups of the FLEX system disc (which is highly recommended) or any other disc for that matter, requires two disc drives. This is one disadvantage of having the UCS on disc.

The FLEX FMS contains some extremely useful disc management routines. When accessing the disc drives through DOS, it automatically 'removes' any defective disc sectors, expands and compresses all text files accordingly and ensures uniform disc wear by using dynamic space allocation techniques.

USING FLEX

To be able to use the CUBE BeebFLEX system with the wide range of FLEX-based software, you will need to buy the FLEX operating system package. The package includes two 5¼" floppy discs enclosed within plastic sleeves attached to a spiral-bound manual. The manual contains 200 sides of well-Written text on the use and functioning of the FLEX operating system, a 6809 text editor and a 6809 mnemonic assembler. The latter two also come supplied on the FLEX operating system diskettes. Both the 5¼" diskettes are exact copies of each other except that one is for 80 track drives while the other is for 40 track. The package is available from Control Universal for about £75.00 (£65.00 excluding VAT).

Control Universal claim that they chose FLEX as the operating system not only because of the enormous range of existing support software, but also because (and I quote from an advertising leaflet) 'among operating sysstems designed for eight-bit machines, FLEX is the only one which is organised well enough to be machine-independent. Even CP/M (for the Z80 processor) is machine specific to a significant degree'.



To start using FLEX on the CUBE BeebFLEX/BBC Micro system, the FLEX diskette should be inserted in drive 0. Once you have turned everything on and entered the monitor, simply press 'U'. The monitor will bootstrap the FLEX operating system off the disc and into the second processor memory. Sounds simple, doesn't it, but try as I might, there were times when I just couldn't upload FLEX. Sometimes the disc stuck and the disc drive kept reading the same track over and over again. At other times the drive simply kept whirring and grunting endlessly. Admittedly, there were times when FLEX was booted without any problems at all but nine times out of 10 this was not the case. I couldn't bring myself to believe that the disc drive was at fault as it functions perfectly perfectly with ALL my BBC discs and hasn't been responsible for currupting or damaging one yet. The culprit could only be the disc. I am not blaming the suppliers on this issue because magnetic media of this nature can get influenced by a number of factors during transport. This was probably the reason why the disc did not arrive in immaculate condition.

You can immediately tell once FLEX has loaded properly because the following message appears on the screen:

```
FLEX 9.1
DATE (MM,DD,YY)?
```

The 9.1 indicates the version number of the FLEX. The date must be typed in upon every hard entry into FLEX (after uploading it off disc). Having typed the date in and pressed RETURN, the FLEX prompt '+++' appears on the screen, signifying that the system is now operating under FLEX and is ready to accept a command line. A command line is usually a name followed by certain parameters depending on the command being executed. There is no RUN command in FLEX so the moment you press Return after entering a command, FLEX immediately goes to the disc drive and loads in the named disc file whether it be utility, language or machine code program. Thus, if you had a machine code game program called 'INVADERS', all you have to do to load and automatically run is type 'INVADERS' followed by Return. In this sense, FLEX is just like CP/M.

All filenames have three-letter extensions which refer to the language with which they are to be used. This feature is very similar to that found on CP/M. All filenames which have the extension 'CMD' are written in 6809 assembly language and are eigher utilities for FLEX or are actual languages or complete machine code programs which are totally independent language and are either commands on the FLEX disc have the CMD extension signifying that they are part of the FLEX utility command set. 'SAVE' is an unusual command and I have yet to find why its filename extension is 'LOW'.

If you have used CP/M before and ever get an opportunity to use FLEX, you will no doubt have an engrossing session. You will quickly find that FLEX is so much like CP/M that you will already have some experience in using it without even having touched a FLEX-based machine. But you will also find that FLEX is so much more powerful and such a joy to use.

FLEX FACILITIES

Unlike CP/M, FLEX allows the user to have a versatile startup option. Once FLEX is bootstrapped, it automatically searches the disc's directory for a file called STARTUP.TXT. If the file is not present on the system's disc then FLEX outputs the user prompt '+++'. If the file exists, however, FLEX automatically loads it into memory and executes the instructions it contains as if the user was typing them in FLEX command mode. Thus, if the user wanted to load BASIC immediately after FLEX was bootstrapped, he would build the STARTUP file as follows.

```
+++BUILD,STARTUP
=BASIC
=£
+++
```

The file would be saved on disc immediately after the pound sign ('£') is entered. Upon the next FLEX upload, the system will automatically be sent directly to BASIC.

The STARTUP file can also include utility commands and their parameters which the user may wish to use straight after initialization (these can include obtaining a disc directory, setting up terminal parametes (like screen width and depth), assigning the system and working drives and a whole host of other functions). In this way the system is geared for the user and provides him/her with an environment which he/she prefers. In many ways the startup option is comparable to the !BOOT File which can be built onto BBC discs, making it very easy to use and pretty versatile.

FLEX has quite a large number of error commands, most of them being associated with discs. All the commands are extended and hence easy to understand. This is because they are all stored on disc (under the filename 'ERRORS') and hence don't take up any of the second processor's available RAM.

The FLEX FMS is unique because it is capable of doing a large number of things with files. The source of this flexibility comes from the information stored on each disc sector alongside the main data. As a result, 40 track 5¼" FLEX formatted discs can only hold approximately 85K of actual program code; the rest of sector space (4 bytes per sector) is 'grabbed' for system use.

THE 6809 TEXT EDITOR

The 6809 text editor and its documentation comes as part of the FLEX operating system package. The editor itself is the 'EDIT' file on the system disc and once loaded in the correct manner, can be used for creating pure textual files for storage on disc. The editor is not intended to be used as a fully-fledged commercial word processor because it simply hasn't got the capabilities for that. In fact, the code it generates is arranged in such a way that a word processor could read it off the disc, justify and format it, and then output the results to the printer or screen.

Thus the text you type in can be anything from a letter to assembly language mnemonics, depending on what the file that you create is to be processed by. There are a wide range of other programs which use this editor to create files for them; they include language compilers and some interpreters. So you can see, the editor is a very useful piece of software and essential for the FLEX programmer.

The text editor uses line numbers to identify lines of text. The first line is 1.00 and the maximum line number possible is 999.99. The manual reassures that this is hardly likely to be a limitation as the amount of disc storage space required for 10000 null lines (line number plus a carriage return) alone exceeds 40K.

The FLEX editor uses the disc as text storage space. This means that when you type in lines they are immediately committed to disc. The length of the text file is therefore limited only by the amount of disc storage space available.

The manual does credit to the editor program. It contains more information about the functions and features provided by the software than you would normally expect to see. The editor is a lot more complex than it seems. Commands relating to line handling include a renumbering routine and block shifting. General purpose editing commands are also provided. Line numbers can also be turned off in order to create a file for the assembler.

Having written all this, you may be excused in thinking that I was able to see the editor. I must confess that I had no such luck. After several miserable and painful sessions on trying to get the editor to load and run as it was supposed to, I finally gave up in frustration and concluded that the 'EDIT' utility on the disc was corrupted.

THE 6809 MNEMONIC ASSEMBLER

Not being able to use the 'EDIT' facility also meant that I was unable to use the assembler. The reason for this is that the assembler requires a pure text file (one created without any control codes or line numbers) created by the editor. There is no demonstration file on disc, which would have helped me a great deal in testing the assembler. So as not to completely omit it from the review, the information presented in the following pragraphs is mostly gathered from the manual.

The 6809 assembler supplied with the FLEX operating system package is not capable of cross-assembly (assembling programs to work on other 6800 series CPUs) but is suitable for assembling all machine code programs to work on the CUBE BeebFLEX system. The assembler accepts all standard Motorola mneomonics for the 6809 instruction set and their respective addressing modes. Labelling is fully supported and there is also a concise section in the manual on the use of macros, giving details of how to incorporate them into your own programs.

The remaining section of the manual concerning the assembler explains how to use the text editor to create files for assembly purposes and illustrates the various features and functions that are provided. The instructions make it very clear, from the outset, that the manual is in no way intended to be a course in 6809 assembly language programming, but just a guide to using the assembler.

The assembler, in certain aspects, is very easy to use. The minimum command required to make it assemble a text file already created for it is:

```
+++ASMB,<filename>
```

The '+++ ' is the FLEX prompt and the filename is the name of the text file created for assembly.

SOFTWARE SUPPORT

To any programmer using FLEX on the BBC Micro, I would strongly recommend that he/she quickly obtain a copy of Windrush PL/9. PL/9 is a high level compiling language which contains a co-resident Editor, Compiler and Debugger. Its structure is said to be very much resemble BBC BASIC but when programs written on it are compiled directly to independent machine code and then RUN, they are executed about four times as quickly. Having had my attention brought to a review in '68' Micro journal (written 11 years ago), I feel that the author of PL/9, Graham Trott has done a commendable job. The language is well worth getting, especially now that it is available on the BBC. PL/9 is available from Control Universal at £98.00 (excluding VAT).

Other high level languages and compilers available for 6809 are TSC Extended BASIC, Lucidata Pascal, 'C' by Introl Corp and Norel Data systems SPL/M.

Full details of all these and other pieces of software/hardware are available from Control Universal Ltd whose address and telephone number can be found at the end of this review.

CONCLUSIONS

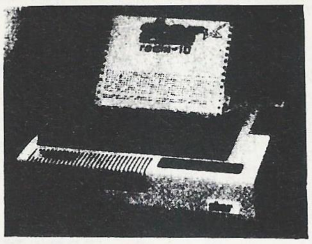
Considering the outlay involved in obtaining the CUBE BeebFLEX, twin disc drives, FLEX operating system disc and other software, the 6809 second processor option cannot possibly sound attractive to most users of the BBC Micro. To be fair on Control Universal, though, the CUBE BeebFLEX system is being pushed towards industry rather than onto home users. Technological establishments can further their research a great deal by using the CUBE in control applications. Most university projects interested in handling robots and other machinery will also find a use for the CUBE.

I was most impressed with the way in which Control Universal dealt with their customers. I recenttly had a telephone conversation with a gentleman working with the company. On mentioning the CUBE, he directly asked me what I wanted to use it for. This sort of reaction suggested that he wasn't prepared to sell me the system unless he was sure I could make use of it.

Of course, there were the usual hardware problems associated with most new products, but Control Universal will soon perfect this system and there is no doubt that a lot of people in research, education and industry will be very pleased with it.

Control Universal Ltd. are at Anderson's Court, Newnham Road, Cambridge CB3 9EZ (Phone 0223 358757).

DOT MATRIX PRINTERS

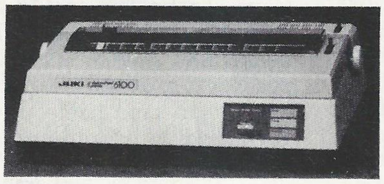


ALL PRINTERS HAVE CENTRONIC PARALLEL INTERFACE UNLESS OTHERWISE STATED. ALL PRINTERS HAVE HI-RES, DOT ADDRESSABLE GRAPHIC MODE, PLEASE SEND SAE FOR FULL DETAILS.

EPSON		
FX 80 160 CPS. 10" wide fric & pin feed	£344 + VAT	£395
FX 100 160 CPS. 15" wide. fric. & tract. feed	£431 + VAT	£495
RX80 F/T 100 CPS. 10" wide. frict. & tract. feed	£261 + VAT	£299
RX80T 100 CPS. 10" wide tractor feed	£234 + VAT	£269
LQ1500 200 CPS Letter quality printer		
8143 RS 23 Interface for FX and RX printers	£29 + VAT	£33
8148 RS232 Interface with 2K buffer. xon.xoff	£60 + VAT	£69
Ribbon Cartridge for RX80 FX80 & MX80	£7 + VAT	£7
Ribbon Cartridge for FX100 & MX100	£12 + VAT	£13
STAR		
Gemini 10X 120 CPS. 10" wide fric & tract feed	£200 + VAT	£229
Gemini 15X 120 CPS. 15" wide. frict. & tract	£295 + VAT	£339
Delta 10 160 CPS. 10" wide. frict & tract. feed	£327 + VAT	£375
Delta 15 160 CPS. 15" wide. frict & tract. feed	£414 + VAT	£475
Radix-10 200 CPS. NLQ. 16K Buffer Par & serial interface. downloadable characters. incredible value at	£431 + VAT	£495
Radix-15 15" wide version of Radix-10	£518 + VAT	£595
SEIKOSHA		
GP100A 50 CPS. 10" wide. tractor feed	£156 + VAT	£179
GP250X 50 CPS. 10" wide. trac feed. ser. & par	£196 + VAT	£225
GP100VC 30CPS. 10" wide. VIC 20 & 64 interf	£174 + VAT	£199
GP500A 50 CPS. 10" wide. tractor feed	£187 + VAT	£215
GP550A 50 CPS. 10" wide friction & tract correspondence quality	£234 + VAT	£269
GP100AS Serial interface. ideal for Spectrum with microdrive	£174 + VAT	£199
SHINWA		
CP80 80 CPS. 10" wide friction & tract. feed	£187 + VAT	£215
Spare ribbon cartridge for CP80	£6 + VAT	£6
RITEMAN		
Compact 120 CPS. 80 col. frict. & tract. feed	£200 + VAT	£229
CANON		
PW1080A 160 CPS. 10" wide. 27 CPS NLQ 24 x 16 matrix	£283 + VAT	£325
PW1156A 160 CPS. 15" wide. 27 CPS NLQ 24 x 16 matrix	£347 + VAT	£399
COLOUR PRINTERS		
Seikosha GP700A 7 colour. 50 CPS printer	347 + VAT	£399
Canon PJ1080A 7 colour. 40 CPS ink jet printer	£391 + VAT	£449
CPP40 40 column 4 colour battery operated	£118 + VAT	£135

ALL OUR PRINTERS HAVE 1 YEAR WARRANTY

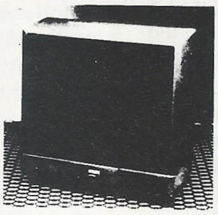
DAISYWHEEL PRINTERS



JUKI 6100		
20 CPS Bi-Directional Logic seeking 10.12.15 CP1 + PS Spacing. 2K buffer. bes selling Daisywheel	£334 + VAT	£395
Singer sheet feeder unit	£217 + VAT	£249
Tractor Unit	£95 + VAT	£109
RS 232 Interface	£52 + VAT	£59
Spare Daisywheel	£17 + VAT	£19
BROTHER HR-15		
13 CPS. Bi-directional. 10.12.15 CP1 + PS	£344 + VAT	£395
Keyboard Unit	£139 + VAT	£159
Single Sheet Feeder Unit	£217 + VAT	£249
Tractor Unit	£95 + VAT	£109
DAISY STEP 2000		
20 CPS Unidirectional. 10.12.15 CP1	£260 + VAT	£299
SMITH CORONA TP1		
12 CPS. 10.12. CP1. unidirectional	£208 + VAT	£239

PROFESSIONAL MONITORS

*SANYO
*MICROVITEC
*FIDELITY
*PHOENIX



SANYO		
DM8112 12" Green screen 18 MHZ Hi-Res	£86 + VAT	£99
DM2112 12" Green screen 15 MHZ Monitor	£66 + VAT	£75
CD3125 14" RGB Normal Res. Colour Monitor	£173 + VAT	£199
CD3117 14" RGB Medium Res. Colour Monitor	£286 + VAT	£329
CD3115 14" RGB High Res. Colour Monitor	£391 + VAT	£449
MICORVITEC CUB		
1431 MS 14" RGB Normal Res. Colour Monitor	£173 + VAT	£199
1451 MS 14" RGB Medium Res. Colour Monitor	£321 + VAT	£369
1441 MS 14" RGB High Res. Colour Monitor	£417 + VAT	£479
FIDELITY Colour Monitor		
CM14 14" RGB RGBY Composite sound col	£187 + VAT	£215
PHOENIX		
Phoenix High Res 12" Green Monitor	£66 + VAT	£75
Phoenix High Res 12" Amber Monitor	£69 + VAT	£79

BBC MICROCOMPUTER SYSTEM

WE ARE AN OFFICIAL BBC COMPUTER DISTRIBUTOR

DEALER ENQUIRIES ARE WELCOMED

Acorn Electron £199 INC VAT

WE SUPPLY FREE 30 HOUR BASIC BOOK AND A DUST COVER WITH EACH BBC COMPUTER

BBC is the best microcomputer currently on the market, 32K RAM, 32K ROM, 8 modes of operation, full colour, full size keyboard, internal expansions such as disc interface, speech synthesizer Econet interface — in short it is a personal computer capable of expanding into a small business system

BBC Microcomputer Model B	£348 + VAT	£399
BBC Mod B - disk interface	£409 + VAT	£469
BBC Mod B - Econet interface	£389 + VAT	£447
BBC Mod B - disk and Econet interfaces	£450 + VAT	£517
BBC 100K disk drive	£230 + VAT	£264
BBC dual 800K disk drive	£699 + VAT	£803
Torch Z80 disk pack including Z80 2nd processor 64K RAM and CPN operating system + Free Perfect Software	£696 + VAT	£799
BBC Teletext receiver (Aug)	£196 + VAT	£225
BBC cassette recorder and lead	£26 + VAT	£29
Disk interface kit (free fitting)	£84 + VAT	£96
Mod A to Mod B upgrade kit	£70 + VAT	£80
Fitting charge for A to B upgrade kit	£20 + VAT	£23
16K memory upgrade kit	£30 + VAT	£34
Games paddles	£11 + VAT	£12
12" Monochrome monitor incl. cable	£89 + VAT	£102
14" Colour monitor incl. cable	£199 + VAT	£22
User guide	£10 + VAT	£10
Econet interface (free fitting)	£60 + VAT	£69
Speech interface (free fitting)	£47 + VAT	£54
BBC disk manual - formating disk	£30 + VAT	£34
Parallel printer cable	£10 + VAT	£11
BBC word processor (view)	£52 + VAT	£59
BBC Fourth language cassette	£15 + VAT	£17
BBC Lisp language cassette	£15 + VAT	£17

APPROVED ECONET SERVICE CENTRE
WE STOCK A LARGE RANGE OF SOFTWARE FOR BBC MICRO INCLUDING ACORNSOFT BBC SOFTWARE LONGMANS SOFTWARE PLEASE SEND LARGE STAMPED ADDRESSED ENVELOPE FOR FULL DETAILS

TORCH Z80 2nd PROCESSOR For only £347 + VAT

Torch 2nd Processor Z80 is supplied with perfect writer (a powerful Word Processor). perfect speller (spelling checking program - I should have used one for making this advert!). Perfect Filer (A Database Program). Perfect Calc (Spread Sheet). It includes 64K memory (Almost 60K available to user) Fits inside BBC Computer

Z80 2nd Process + Perfect Software	£347 + VAT	£399
Z80 Processor + Perfect Software + Dual 800K Disk Drives	£695 + VAT	£799

GUARANTEED LOWEST PRICES

100% BBC COMPATIBLE MITSUBISHI AND TEAC SLIMLINE DISK DRIVES

These drives are supplied ready cased with all the necesary cables. formating program and User Guide.
There are some very useful utilities included on formating disc e.g.
- DISASSEMBLER: This is 6502 machine code disassembler
- DUP: To copy and rename a file on disc
- FORMAT: Formating program for 40 & 80 tracks
- FREE: This utility provides a disk usage analysis
- MDUMP: Enables you to display and modify any part of BBC memory
- MERGE: Merge a number of text files into one file
- RELOCATE: Downloads a basic program to &E00
- SDUMP: Screen dump for EPSON in all graphic modes
- VERIFY: Verifies every sector on a disk
- MENU: A flexible menu program

Single drive 100K 40 trks. single Sided	£147 + VAT	£169
Dual drive 200K. 40 trks. single sided	£286 + VAT	£329
Single drive 200K. 40 trks. double sided	£173 + VAT	£199
Dual drive 400K. 40 trks. double sided	£330 + VAT	£379
Single drive 400K 80 trks. double sided	£199 + VAT	£229
Single drive 400K 40/80 trks. switchable. DS	£217 + VAT	£249
Dual drive 800K 80 trks. double sided	£382 + VAT	£439
Dual drive 800K 40/80 trks. switchable. DS	£408 + VAT	£469

All above drives are low power slimline. (0.3 A typ. at + 12v and 0.4 A at + 5v per drive) Normally extra power supply is not required. the BBC Computer power supply is designed to drive two low power drive (IT IS NOT DESIGNED TO DRIVE INTERNAL ROM BOARD)

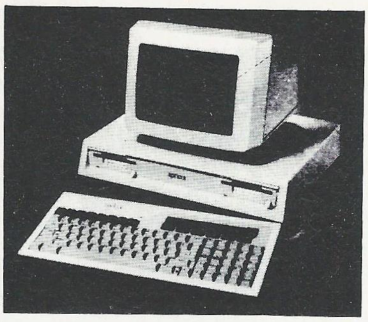
40 Track SS/DD disketts (10/Box)	£18 + VAT	£20
40 Track DS/DD disketts (10/Box)	£23 + VAT	£26
80 Track SS/DD disketts (10/Box)	£28 + VAT	£32
80 Track DS/DD disketts (10/Box)	£30 + VAT	£34

ALL ABOVE DISKETT ARE CERTIFIED EITHER MEMOREX OR DETALIFE

YOUR CONTACTS AT AKHTER ARE

DEALER/BULK ENQUIRIES	HAMAYUN MUGHAL
TELEPHONE ORDERS	PAULA HAYES
DEALER ORDERS	TONY GLOVER
BUSINESS SYSTEMS ENQUIRIES	DENNIS SUTCH
EXPORT ENQUIRIES	MAHAMAD EDIB
ECONET SYSTEM TECHNICAL ENQUIRIES	ALAN LAFFOLEY
ACCOUNTS	CARON ANDREWS
DESPATCH	PAUL SWIFT
REPAIRS	JOHN MAULE

BUSINESS SYSTEMS



APRICOT — SIRIUS — SANYO IBM — TEXAS — TORCH

APRICOT "Portable Executive Computer". 16 Bit Micro, 256K RAM, up to 1.44 megabytes floppy disk storage. 3½" Sony disks. Portable brief case styling. Modem with auto dialler (optional) hard disk optional. Vast software library (compatible with Sirius 1)

Apricot with Single Drive and Motitor	£1690 + VAT
Apricot with Double Drive and Monitor and Free Printer or 2nd Monitor	£1890 + VAT

SANYO PROFESSIONAL COMPUTER
16 Bit Micro 128K RAM expandable to 256K. Single or Double Disk Drive built in full colour graphics (640 x 200 pixels in 8 colours) IBM compatible. Free software. Sanyo MBC 550 128K RAM single drive and free software Including Wordstar and Calcstar £749 + VAT
Sanyo MBC555 128K double drive and free software including Wordstar, Calcstar. Inforstar Datastar etc. £999 + VAT

SIRIUS 1 Sirius 1 Computer with 128K RAM and 1.2 megabyte Floppy disc storage including CP/M 86. MS DOS and Microsoft Basic £2195 + VAT

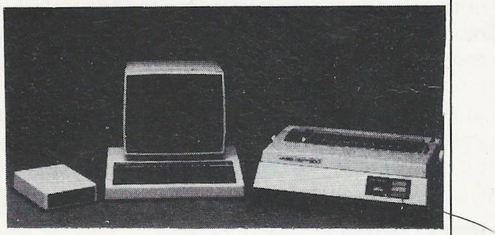
Sirius 1 Computer with 256K RAM and 2.4 megabyte Floppy disk storage Including C C/M 86 MS DOS & Microsoft Basic £2895 + VAT

TORCH
CF500 COMPUTER with colour monitor includes Twin 400K disks. Torchnet Operating System free perfect software (perfect writer. perfect speller. perfect calc perfect filer). Torchmail Plus (Electronic Mail). Torch Mars (Financial modelling with graphics). Torchtel (Viewdata Access System) Executive Aid (Menu System).
Datel 600 (Link to outside mainframe services i.e. Telecom Gold) £2950 + VAT

TEXAS INSTRUMENTS PROFESSIONAL COMPUTER
TI 502 Dual 320KB Drives 256K RAM. Up to 10 megabyte on board storage possible TI speech command hardware option brings a new level of user friendliness to computing. Allows voice operated keyboard control (Optional) Price £2650 + VAT

MAINTENANCE CONTRACTS ARE AVAILABLE.

COMPLETE WORDPROCESSING SYSTEMS INCLUDING DAISYWHEEL PRINTER From £695 + VAT



SYSTEM 1BBC Micro model B. Wordwise. Phoenix monitor. Smith Corona TP1 Daisywheel Printer. Cassett Recorder + all the necessary cables and documentation £696 + VAT = **£799.25**

SYSTEM 2. BBC Micro model B + Disk Interface. Phoenix Monitor. 100K Single Disk Drive. Wordwise. Smith Corona Daisywheel Printer. + all the necessary cables and documentation.
Price only £895 + VAT = **£1029.25**

SYSTEM 3. BBC Micro model B + Disk Interface. Phoenix Monitor. Dual Disk Drives 200K. Wordwise. Smith Corona Daisywheel Printer + all the necessary cables and documentation.
Price £999 + VAT = **£1148.85**

SYSTEM 4. BBC Micro model B + Disk Interface. Sanyo High-Res Green Monitor. Wordwise (or view). Juki 6100 Daisywheel Printer. Single 400K Disk Drive + all the necessary cables and documentation.
£1049 + VAT = **£1206.35**

SYSTEM 5. BBC Mod B Micro + Disk Interface. Sanyo High Res Green Monitor. Wordwise (or view). Juki 6100 Daisywheel Printer. Dual Disk Drive 800K + all the necessary cables and documentation.
Price £1195 + VAT = **£1374.25**

SYSTEM 6. Sanyo 16 Bit. IMB pc Compatible Micro. 128K RAM. Single Disk Drive 160K. Wordstar. Calc Star. Sanyo High Res. Green Monitor. Juki 6100 Daisywheel Printer + all cable and documentation.
Price Only £1145 + VAT = **£1316.75**

SYSTEM 7. Sanyo 16 Bit. IBM pc Compatible Micro 128K RAM. Dual Disk Drives 360K. Wordstar. Calc Star. Mailmerge. Infostar. Spellstar Datastar. Sanyo High Res Green Monitor. Juki 6100 Daisywheel Printer + and all necessary cables and documentation.
Price £1395 + VAT = **£1604.25**

SYSTEM 8. All the components of System 6 but with Sanyo High Res Colour Monitor and Wordstar with Colour Facilities.
Price £1,445 + VAT = **£1661.75**

SYSTEM 9. All the components of System 7 but with Sanyo High Res Colour Monitor and Wordstar with Colour Facilities.
Price £1695 + VAT = **£1949.25**

WE CAN ALSO SUPPLY PACKAGES WITH COMPONENTS OF YOUR OWN CHOICE. PLEASE PHONE (0279) 412639 AND ASK FOR HAMAYUN MUGHAL FOR A PRICE.

AKHTER INSTRUMENTS LTD T A

AKHTER COMPUTER GROUP

28/29 BURNT MILL
HARLOW, ESSEX. CM20 2HU U.K.
Tel. HARLOW (0279) 443521 Telex: 818894 AKHTER G

ORDERING INFORMATION:

We accept official orders from UK Government and Education establishments. Carriage is Free (UK only) for normal delivery. If express delivery is required please add £8.00 + VAT per parcel. We accpet telephone orders on Barclay and Access Card please ring (0279) 443521 (10 lines). All cheques made payable to "AKHTER INSTRUMENTS".

Access VISA

N.B. ALL VAT INCLUDED PRICES ARE THE CORRECT PRICES, EXCL. VAT PRICES HAVE BEEN ROUNDED UP TO NEAREST POUND.

OPENING HOURS: MON-FRI 9am-6.30pm, SAT 10am-5pm
We welcome callers, no parking problems.

AN ARGUS SPECIALIST PUBLICATION

Orchard COMPUTING

Volume 1
Number 4

Autumn
1984

Britain's best buy for Apple users £2.95

A great harvest of software for your Apple II and IIe

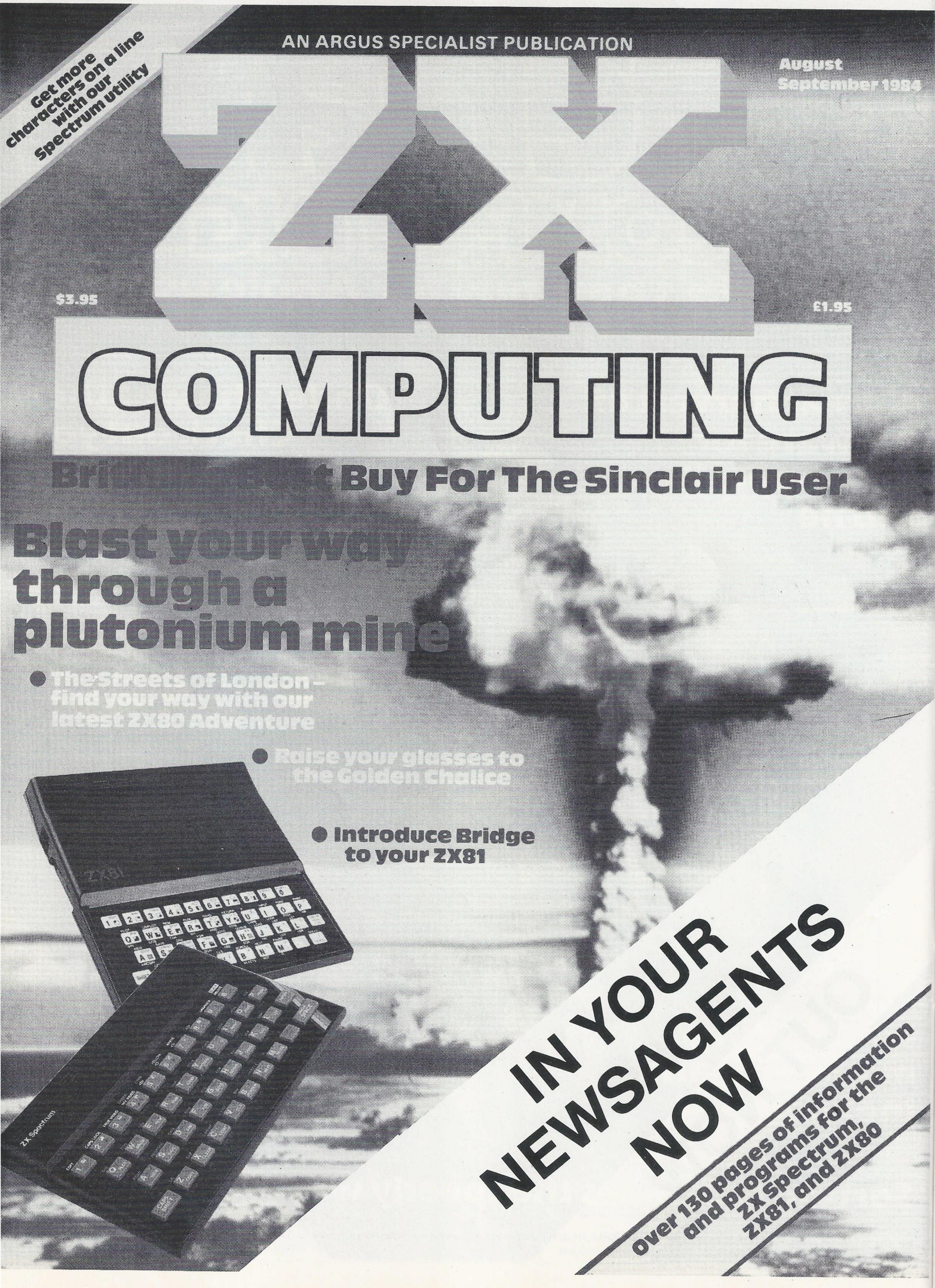
Under the weather? Check out your biorhythms

OUT NOW

Lin ing
mad and
inexpe

Sorting tho s the speedy way

Have your own Electronic Message Centre


AN ARGUS SPECIALIST PUBLICATION
Get more characters on a line with our Spectrum utility
August September 1984
ZX
$3.95
£1.95
COMPUTING
Britain's Best Buy For The Sinclair User
Blast your way through a plutonium mine
● The Streets of London – find your way with our latest ZX80 Adventure
● Raise your glasses to the Golden Chalice
● Introduce Bridge to your ZX81
ZX81
ZX Spectrum

MICROTANIC COMPUTER SYSTEMS LTD.

MICROTAN 65 *NO OTHER COMPUTER IS AS PERSONAL!*

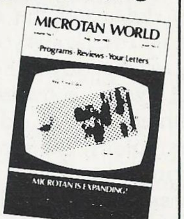
For less than £60 you can start building your own Computer that truly suits your needs and, of course, eventually far more superior to any Computer available off-the-shelf.

MICROTAN 65 comes in kit form, complete with manual, full instructions, board with components, (kit form or fully built) our full back-up service, and your own Microtan World Magazine available on subscription.

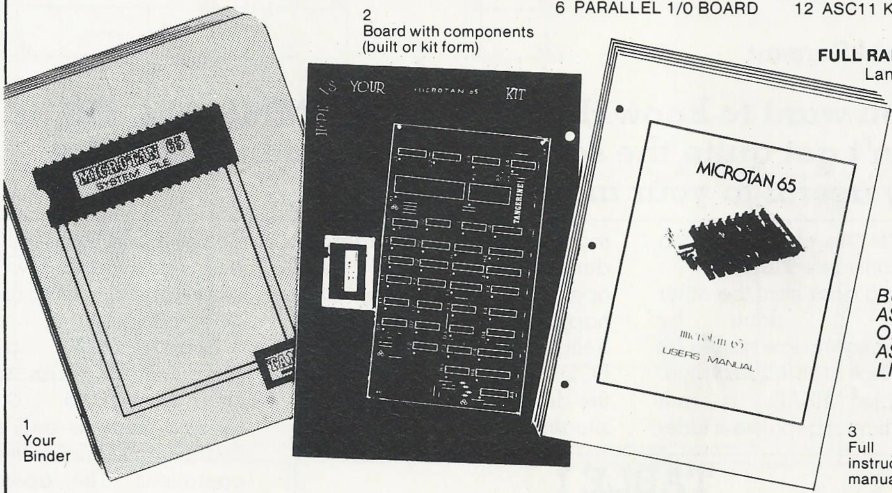
FLEXIBLE & EXPANDABLE SYSTEM — 1K to 256K!

Just look at the options:

1 DISK CONTROLLER
2 REAL TIME CLOCK
3 EPROM PROG. CARD
4 SOUND BOARD
5 SERIAL 1/0 BOARD
6 PARALLEL 1/0 BOARD
7 MASS EPROM STORAGE BOARD
8 INDUSTRIAL CONTROLLER BOARD
9 40K RAM BOARD
10 HIGH RES. GRAPHICS 256x256
11 PRINTER FACE BOARD
12 ASC11 KEYBOARD



Microtan World Magazine



2
Board with components
(built or kit form)

1
Your
Binder

3
Full
instructions
manual

BUILD AS FAST OR SLOW AS YOU LIKE!

FULL RANGE OF SOFTWARE
Languages available: Machine Code, Assembly, Basic, Forth, and Pilot

HOW TO ORDER:

Enter details in the coupon below, enclosing your cheque made payable to: Microtanic Computer Systems Ltd. Prices include VAT add £1.50 p&p. Please allow 14 days for delivery.

Post to:
MICROTANIC COMPUTER SYSTEMS LTD
16 UPLAND RD. LONDON SE22
Tel No. 01-693 1137

Please rush me my starter kit:
(Please tick)

☐ kit form — I will build myself £59.95
☐ Fully Built £69.95
☐ Complete system wall chart

I enclose my cheque/P.O. for £

Name ....
Company ....
Address............................
....
....
Tel (Day) ....
Tel (Eve) ....

Access

CT/9

MICROTANIC COMPUTER SYSTEMS LTD

SHOWROOM: 16 UPLAND RD
DULWICH, LONDON SE22
TEL: 01-693-1137

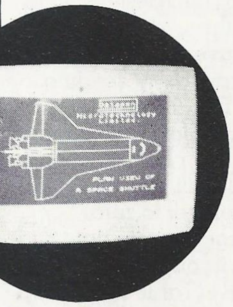
MAIL ORDER:
235 FRIERN RD. DULWICH
LONDON SE22

Also available from:
Waltham Forest Computer Centre
889 Lee Bridge Rd.
Nr Whipps Cross, Walthamstow E7
Tel: 01-520 7747

Datapen

A QUALITY LIGHTPEN
for use with the SPECTRUM computer

The DATAPEN lightpen enables you to create high resolution pictures and technical layouts directly on your T.V. screen. The accompanying software allows you to draw any shape or *filled area* you wish, to *pixel accuracy*, in full colour and the results may be utilised within your own programs, e.g. for animation, or to illustrate your title pages. At just £29 inclusive, the Datapen lightpen package represents superb value – just look at the actual screen photographs and you will agree that this must be the best value for money on the market.



PLUS: 3 SOFTWARE PROGRAMS

- Routines and ideas for your own programs. (Menus, games, music, etc.)
- User-defined graphics creation program.
- Superb full colour drawing program as illustrated in these actual screen photographs.

- Incorporates features not provided with other pens.
- Push button operation on pen – no need to use keyboard.
- Works under any lighting conditions.
- Plots to pixel accuracy.
- 20 pre-defined commands allow plotting of geometric shapes, including triangle, lines, circles, etc., text, or user-defined characters.
- Uses all paper and ink colours.
- A screen grid may be turned on and off, at will, to assist drawing.
- Flexible erase capability.
- All drawings can be saved to tape for further use.
- Plugs in direct – no batteries, additional components, or adjustments needed.
- Handbook, plus printout of routines for use in your own programs.

Send cheque or P.O. for £29.00 to: Dept. DATAPEN MICROTECHNOLOGY LIMITED (CT2), Kingsclere Road, OVERTON, Hants RG25 3JB
or ask at your local computer shop. Send S.A.E. for details. 0256 770488 Lightpens and software are also available for BBC B, Dragon, CBM-64 and VIC-20.

REAL-TIME CLOCK

Richard Sargent

If you want to know the time, ask an MM58174. OK, it hasn't got quite the same ring about it but it can be very useful to your micro.

These days most computers have internal timers which can be configured to count and display in hours, minutes and seconds, but they can't tell you the date and the day of the week. Yes, I know you probably know what day of the week it is, but your software doesn't. Futhermore, if you disable your computer's interrupts, the chances are that it will lose all sense of time completely.

This project describes the construction of a Real-Time Card with battery back-up. Designed to interface with an eight-bit output latch and an eight-bit input buffer, it should prove suitable for a host of micros. The software and port addresses given in this project are for the powerful MTX 500/512 Micro, which has the advantage of an uncommitted 16-bit I/O socket nestling on its PCB!

THE MM58174

The MM58174 has a total of 16 internal registers which hold time and control data (see Table 1). Normally you would use the CPU address, data and READ and WRITE lines to access these registers as though they were any normal memory location, but in the design presented here it is software acting through the I/O port which communicates with the chip. For the present, though, it is easier to assume that straightforward READs and WRITEs are being made to the timer chip.

When the timer is first powered up it is necessary to enter the correct data into the device registers and start the clock running. A nibble (four bytes) on the data pins is used to pass BCD data to a correctly addressed register and in this manner all the internal counters are set to the desired time. When the time-keeping registers have all been set, the clock is started by sending a high on D0 to register 14. Conversely, a low written there will stop the clock counting. Incidentally, all starts reset the seconds counter to zero.

Don't forget to write data 8 to register 15 if you're building this card in 1984 (Table 2, the leap year status register). By some strange oversight there is no readout capability on this register, so the micro can't tell if the current year is a leap year! However, there is a hardware solution to this problem which will be considered later.

Reading data from the other registers is done by interrogating the low nibble for a valid clock or calendar value. A simple READ is not quite sufficient because a timer register might be updated during the actual read operation, and when this happens the MM58174 will deliberately return the illegal BCD code 1111. This enables the detection of a faulty READ situation. Software running slowly under BASIC will tend to pick up quite a few "1111" codes, and they must, of course, all be trapped.

TABLE 1

MM58174 INTERNAL REGISTERS

No	Name	Mode
0	Not used	
1	Tenths of sec	R
2	Units of sec	R
3	Tens of sec	R
4	Units of mins	R/W
5	Tens of mins	R/W
6	Units of hours	R/W
7	Tens of hours	R/W
8	Units of days	R/W
9	Tens of days	R/W
10	Day of week	R/W
11	Units of month	R/W
12	Tens of month	R/W
13	Years	W
14	Stop/start	W
15	Interrupt	R/W

TABLE 2

YEARS STATUS REGISTER

	DB3	DB2	DB1	DB0
Leap year	1	0	0	0
Leap year + 1	0	1	0	0
Leap year + 2	0	0	1	0
Leap year + 3	0	0	0	1

This register is a shift register and the contents are rotated to the right every 31st December.

TABLE 3

INTERRUPT READ/WRITE REGISTER

Function	DB3	DB2	DB1	DB0
No interrupt	0	0	0	0
Interrupt at 60sec intervals	0/1	1	0	0
Interrupt at 5.0s intervals	0/1	0	1	0
Interrupt at 0.5s intervals	0/1	0	0	1

Write mode:
DB3=0 single interrupt DB3=1 repeated interrupt
Read mode:
DB3=0 no interrupt DB3=1 no interrupt

Register 15 can be programmed as an interrupt timer giving 0.5, 5.0 or 60 second intervals and can be coded for single or repeated operations. The open drain interrupt output is pulled to ground when the timer times out and reading the interrupt register provides status and internal selected information. See Table 3.

CIRCUIT DESCRIPTION

Points T0-T7 are the eight inputs of the host computer's input port which will typically be a 74LS244 (tri-state buffer) or a PIA/VIA configured as a tri-state buffer. The MTX uses a 74LS373, a transparent latch, and this may be considered as an ordinary tri-state buffer in this application.

Points L0-L7 are the eight outputs from the host computer's output port. This will usually be a 74LS374 (octal latch), as is the case of the MTX, or it might be the other half of the PIA etc. The outputs must be latched so they maintain the same state until told by software to change.

The outputs L0-L3 pass through another latch, IC4, which holds their value and thus stabilises an address on the A0-A3 pins of the timer, IC1. An ALE (address latch enable) pulse achieves this. While this is happening, READ and WRITE will be held high, thus preventing the same data from entering at the D0-D3 pins of IC1. Writing data to IC1 involves enabling the four tri-

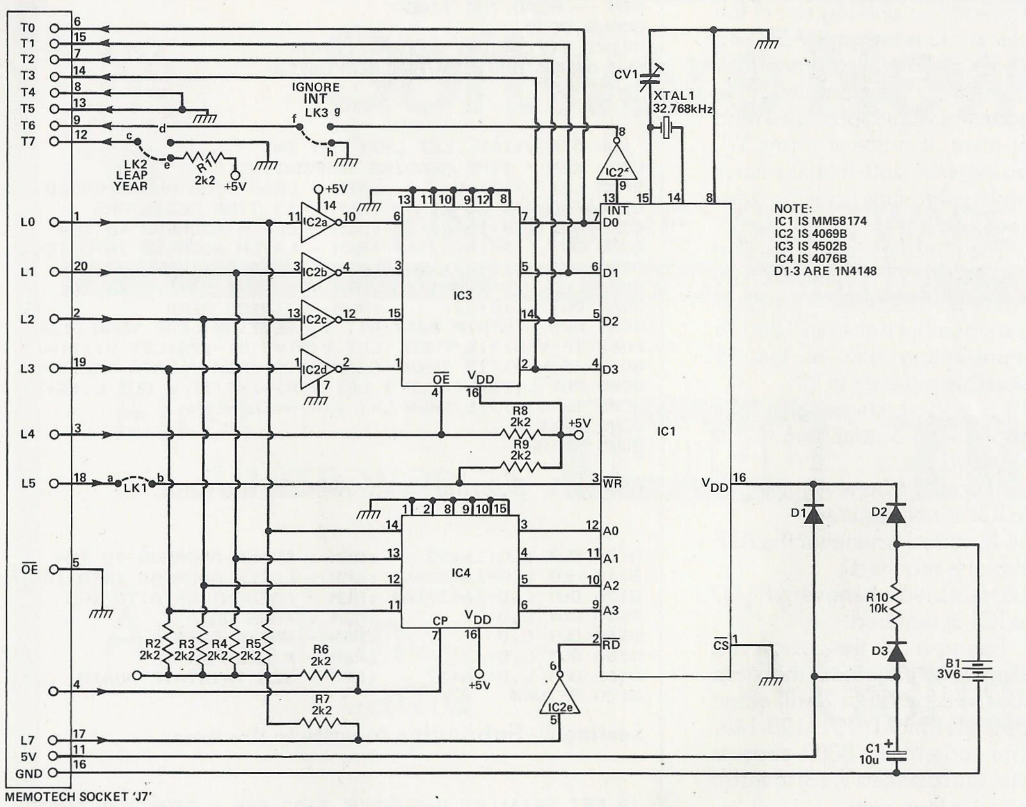


Fig. 1 Circuit diagram of the Real-time clock.

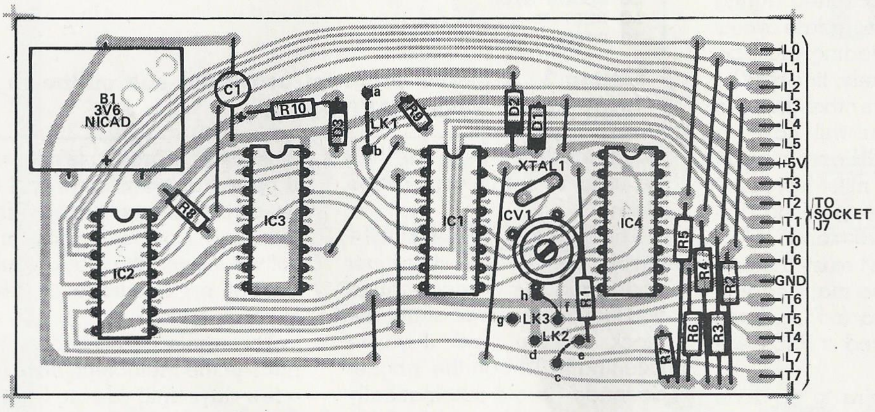


Fig. 2 Component overlay.

PARTS LIST

IC1	MM58174 CMOS timer
IC2	4069B CMOS inverter
IC3	4502B CMOS tri-state inverter
IC4	4076B CMOS latch
R1	10k ¼W 5%
R2-10	2k2 ¼W 5%
C1	10uF 16V tantalum capacitor
CV1	5-60pF trimmer (Cirkit 06-60001)
XTAL1	32,768 kHz crystal
D1-3	1N4148 diodes
NiCad	3V6 (Cirkit 01-03100)
IC sockets	three off 16-pin, one off 14-pin
One 20-pin header	
18 or 20-way ribbon cable	
PCB	

state buffes in IC3 with a PWR (pre-write) pulse and then allowing the data through to IC1 with a WR pulse.

The inverter is needed on the READ line to guard against a situation where zero is written to the output latch L0-L7. The host computer's ROM might do this as an initialisation, or it might occur after a system crash. Without IC2 an all-zeros situation would set the MM58174 to a simultaneous READ and WRITE and that's not a good idea . . . .

If you want to permanently disable the WRITE line you should cut link LK1 on the PCB. The clock data cannot now be accidentally overwritten. Access to the interrupt register is also denied but periodic interrupts may still be read.

The T0-T7 lines read the timer directly, and it may prove convenient to tie the four most-significant bits to ground so that the complete eight-bit number represents the time value. Personally, I would rather allocate T7 as a leap-year indicator, and T6 as the interrupt reader, in which case the software must unscramble the information contained in those eight bits.

The 2k2 pull-up resistors on the L0-L7 are a necessary requirement when driving CMOS from TTL. The timer's clock is formed using a standard 32.768 kHz crystal across pins 14 and 15 and it is recommended that a 5-6pF trimmer be used to fine-tune the oscillator.

The remaining gate in IC2 is used as a buffer on the interrupt line, and its output may be fed to external equipment, if desired.

IC1's V_{DD} is connected to the battery supply line and the chip will maintain its data in standby mode on voltages down to 2V2. The MTX 5 V rail is capable of supplying the 1mA needed to trickle-charge the small NiCad battery which provides a nominal 3V6 standby voltage to IC1. Running the computer for a few hours every week will keep the battery charged, and should you go on holiday the timer will retain its data. A fully charged NiCad should be capable of running the MM58174 for three months.

CONSTRUCTION

The integrated circuits are all CMOS devices so take extra care of them until the board is fully built. They should, of course, be the last items to go onto the board, and they must all go into IC sockets. The best way to bend the pins prior to pushing the chips into their sockets is to lay the IC on a sheet of kitchen-foil and exert the necessary pressure, having first removed any harmful static electricity by grounding yourself and the foil on the nearest water-pipe.

A PCB foil pattern is given for this project, so you shouldn't have any difficulties with the board itself, providing that you purchase the specified components. What you will have difficulty in making or obtaining is a 20-pin header to

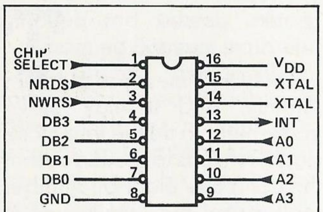


Fig. 3 The pin designations of the MM58174.

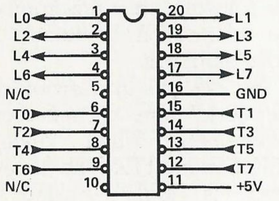


Fig. 4 Pin connections of the MTX parallel I/O port (port 07).

go into the 20-pin I/O socket on the MTX PCB. The socket is a standard 20-pin IC socket but it seems there isn't much call for 20-pin headers. I would suggest making one by splitting a 24-pin header into two 10-pin lengths. A short length of ribbon cable connects across to the Real Time board which is small enough to fit somewhere in the case of an unexpanded MTX.

Alternatively, it is but a simple matter to pass the ribbon cable out through the convenient gaps at the rear of the MTX case, in which case the timer can be housed in a tiny box Blu-Taked to the computer. The pin-outs of the MTX I/O socket are given in the user manual, but in case you have mislaid that, they are reproduced here as Fig. 4. Note that the +5 V pin on the MTX I/O port is only guaranteed to deliver 20 mA, so don't be tempted to replace IC2, IC3 and IC4 with TTL chips with their higher supply current requirements.

SOFTWARE

To read just one register of the clock/calendar from BASIC is a fiddle and a waste of time since for virtually the same length of code the whole set can be read at once. A numeric array, V(), accepts the data from the clock and a subroutine does the work: see Listing 1.

The numbers sent to the port on the OUT command are made up as follows:

R is a number between 0 and 15 representing one of the 16 possible registers in IC1.

16 is added whenever a high impedance output from IC3 is required.

32 is added whenever a WRITE to IC1 is *not* required.

64 is added whenever the ALE signal is required.

128 is added whenever a READ to IC1 *is* required.

Because of the hardwired leap-year indicator, the data received by INPut 7 will either be in the range 0-15 or 128-143. The code at line 8062 corrects this, and also sets V(13) to zero if it isn't a leap year.

Note the check for valid date at line 8070. If the data is 15, then the read occurred during one of the timer's update cycles and another reading must be taken immediately, line 8080.

For computers other than the Memotech, you will have to change INP to IN and the port values L and T must have port numbers appropriate to the machine. If IN and OUT commands don't exist, the port facilities may be mapped into memory and so a POKE will replace OUT and a PEEK will replace IN.

All that remains to be done now is to set the timer chip to GMT. For this a WRITE subroutine is required. Variable R passes the register number, and D passes the data, to the subroutine given in Listing 2. A simple loading routine can then be used to set the timer. See Listing 3.

```
REM -- READ THE TIME
GOSUB 8000
PRINT V(7);V(6);":";V(5);V(4)
REM HOURS AND MINUTES DISPLAYED


 100 DIM V(13): LET L=7: LET T=7
8000 REM - TIME READING SUBROUTINE
8005 LET V(13)=0          :REM - LEAP YEAR FLAG ZEROED
8010 FOR R=1 TO 12        :REM - 12 TIME REGISTERS
8020 OUT L,R+16+32        :REM - OFFER ADDRESS TO IC4
8030 OUT L,R+16+32+64 :REM - LATCH ADDRESS INTO IC4
8040 OUT L,16+32+128  :REM - READ IC1
8050 LET V(R)=INP(T)  :REM - ACCEPT DATA FROM IC1
8060 OUT L,16+32          :REM - FINISH READ
8061 REM - STRIP AWAY BIT 7, SETTING LEAP YEAR FLAG
8062 IF V(R)>15 THEN  LET V(R)=V(R)-128:LET V(13)=1
8070 IF V(R)<15 THEN  GOTO 8090 :REM - DATA VALID ?
8080 OUT L,16+32+128 : LET V(R)=INP(T) : OUT L,16+32
8085 IF V(R)>15 THEN LET V(R)=V(R)-128
8090 NEXT R
8095 RETURN
```

Listing 1. Subroutine for reading the time.

```
8100 OUT L,R+16+32        :REM - OFFER ADDRESS TO IC4
8110 OUT L,R+16+32+64 :REM - LATCH ADDRESS INTO IC4
8120 OUT L,D+16+32+64 :REM - PRESENT DATA TO IC3
8130 OUT L,D+32           :REM - PASS DATA
8140 OUT L,D              :REM - WRITE DATA
8150 OUT L,D+32           :REM - FINISH WRITE
8160 OUT L,D+16+32        :REM - IC3 TRISTATE AGAIN
8170 RETURN
```

Listing 2. Subroutine to write to the timer.

```
10 LET R=14:LET D=0:GOSUB 8100:REM - STOP THE TIMER
20 FOR R=4 TO 14
30 PRINT "REGISTER ";R
40 INPUT "VALUE -- ";D
50 GOSUB 8100
60 NEXT R
70 STOP
```

Listing 3. A loader program using the subroutine in Listing 2 to load the GMT values.

The response to Register 14 should be 1, the start-timer code, and the carriage-return you make after that starts the clock "on the pips" if you're listening to TIM on the 'phone. Seconds are automatically zeroed whenever register 14 is used, so in theory total accuracy can be achieved. However, don't necessarily expect your timer to be 100 % accurate from the word go. You must be prepared to test it over a period of a few days and to tweak the trimmer if necessary, although I should think any software using the clock would accept an accuracy of ±60 seconds a week.

The whole board is best left outside the Memotech case for a few days until you are happy with the clock's timekeeping. When it's OK, I would suggest you cut the link on the WRITE line, and pop the unit inside the computer case or in its own little plastic box. The link on input line D6 should not be made unless you are going to experiment with interrupts, in which case you will need machine-code driving routines rather than the simple BASIC routines presented here.

Finally, a reminder to all computer buffs — it's late evening on December 31st and the revellers are abroad in Trafalgar Square and Scottish programmes are on all television channels: *what must you do to link LK2 on your real-time board?* Answers please on a postcard . . .

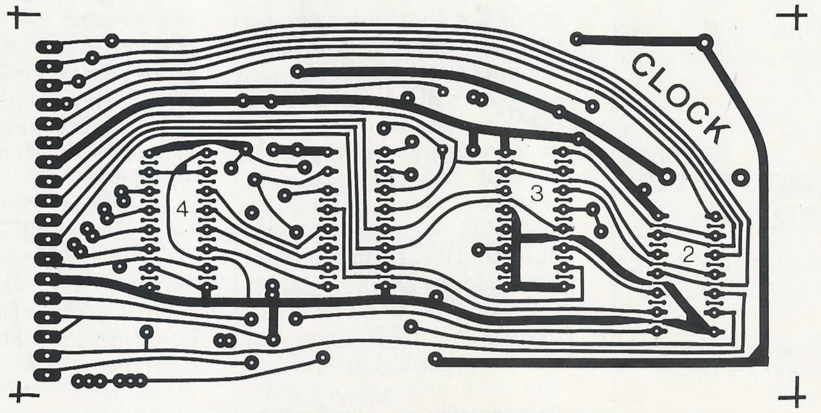


Fig. 5 Actual size PCB foil pattern.

TM

AMERICAN EXPRESS
Cards Welcome

Software News

TM

BARCLAYCARD
VISA
Buy it with Access

INNOVATIVE SOFTWARE

from the professionals

First
there was **Tandy TRS-80**™

And Then
came **Genie**

Followed
by **BBC**

Penultimately
we added **IBM**

And Now **SANYO**

From the original Tandy Model I in 1978, Molimerx now supply software for fifteen different machines and close to fifty different configurations. A 180 page Catalogue is available for the Tandy and Genie ranges. We have to charge £3.00 for this, although £1.00 of this is for postage and is refundable on the first order. For the BBC we have a 28 page Catalogue which is available for a stamped addressed A4 size envelope for 17 pence. For the IBM and Sanyo machines we have lists which are available free of charge.

Trade Mark Credits: As apparent from the manufacturer's name.

TEL: [0424] 220391/223636

TELEX 86736 SOTEX G

MOLIMERX™ LTD
A J HARDING (MOLIMERX)
1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

CARRY A COMMODORE

Grahame Davies

The Commodore SX-64 has now been available for several months — depending on who you listen to, it's either moderately successful or selling like hot cakes. Exactly who is buying the SX? Our reviewer looks at the Commodore with an identity crisis.

The SX64 consists of the popular Commodore 64 home computer, a five inch colour monitor and a 1541 disc drive all contained in one case. It falls into the transportable or 'luggable' category as it requires mains power — no battery facilities are available and none are planned (the monitor requires too much power).

With the SX housed in a smart grey and silver plastic case, the external appearance is very pleasing. Other leading portables are cased in metal which, although providing greater strength, especially when moving the computer, is not strictly necessary and also far heavier than the plastic used by Commodore. Plastic does have other advantages such as absorbing vibration, as well as being far cheaper!

The back of the SX is a huge (metal) heatsink and the machine feels and looks more than strong enough to endure the sort of battering that any computer should (or should not) go through.

Incidentally, a salesman from Kaypro (one of the leading portable business computers in this country) told me that the vast majority of his customers buy a machine, install it and never move it! Apparently, it is the compactness of the complete system which appeals to them rather than the portability.

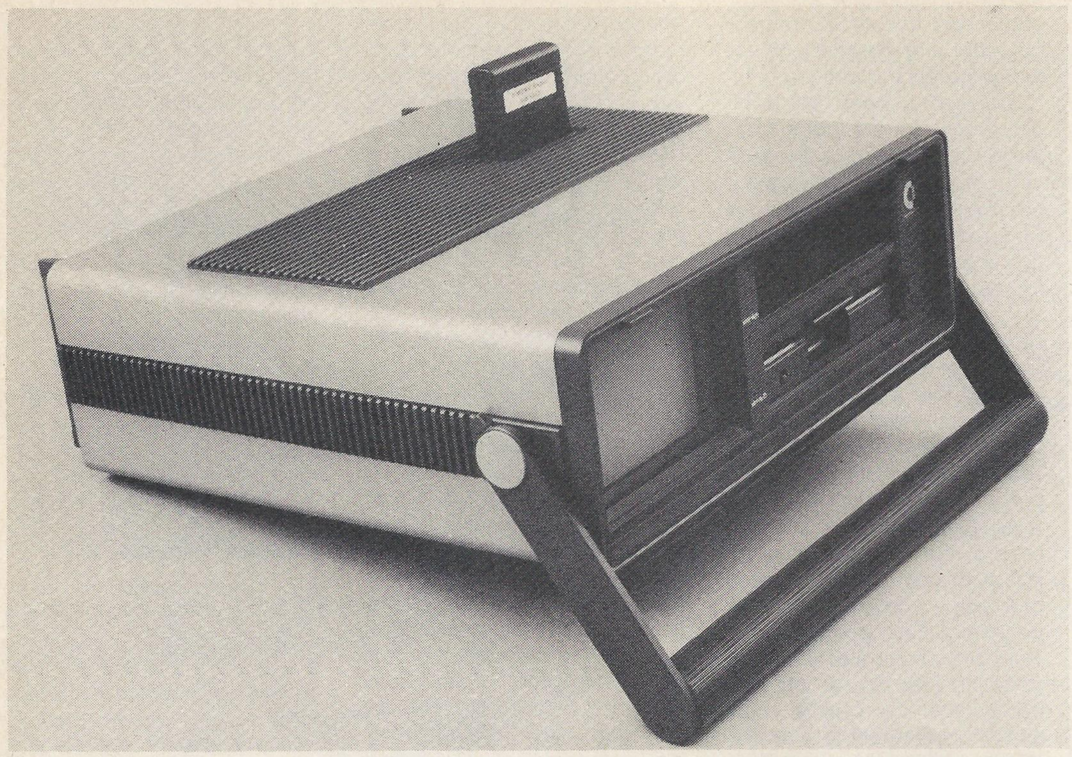
Hidden behind a small flap to the right hand front of the machine are controls for the volume, contrast, brightness, colour and vertical hold, all giving a good range of adjustment. There is also a disc drive reset switch located here but it is of limited use. The drive reports an error by flashing its single red LED at an irritating rate; the reset switch does at least stop that. However, if a read error occurs while loading a program, pressing this switch will reset the drive but leave the SX hung up. The spring eject mechanism on my model has broken — sadly, a common falt with this drive and one which should have been rectified by now.

The monitor is located next to the storage space and the disc drive. Presumably there is no problem with static from the screen, but I would not rely on this and would store my discs well away from the machine even though there is a small shield behind the monitor and the disc drive itself. The SX may be connected to an external monitor directly or to your television set via an RF converter which should cost just a few pounds. The internal speaker provides plenty of sound for normal use but will distort badly at the higher volume levels. Alternatively, you can connect your SX to your hi-fi equipment. The cartridge slot is located on the top of the unit, hidden behind a pair of sprung loaded flaps which help to keep foreign objects out and also make locating the cartridge far easier. Anyone who has tried to push a cartridge on to the back of a Commodore 64 will appreciate this feature.

INSIDE STORY

Opening the SX involves undoing several screws and removing slithers of plastic: it's certainly not easy, but that can hardly be an important requirement for a computer in this price range. What's inside is more surprising. Having designed the 64 on one board, it seems strange that the SX has been split up, but this must have been necessary due to the dimensions of the case. This in itself does not matter so much, but the mass of connecting wires and more importantly the connectors on the end of those wires may well lead to unreliability. I can only say that this is a little disappointing, especially in the light of the latest business machines from Commodore. One of these, the 8296, is the neatest and best produced machine yet to be released by Commodore. If you have seen inside an 8032 with a 64K upgrade board and all its wires and connectors, then looking inside the 8296 becomes a delight. This makes comparisons with the SX all the more disappointing.

Typically, all the advertising for this machine, especially the photographs, generally give the impression of a one-box



machine, implying you can just remove the plug and cart it away. This is actually misleading as Commodore also supply a separate purse in which to carry the mains lead and cartridges and so on that is essential to the average user. There is a little space above the disc drive which is labelled 'storage' but you could not get a simple thing like a joystick or a box of floppies in it.

The SX weighs in at a shade under 25 pounds (plus the extra purse). Very reasonable, you might think; but carry it any distance, especially up and down stairs, and the ridiculous tubular handle with its specially designed painful grooves soon starts to dig into the palm of your hand. It's amazing that it was ever produced with this handle. At only five inches wide, the SX is easy to get through doors or walk through crowded corridors — the overall dimensions are 14.5 by 14.5 by 5 inches.

CARRY ON COMPUTING

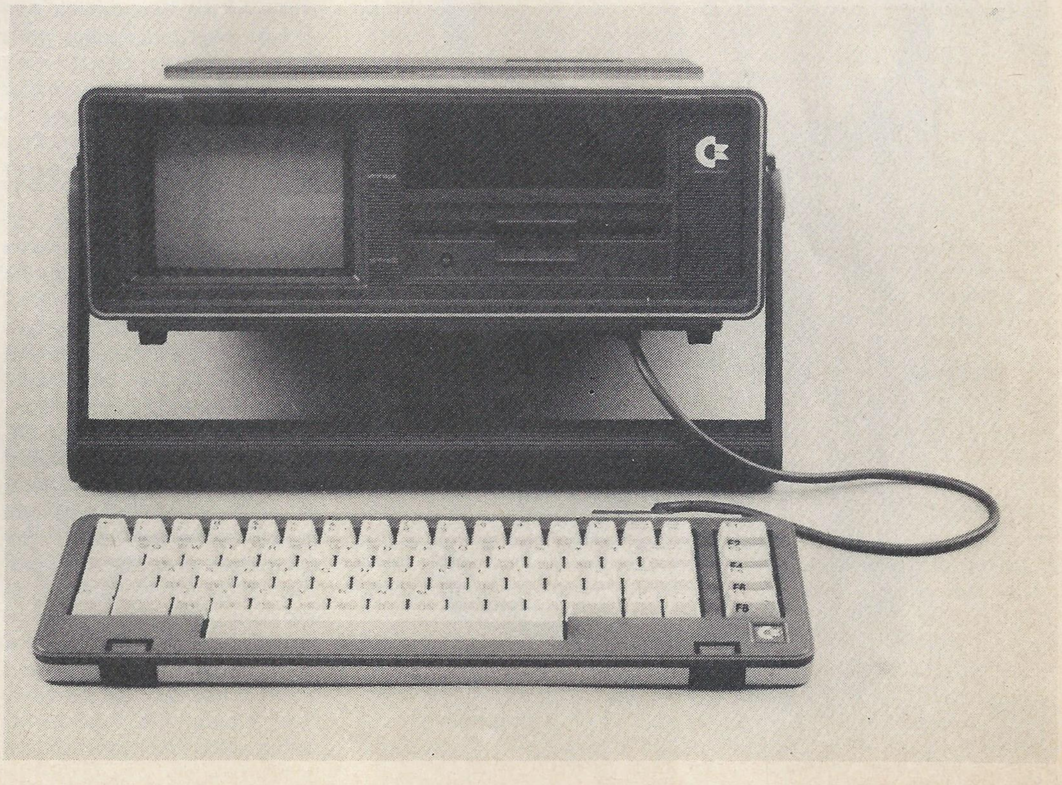
To test its portability, I took the SX to an exhibition. It was held inside a leisure centre and wasn't associated with computers — so there was not an abundance of mains power points available. This meant I had to take a mains extension cable along with all the other accessories, so that neat little purse went out the window to be replaced by a larger bag. Having found a table and a power point, plugged in the power cable and the keyboard connector, we were up and running. If you were doing the same with a standard home computer you could easily spend three times as long plugging in all the cables, use three times the number of power points and end up with a tangle of messy cables — not forgetting that in twenty minutes you may have to move because you have just blocked a fire exit or similar (as happened to me). The idea of a portable starts to look quite promising.

The handle is a multi-positioned device with horrible fiddly locking devices on either side. This means that the angle and the height of the machine may be adjusted, enabling the keyboard to be positioned as near to the computer (or even under) as is required — cutting down on the desk top space used. The keyboard is in the cap of the computer and is held on by two rather fragile-looking, but in fact quite secure catches at the top. The bottom is just slotted in so that the keyboard is not connected to the body when in use.

The keyboard lead is of adequate length — a bit too short for every conceivable use, perhaps, but too much cable can be as much of a nuisance as too little. It plugs in firmly to the SX and to the keyboard itself — a step forward from the Commodore SK business machines which have a very poor and loose connector to the computer.

The keyboard can, in fact, be refitted to the body without unplugging the cable, although if you now want to move the SX a short distance, releasing the locks to straighten the handle means the latter now hits the keyboard connector that you left in for convenience. Incidentally, there is no warning about plugging the keyboard connector in and out while the machine is running, but I don't recommend doing so as on certain models of the business machines, this has caused a problem with the monitor.

The power cable plugs in to the back of the SX but with the straight-in connector supplied



with my machine, this meant that you could not stand it on its end. Even if an angled plug was supplied, the other leads that plug into the back such as the joysticks and serial cables would get squashed and most probably damaged.

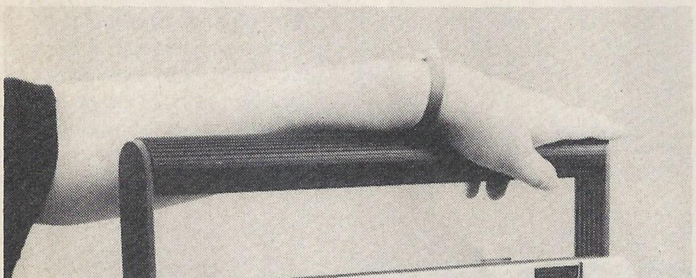
THE USER'S VIEW

Having established what the SX is like to live with, let's see if it is your sort of machine. First of all, it has no cassette port and the ROMs have been altered to return messages such as 'Illegal Device Number' if you try to access the tape channels. But then, who needs a cassette port on a business computer? Next question, then: is the SX a business computer? Ask Commodore and you will get a very non-committal reply. After all, they want to sell it to anybody who will buy it. The glossy sales pamphlet accompanying the machine shows it in all sorts of environments — in the office, in the home, on a building site — on a building site? There is man standing with a metal hat and his SX by his side — it's a good job that he forgot to take his power cable and keyboard connector with him as there is no mains supply in sight!

So, having been given no clue so far, we are further confused by the name on the front of the computer — 'Executive'. Maybe it just hints at being a business computer? Assuming that it *is* a business computer for the moment, let's take a look at what it has to offer, starting with the disc drive. This is the good old (bad old?) Commodore 1541 which has been compared in speed to certain cassette systems and snails. To be fair, the main problem with the speed of the 1541 is the serial bus connecting it to the computer. It is not suited to any database but is certainly convenient for individual files, program files, word processing files, financial planning files and so on. The capacity of the disc is 165K and it is compatible with the 1540 and 4040 disc drives.

The other thing about the drive is that there is only one of them — something which is very rare in true business systems, be they portable or not.

In my opinion, twin drives are virtually essential: they provide ease of backing up, greater reliability (one drive may be able to read important data whereas the other one may not, due to mis-alignment of the heads perhaps) and far better flexibility. There is a twin drive version of the SX available in the USA called the DX, but there has been no hint to date from Commodore as to the likelihood of it being available over here.



The colour monitor is very good, especially if you have been used to running a computer on an old television set. However, being only five inches, it has the obvious drawback of being tiring to use. You could, of course, plug the SX into a RGB monitor but then it loses its portable appeal. The other problem with the screen is that it is still the standard 40 columns, the same as the Commodore 64. This is an important drawback as all business machines (including portables) recognise the need for 80 columns. Eighty column adapters are available for the 64, but whether they could fit in the cartridge slot on the SX is another question, and a separate monitor would then be essential as 80 columns on the standard screen does not bear thinking about. Still on the business side of things, the SX has the same user ports and serial ports as the 64: thus printers, plotters, disc crives, modems and so on may be added if required. The only other port is the joystick port — who needs a joystick on a business computer? Its only use in a business sense is that of allowing protection keys to be fitted, although some protection keys (see below) require the cassette port.

The keyboard, it seems, has been designed to be as thin as is practical. It is, in fact, the same size as the 64's keyboard except that the gap between the function keys and the main keys has been reduced and although it contains the same keys as the 64, they do not travel as far and seem less positive. The keys are white, with black and light brown lettering and the characters are easy to read. The graphic characters are still shown (unnecessary with a business computer) but are in a light brown colour and only on the front of the keys, thus leaving the tops nicely uncluttered. The letters on the top of the keys are offset to the top left corner, perhaps so that they are not easily obscured by dirty fingers (when you are 'out in the field').

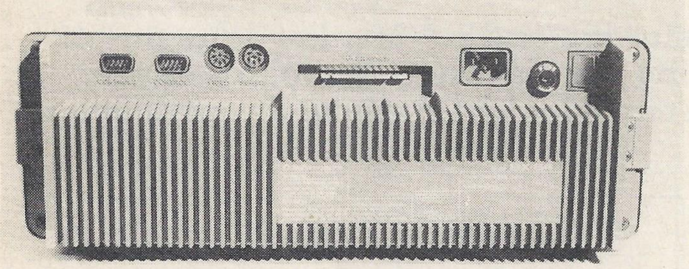
The keyboard layout is consistent with business purposes, with letters, numbers and mathematical signs easily accessable (they do not have to be 'shifted'). The one glaring omission, though, is that of a numeric keypad. A numeric keypad is essential to virtually all business uses and without one, entering data is not only slow and laborious but also leads to a drastic increase in the number of errors made.

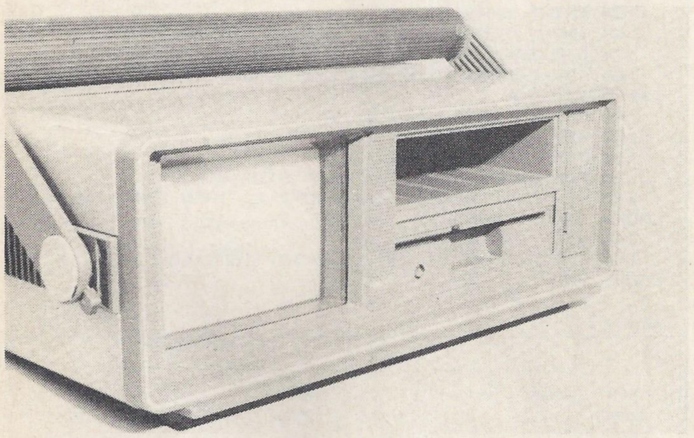
SOFTWARE

All non-cassette based software for the 64 should run perfectly well on the SX, so business software is immediately available. In fact, Commodore supply business software valued at £210 with the SX — they sometimes have a strange sense of values. It's at this time that I thought I could benefit from one of the few definite and positive changes from the 64. On the 64, pressing SHIFT/RUN causes the first program from the cassette to be loaded. On the SX, the same action causes the first program from the disc to be LOADed (LOAD":*", 8). If you are familiar with Commodore BASIC, you may or may not know that the above statement loads the first program from the disc and if the colon was omitted, the next program on the directory following the last file accessed, would be loaded. So I put in one of the business discs supplied and pressed away — and got a syntax error in the program.

In fact, all of the software supplied either produced a syntax error when using this command or just hung up the machine. The software actually requires the same command but with ',1' added, which relocates the machine code being loaded to the address given by the first two bytes of the file. Therefore, the change to the machine was totally useles, being suited only for BASIC files. This is doubly ridiculous as BASIC files will still load if the ',1' is added — there' s nothing like being given half of a good thing!

It would seem then, that the





SX is a long way from being a particularly competent business machine. However, it can hardly be directed at the games and home market. The vast majority of games software is still only available on cassette and as already stated, the machine does not support cassettes.

The SX may complement an existing Commodore business system nicely and also is a good home computer. Its portability allows the entering and editing of data with the minimum of fuss, allowing the same data to be transferred (from home to office for instance) by storing it on the floppy disc . As with the 64, it may be used in conjunction with a modem but this is unlikely in a business application, being far more useful for home users using one of the up and coming networks such as Compunet.

Another point to note is that the location of the cartridge slot means that a lot of the extra hardware available for it (aside from standard games cartridges) may well not fit in and so some sort of adaptor will have to be found. The most likely buyers of the machine would seem to be the wealthy home user — at £895 it is not a cheap machine although there are rumours of a dramatic price drop being circulated. If this does happen, I think the SX will sell extremely well.

VIVE LA DIFFERENCE

For those of you who are familiar with the 64, I will go through the few differences between the operating system on that and the SX. It is widely felt that Commodore missed a great opportunity (deliberately or otherwise) to improve on the 64. Factors that would have been useful to improve include getting rid of the serial bus in exchange for IEEE 488 and expanding the number of BASIC commands to include hi-res and sound commands and so on to the standard of BBC BASIC.

On powering up the machine, the monitor takes a few seconds to warm up, and the first difference found is that the default colours have altered to sensible, readable and nicely contrasting colours. The background colour is white, the border colour is light green and the character colour is black. Also the words 'SX-64' appear on the screen.

FACTSHEET	Commodore SX-64
CPU	6510
RAM	64K
ROM	20K
Language	BASIC v2
Keyboard	62 keys (QWERTY) plus four function keys (Shift gives eight functions)
Display	Text: 24 lines of 40 characters Graphics: 320 by 200 pixel resolution in high-res mode; programmable, multicolour graphics; eight sprites; block graphics from keyboard in text mode.
Sound	Three channels with four waveforms, plus programmable filters and envelopes
Cassette	None
Disc	Integral 5¼" floppy disc
I/O	Parallel user port, serial interface, two joystick ports, cartridge port, monitor/sound port
Bundled Software	Easy Script, Easy File and Future Finance
Size	14½" by 14½" by 5"
Weight	25 lbs

As mentioned earlier, pressing SHIFT/RUN loads the first program from the disc. The BASIC used is still BASIC 2, which is a great shame because an upgrade to BASIC 4 with its associated disc handling commands would make the machine far nicer to work with. The new home computers from Commodore (the 16 and the Plus/4) will have BASIC 3.5 which has a lot of extra and useful commands.

Other features of the SX are unchanged — there is the SID chip (sound interface device) which has three channels, four waveforms, envelopes and filters and is perhaps the most extensive sound generator on the market. There are the eight sprites (or MOBs — movable object blocks), multicolour graphics and a resolution of 320 by 200 pixels in hi-res mode. The CPU is the 6510 (an updated 6502 still using 6502 operands), there is 64K of RAM (38K for BASIC programs and variable/string storage) and 16K of ROM available through the cartridge slot.

The lack of cassette port means that you can't use any program requiring a protection key to be fitted to this port (such as the popular DTL compiler). A program requiring a protection key on one of the joystick ports should run perfectly well. However, there appears to be some confusion about this regarding the other major compiler, Petspeed. I have heard from different (reliable) sources that Commodore say that Petspeed will not run on the SX due to timing differences on the joystick ports. I have been told by others that Petspeed has been run without any trouble!

In an effort to establish the true situation, I ran this compiler on the SX but was forced to give up due to read errors from the disc drive and then a lack of time. This was inconclusive as on the 8032 computers, this compiler will crash anywhere between pass 1 and pass 3 if there is no protection key. Therefore, the only safe thing to assume is that it is unlikely to run. Of course, programs that have been compiled and do not require a protection key, will run perfectly well on the SX.

CONCLUSIONS

To summarise, the SX is a very elegant machine. It is nicely finished and tastefully coloured, making a very neatly packaged home computer with reasonable portability. As far as business is concerned, it has the same potential as a standard 64 (which is after all just a home computer) and I cannot see it making much of an impact on the business market.

ALL SORTS OF SORT

M. Vivian

In any system designed to handle the storage and retrieval of data, one of the more useful abilities is that of sorting the information alphabetically. There are slow sorts and fast sorts, here we look at a mixed assortment.

Sorting is a very common data processing technique. A file must be sorted from physical(order of entry) state into some logical order (usually alphabetic) to allow for searches on the file, respectable looking print-outs and so on. But what kind of sort do you use for the particular application that you are considering? An analysis of a small number of alphanumeric sorts follows, and this will hopefully help you to make your mind up, should you face such a decision.

BUBBLE SORT

This is perhaps the most common type of sort in existence, primarily because of its simplicity. As the name suggests, data items are 'bubbled' up to their correct ordered places in the array concerned, by means of comparisons with neighbouring items. The array is passed over a finite number of times until the entire array can be designated 'sorted', ie each data item has been bubbled to its correct place.

This is perhaps better explained with the use of diagrams. Consider the array shown in Fig 1. Listing 1 is an example program.

For those mathematicians amongst you, the program makes n-1 factorial comparisons and this should be taken into consideration when looking at the other sorts in this selection.

Being one of the simplest sorts in existence, the Bubble Sort is also one of the slowest: its best asset probably being that it is most effective when just one record needs to be placed in a previously sorted file. The new record can be placed at the end of the file and then just bubbled up until it reaches its proper place (ie no swap needs to take place with the item immediately above).

RIPPLE SORT

This is the same as the Bubble Sort in principle, except that instead of 'bubbling' items up the array or file, the Ripple Sort 'ripples' items down.

With this program, however, I have introduced a flag called MARKER, which tells us that one or more exchanges have taken place on any single pass of the array, if set. If an exchange has not been made, the variable MARKER will not be set and the array must be sorted. Thus, an early exit from program execution can be made, reducing the run time. Taking our original example, we can see how the Ripple Sort works and appreciate the use of this flag (see Fig 2).

In this case, the early exit didn't make much difference (one comparison) but on larger files, this could make an appreciable difference in the time taken to sort, especially if the file were to be partially sorted (ie just one record out of place) as the maximum of n-1 factorial comparisons is reduced. If, however, all comparisons are made, this sort is fractionally slower than the Bubble Sort, as time is taken assigning and testing MARKER within the nested loops.

SELECTION SORT

The Selection Sort is slightly superior as a sorting technique over those previously mentioned. As vefore, a number of passes of the array in question are made and these get progressively smaller as the sort proceeds. However, instead of comparing the item in question with the one above it, it is compared with that item currently at

```
    10  REM ** BUBBLE SORT
    20  REM ** ENTER DATA
    30 OPEN "R",#1,"RANDOM.DAT",16
    40 FIELD #1,16 AS I$
    50 PRINT CHR$(12)
    60 INPUT "ENTER NUMBER OF DATA ITEMS ";NUM
    70 DIM IMF$(NUM-1)
    80 FOR LOOP=1 TO NUM: GET#1,LOOP: IMF$(LOOP-1)=
I$: NEXT LOOP
    90 PRINT "DATA LOADED..."
   100 REM ** SORT
   110 FOR LOOP=1 TO NUM-1
   120 FOR LOOP2=NUM-1 TO LOOP STEP -1
   130 IF IMF$(LOOP2)<IMF$(LOOP2-1) THEN SWAP IMF$(
LOOP2),IMF$(LOOP2-1)
   140 NEXT LOOP2,LOOP
   150 FOR LOOP=0 TO NUM-1: PRINT IMF$(LOOP): NEXT
LOOP: END
```

Listing 1

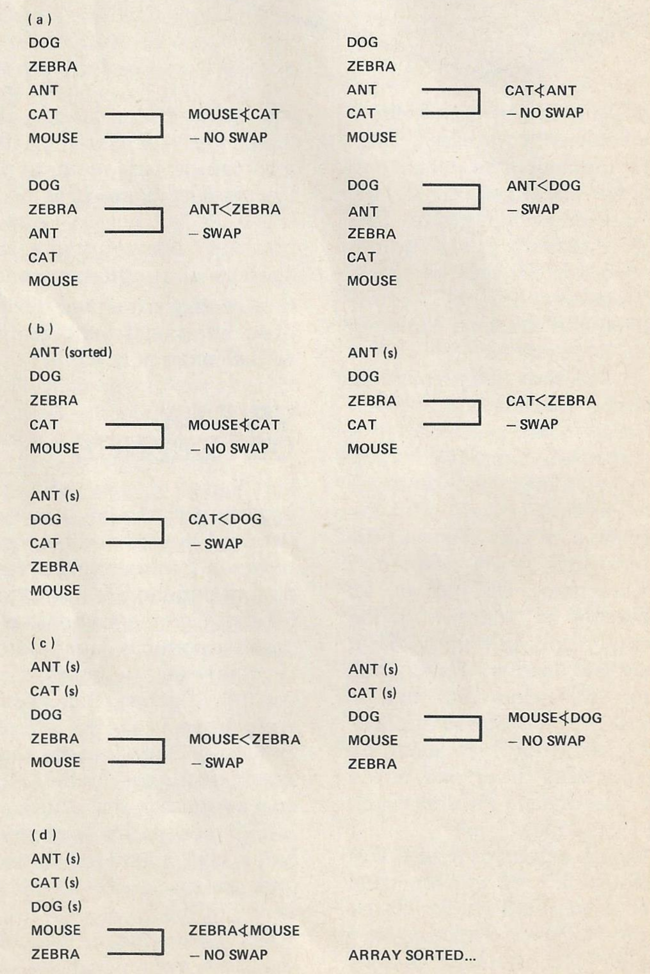


Fig 1 Bubble Sort

```
    10 REM ** RIPPLE SORT
    20 REM ** ENTER DATA
    30 OPEN "R",#1,"RANDOM.DAT",16
    40 FIELD #1,16 AS I$
    50 PRINT CHR$(12)
    60 INPUT "ENTER NUMBER OF DATA ITEMS ";NUM
    70 DIM IMF$(NUM-1)
    80 FOR LOOP=1 TO NUM: GET#1,LOOP: IMF$(LOOP-1)=
 I$: NEXT LOOP
    90 PRINT "DATA LOADED..."
   100 REM ** SORT
   110 FOR LOOP=0 TO NUM-2
   120 MARKER=0
   130 FOR LOOP2=0 TO NUM-LOOP-2
   140 IF IMF$(LOOP2)>IMF$(LOOP2+1) THEN SWAP IMF$(
 LOOP2),IMF$(LOOP2+1): MARKER=1
   150 NEXT LOOP2
   160 IF MARKER=0 THEN GOTO 180
   170 NEXT LOOP
   180 REM ** OUTPUT DATA ARRAY
   190 FOR LOOP=0 TO NUM-1: PRINT IMF$(LOOP): NEXT
 LOOP: END
```

Listing 2

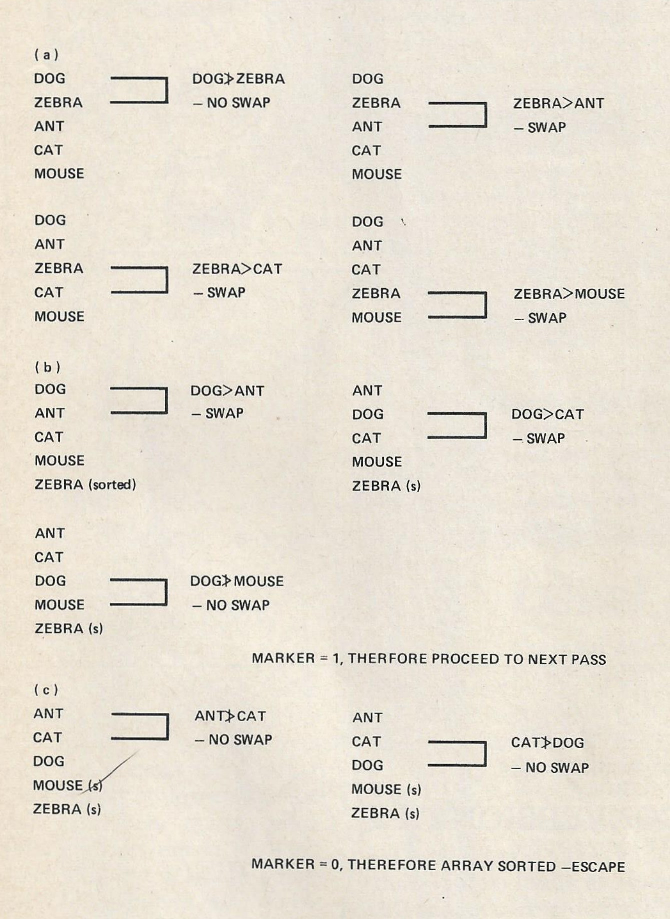


Fig 2 Ripple Sort

```
    10 REM ** SELECTION SORT
    20 REM ** ENTER DATA
    30 OPEN "R",#1,"RANDOM.DAT",16
    40 FIELD #1,16 AS I$
    50 PRINT CHR$(12)
    60 INPUT "ENTER NUMBER OF DATA ITEMS ";NUM
    70 DIM IMF$(NUM-1)
    80 FOR LOOP=1 TO NUM: GET#1,LOOP: IMF$(LOOP-1)=
 I$: NEXT LOOP
    90 PRINT "DATA LOADED..."
   100 REM ** SORT
   110 FOR LOOP=0 TO NUM-2
   120 FOR LOOP2=LOOP+1 TO NUM-1
   130 IF IMF$(LOOP2)<IMF$(LOOP) THEN SWAP IMF$(LOO
 P2),IMF$(LOOP)
   140 NEXT LOOP2,LOOP
   150 REM ** OUTPUT DATA ARRAY
   160 FOR LOOP=0 TO NUM-1: PRINT IMF$(LOOP): NEXT
 LOOP: END
```

Listing 3

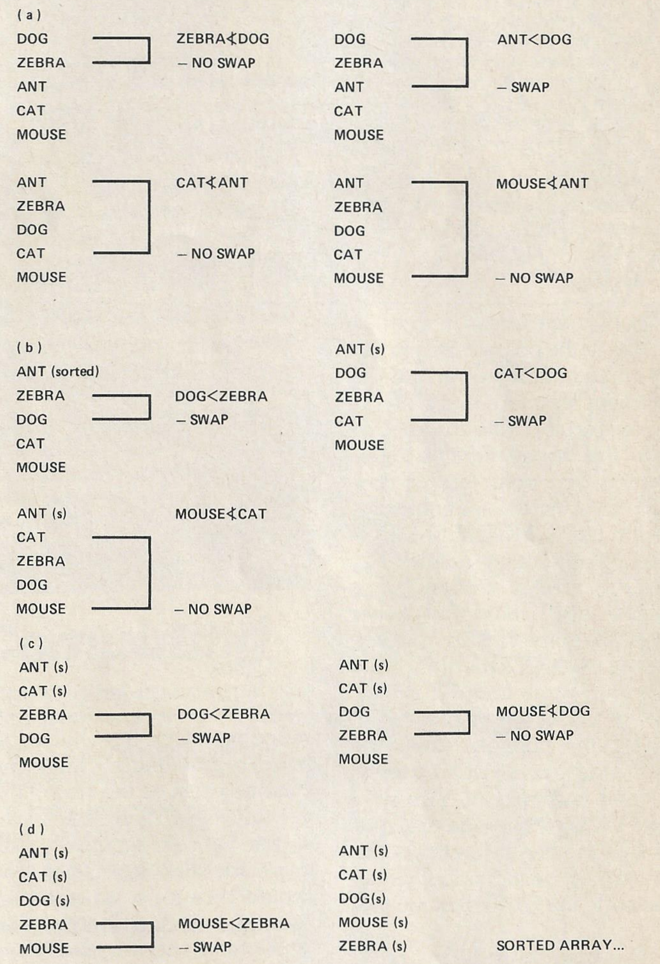


Fig 3 Selection Sort

the top of the list on that particular pass and exchanged with it if necessary. In this way, the name 'Selection Sort' is appropriate to the way in which it works, as the top item on each pass is selected from those below (and including) it. This can be demonstrated with reference to our miniature array as in Fig 3.

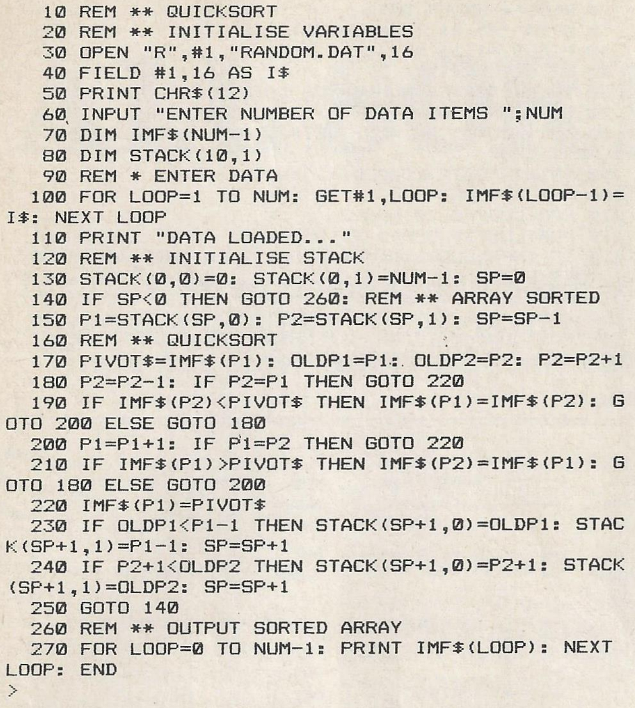
In this way, the same number of comparisons are required as the Bubble Sort but on the whole, less actual exchanges have to be named. In this way the overall time taken by the Selection Sort is less than that taken by previously described methods, for the same number of records.

QUICK OR BINARY SORT

The previous three methods of sorting items in an array of data relied on the same basic principles: two nested loops, which had the effect of causing the program to pass over the data a finite number of times, the result at the end of each pass being one element that was in its correct place and could therefore be discounted from further passes. This continued until there were no more items left to consider.

The Quick Sort relies on a different concept from these previous methods. The first item in the array is taken as a 'starting block' (or pivot) with which to compare further items. All items that are less than this pivot are placed above it and all that are greater than it are placed below it. In this way, the record that was originally first has found its place in the array and the other records are on the correct side of this. The whole procedure is then repeated for the items above and below this pivot, considering each as a separate entity and for the subsets within them, until a single item is left in a subset, whereupon it can be dismissed. When there are no subsets left to consider, the array is then sorted. Figure 4 will best explain this method.

You will notice the use of pointers in this diagram: one marks the current record, whilst the other marks the available



```
  10 REM ** QUICKSORT
  20 REM ** INITIALISE VARIABLES
  30 OPEN "R",#1,"RANDOM.DAT",16
  40 FIELD #1,16 AS I$
  50 PRINT CHR$(12)
  60 INPUT "ENTER NUMBER OF DATA ITEMS ";NUM
  70 DIM IMF$(NUM-1)
  80 DIM STACK(10,1)
  90 REM * ENTER DATA
 100 FOR LOOP=1 TO NUM: GET#1,LOOP: IMF$(LOOP-1)=
I$: NEXT LOOP
 110 PRINT "DATA LOADED..."
 120 REM ** INITIALISE STACK
 130 STACK(0,0)=0: STACK(0,1)=NUM-1: SP=0
 140 IF SP<0 THEN GOTO 260: REM ** ARRAY SORTED
 150 P1=STACK(SP,0): P2=STACK(SP,1): SP=SP-1
 160 REM ** QUICKSORT
 170 PIVOT$=IMF$(P1): OLDP1=P1: OLDP2=P2: P2=P2+1
 180 P2=P2-1: IF P2=P1 THEN GOTO 220
 190 IF IMF$(P2)<PIVOT$ THEN IMF$(P1)=IMF$(P2): G
OTO 200 ELSE GOTO 180
 200 P1=P1+1: IF P1=P2 THEN GOTO 220
 210 IF IMF$(P1)>PIVOT$ THEN IMF$(P2)=IMF$(P1): G
OTO 180 ELSE GOTO 200
 220 IMF$(P1)=PIVOT$
 230 IF OLDP1<P1-1 THEN STACK(SP+1,0)=OLDP1: STAC
K(SP+1,1)=P1-1: SP=SP+1
 240 IF P2+1<OLDP2 THEN STACK(SP+1,0)=P2+1: STACK
(SP+1,1)=OLDP2: SP=SP+1
 250 GOTO 140
 260 REM ** OUTPUT SORTED ARRAY
 270 FOR LOOP=0 TO NUM-1: PRINT IMF$(LOOP): NEXT
LOOP: END
>
```

Listing 4

```
(a) DOG
DOG –P1        CAT –P1        CAT            CAT
ZEBRA          ZEBRA          ZEBRA –P1      ANT
ANT            ANT            ANT            ANT –P1 + P2 = DOG
CAT            CAT –P2        ZEBRA –P2      ZEBRA
MOUSE –P2      MOUSE          MOUSE          MOUSE

MOUSE≮DOG      ZEBRA>DOG      ANT<DOG
CAT<DOG

(b) CAT
CAT –P1        ANT                 –SUBSET ONE RECORD LONG –SORTED
ANT –P2        ANT –P1 + P2 = CAT

ANT<CAT

(c) ZEBRA
ZEBRA –P1      MOUSE                  –SUBSET ONE RECORD LONG –SORTED
MOUSE –P2      MOUSE –P1 + P2 = ZEBRA

MOUSE<ZEBRA

ANT   }
CAT   } 2
DOG     1
MOUSE }
ZEBRA } 3
```

Fig 4 Quick or Selection Sort

space in the array. In the first instance, the pointers are placed at either end of the set and the element pointed to by P1 is stored in a separate variable, so that this space can be overwritten. The comparisons now commence at the element pointed to by P2: MOUSE < DOG, hence detrement pointer P2, CAT < DOG, hence place ZEBRA at P1, now work down from this point. ZEBRA > DOG, hence place ZEBRA at P2. Now work upwards from P2 again, ANT < DOG, hence place ANT at P1. the pointers have now met and the pivot, DOG, can be placed at this point. Two subsets are now in evidence, one above and one below this point: these must be individually subjected to the Quick Sort routine.

This method is clearly more involved than those considered previously, as the size of the program (Listing 4) indicates.

More memory is required for this method of sorting, not only because of the relative size of the program but also because it uses more variables: a stack is employed to store the parameters of the subsets that have yet to be sorted (remember that only one subset can be considered at once).

However, the overriding factor is the speed at which this sort operates (see timings below). Wheras the first three sorts increase exponentially in the time they take as the number of items increases, the Quick Sort takes approximately the same time per item as the size of the list increases. With this knowledge, we can accurately estimate the time the Quick Sort would take for a list of a previously untested length. This type of sort is at its most effective on a totally randomly ordered file but is likely to be somewhat slower on a partially ordered file.

NB All of the sorts in this selection have the ability to handle duplications of any particular item in the array being sorted.

TABLE 1

	NUMBER OF RECORDS						
TYPE OF SORT	25	50	75	100	150	200	250
Bubble	3.9	15.5	35.5	1:03.1	2:19.0	4:04.2	6:19.3
Ripple	3.7	15.0	38.1	1:07.0	2:30.4	4:23.2	6:55.1
Selection	3.3	13.2	29.9	52.1	1:58.0	3:25.0	5:21.0
Quick	2.1	5.1	8.7	10.9	19.0	26.7	33.9

TIMINGS

The programs shown above were written on the Superbrain QD in MBASIC. Each program uses a common data file called 'RANDOM.DAT', which contains 16 byte records created at random using the RND function. The timings in Table 1 are those taken by the program mentioned to sort increasing numbers of records from this file. All timings in Minutes: Seconds. Tenths of seconds.

CONVERSION TIPS

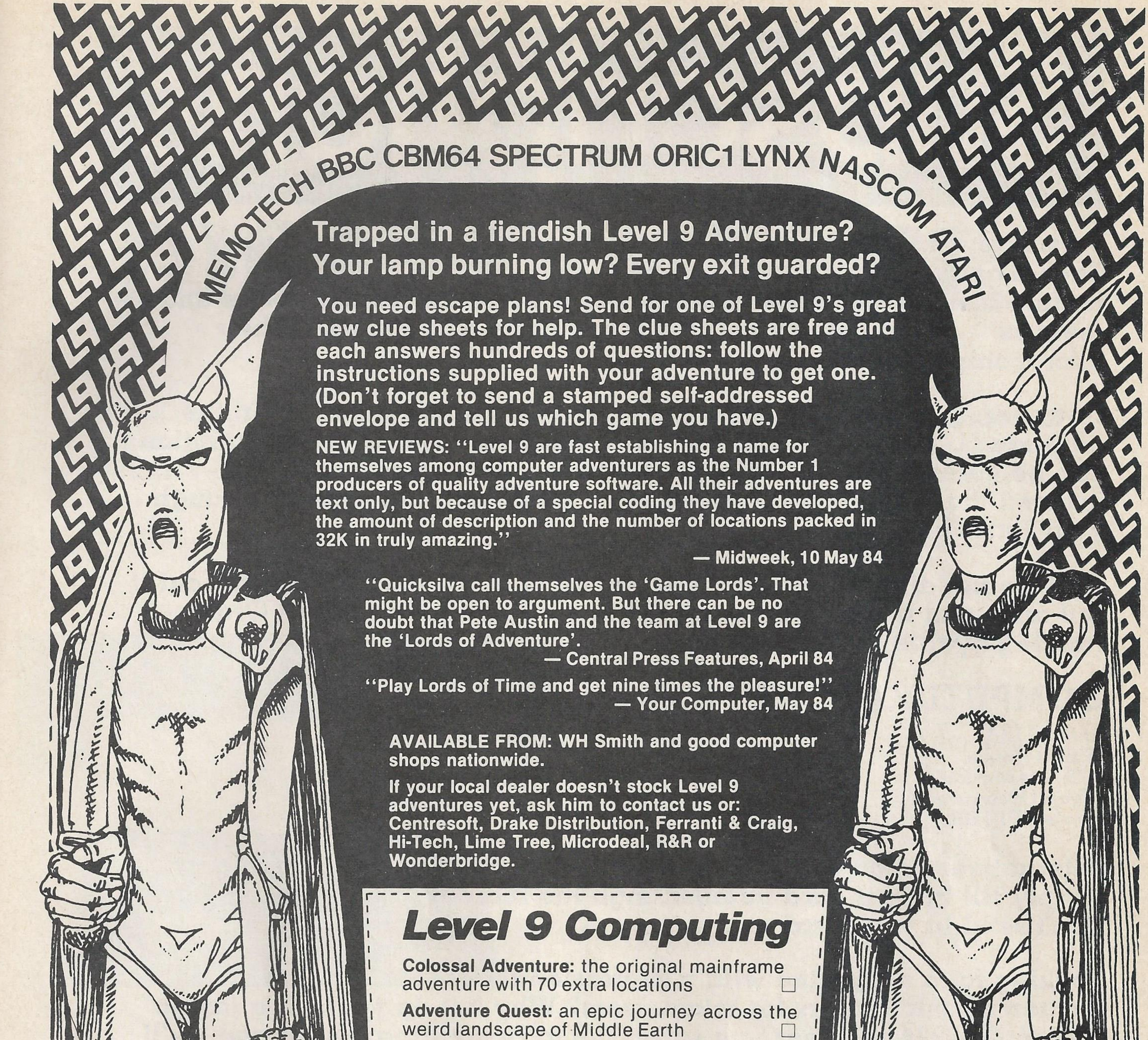
It is important to note that in the programs listed, all array subscripts start at zero, hence DIM IMF$(10) reserves eleven places for IMF$. All zero elements have been used in order to save memory and it is important to allow for this when using machines that start subscripts from one.

The other major point concerns the use of the SWAP statement, which exchanges the contents of the two variables which follow it. On machines wthout this statement, SWAP IMF$(X),IMF$(Y) can be substituted with:

TEMP$ = IMF$(X):IMF$(X) = IMF$(Y):IMF$(Y) = TEMP$.

SUMMARY OF USAGE

Bubble sort	● partially sorted files ● particularly small files ● conditions where memory capacity doesn't permit more elaborate methods
Ripple sort	● as above
Selection sort	● small/medium files ● conditions where memory capacity doesn't permit more elaborate methods
Quick sort	● mainly medium/large files ● preferably not partially sorted



MEMOTECH BBC CBM64 SPECTRUM ORIC1 LYNX NASCOM ATARI

Trapped in a fiendish Level 9 Adventure? Your lamp burning low? Every exit guarded?

You need escape plans! Send for one of Level 9's great new clue sheets for help. The clue sheets are free and each answers hundreds of questions: follow the instructions supplied with your adventure to get one. (Don't forget to send a stamped self-addressed envelope and tell us which game you have.)

NEW REVIEWS: "Level 9 are fast establishing a name for themselves among computer adventurers as the Number 1 producers of quality adventure software. All their adventures are text only, but because of a special coding they have developed, the amount of description and the number of locations packed in 32K in truly amazing."

— Midweek, 10 May 84

"Quicksilva call themselves the 'Game Lords'. That might be open to argument. But there can be no doubt that Pete Austin and the team at Level 9 are the 'Lords of Adventure'.

— Central Press Features, April 84

"Play Lords of Time and get nine times the pleasure!"

— Your Computer, May 84

AVAILABLE FROM: WH Smith and good computer shops nationwide.

If your local dealer doesn't stock Level 9 adventures yet, ask him to contact us or: Centresoft, Drake Distribution, Ferranti & Craig, Hi-Tech, Lime Tree, Microdeal, R&R or Wonderbridge.

Level 9 Computing

Colossal Adventure: the original mainframe adventure with 70 extra locations □

Adventure Quest: an epic journey across the weird landscape of Middle Earth □

Dungeon Adventure: a truly massive game which completes the Middle Earth Trilogy □

Snowball: could this be the biggest SF adventure in the world? □

Lords of Time: an imaginative romp through World History □

I ENCLOSE A CHEQUE/PO FOR £9.90 EACH

My name: ....

My address: ....

....

....

My micro is a: ....
(one of those listed on the arch above, with at least 32K of memory).

Contact:
LEVEL 9 COMPUTING
Dept. C, 229, Hughenden Road
High Wycombe, Bucks. HP13 5PG
Telephone: (0494) 26871

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, software listings, news and reviews — all to help committed micro users make more of their microcomputers at home or at work.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,

Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1984 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
COMPUTING TODAY Subscriptions
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

Please commence my subscription to Computing Today with the .... issue.

SUBSCRIPTION RATES (tick □ as appropriate)	£13.90 for 12 issues UK	□
	£17.55 for 12 issues Overseas Surface	□
	£37.20 for 12 issues Overseas Airmail	□

I am enclosing my (delete as necessary)
cheque/ Postal Order/ International Money
Order for £..........
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard*
(*delete as necessary)

□□□□□□□□□□□□□□□□□□

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) ....
delete accordingly

ADDRESS ....

....

.... POSTCODE ....

Signature ....

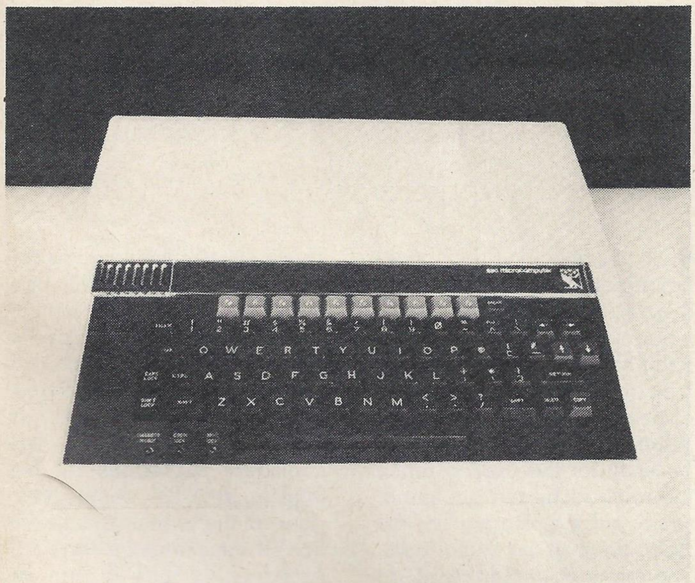
Date ....

CT Aug '84

BEEB DISASSEMBLER

Wayne Henderson

In the April 1984 issue, CT presented an article by Don Thomasson on Disassembly Techniques with examples based on the BBC operating system version 1.2. That was the theory . . . This program gives you the necessary tool to put that theory into practice.



In Disassembly Techniques, Don Thomasson specified the necessary tools for disassembly as:

- A disassembly routine to display both Hexadecimal and Alphabetical interpretations of data with an optional printing facility.
- A sorting program to sort the accumulated list of entry points into the correct sequence.

While I can't help on the second point, I have managed to produce a disassembler specifically for the Beeb which does a little more than the normal 'run of the mill' disassembler. This program not only disassembles the machine code in ROM but undoes the actions the interpreter performs when storing a BASIC program.

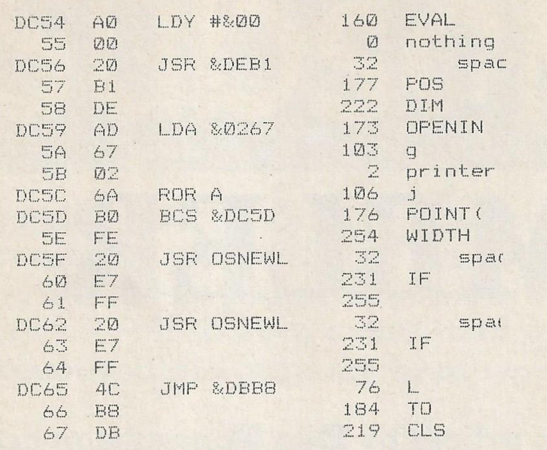
I hope this program will not only be a useful utility but also give programmers a better idea of what happens to their BASIC programs as they are typed in.

OBJECTIVES

When I started to write this program I had the following objectives in mind:

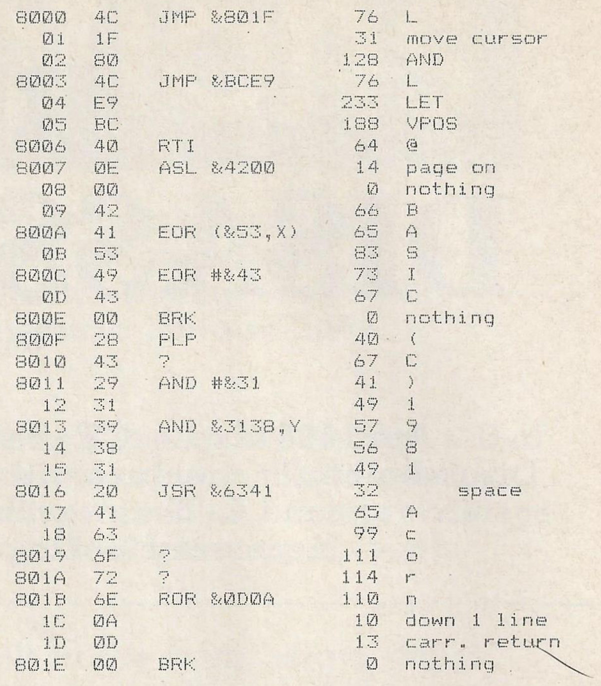
```
15E9  ØD   ORA &8AØ2       13  carr. return
  EA  Ø2                    2  (LINE NO.)
  EB  8A                  138  ( 65Ø      )
15EC  2B   ?               43  43   bytes
15ED  F7   ?              247  RESTORE
15EE  2Ø   JSR &448D       32      space
  EF  8D                  141  coded line
  FØ  44                   68  (          )
15F1  7C   ?              124  ( 166Ø     )
15F2  46   LSR &3A         7Ø  (          )
  F3  3A                   58  :
15F4  E3   ?              227  FOR
15F5  2Ø   JSR &3D58       32      space
  F6  58                   88  X
  F7  3D                   61  =
15F8  31   AND (&B8),Y     49  1
  F9  B8                  184  TO
15FA  31   AND (&39),Y     49  1
  FB  39                   57  9
15FC  3A   ?               58  :
15FD  F3   ?              243  READ
15FE  2Ø   JSR &2452       32      space
  FF  52                   82  R
  ØØ  24                   36  $
16Ø1  2C   BIT &2552       44  ,
  Ø2  52                   82  R
  Ø3  25                   37  %
16Ø4  3A   ?               58  :
16Ø5  E7   ?              231  IF
16Ø6  2Ø   JSR &2552       32      space
  Ø7  52                   82  R
  Ø8  25                   37  %
16Ø9  3D   AND &5Ø4F,X     61  =
  ØA  4F                   79  O
  ØB  5Ø                   8Ø  P
16ØC  25   AND &2Ø         37  %
  ØD  2Ø                   32      space
16ØE  4F   ?               79  O
16ØF  5Ø   BVC &1635       8Ø  P
  1Ø  24                   36  $
1611  3D   AND &2452,X     61  =
  12  52                   82  R
  13  24                   36  $
1614  ØD   ORA &94Ø2       13  carr. return
  15  Ø2                    2  (LINE NO.)
  16  94                  148  ( 66Ø      )
1617  Ø7   ?                7  7    bytes
1618  ED   SBC &E13A      237  NEXT
  19  3A                   58  :
  1A  E1                  225  ENDPROC
```

Example 1 How coded lien numbers and tokenised keywords are stored in user memory.



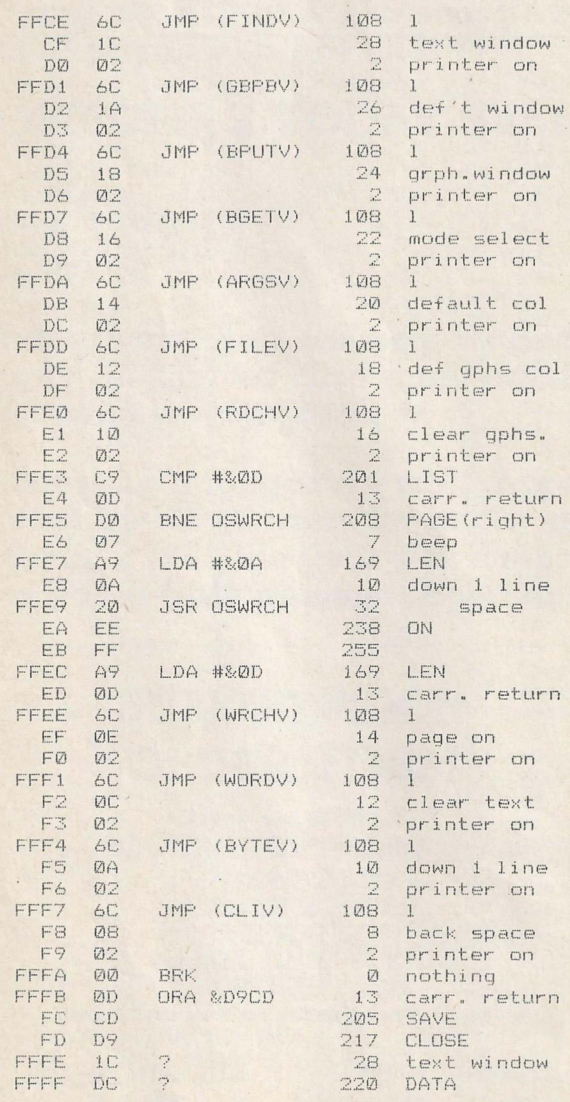
```
DC54  AØ   LDY #&ØØ        160  EVAL
  55  ØØ                     Ø  nothing
DC56  2Ø   JSR &DEB1        32      spac
  57  B1                   177  POS
  58  DE                   222  DIM
DC59  AD   LDA &Ø267       173  OPENIN
  5A  67                   1Ø3  g
  5B  Ø2                     2  printer
DC5C  6A   ROR A           1Ø6  j
DC5D  BØ   BCS &DC5D       176  POINT(
  5E  FE                   254  WIDTH
DC5F  2Ø   JSR OSNEWL       32      spa(
  6Ø  E7                   231  IF
  61  FF                   255
DC62  2Ø   JSR OSNEWL       32      spa(
  63  E7                   231  IF
  64  FF                   255
DC65  4C   JMP &DBB8        76  L
  66  B8                   184  TO
  67  DB                   219  CLS
```

Example 2 One of the routines explained in Disassembly Techniques by Don Thomasson.



```
8ØØØ  4C   JMP &8Ø1F        76  L
  Ø1  1F                    31  move cursor
  Ø2  8Ø                   128  AND
8ØØ3  4C   JMP &BCE9        76  L
  Ø4  E9                   233  LET
  Ø5  BC                   188  VPOS
8ØØ6  4Ø   RTI              64  @
8ØØ7  ØE   ASL &42ØØ        14  page on
  Ø8  ØØ                     Ø  nothing
  Ø9  42                    66  B
8ØØA  41   EOR (&53,X)      65  A
  ØB  53                    83  S
8ØØC  49   EOR #&43         73  I
  ØD  43                    67  C
8ØØE  ØØ   BRK               Ø  nothing
8ØØF  28   PLP              4Ø  (
8Ø1Ø  43   ?                67  C
8Ø11  29   AND #&31         41  )
  12  31                    49  1
8Ø13  39   AND &3138,Y      57  9
  14  38                    56  8
  15  31                    49  1
8Ø16  2Ø   JSR &6341        32      space
  17  41                    65  A
  18  63                    99  c
8Ø19  6F   ?               111  o
8Ø1A  72   ?               114  r
8Ø1B  6E   ROR &ØDØA       11Ø  n
  1C  ØA                    1Ø  down 1 line
  1D  ØD                    13  carr. return
8Ø1E  ØØ   BRK               Ø  nothing
```

Example 4 Acorn's copyright message at the start of the Interpreter.



```
FFCE  6C   JMP (FINDV)     1Ø8  l
  CF  1C                    28  text window
  DØ  Ø2                     2  printer on
FFD1  6C   JMP (GBPBV)     1Ø8  l
  D2  1A                    26  def't window
  D3  Ø2                     2  printer on
FFD4  6C   JMP (BPUTV)     1Ø8  l
  D5  18                    24  grph.window
  D6  Ø2                     2  printer on
FFD7  6C   JMP (BGETV)     1Ø8  l
  D8  16                    22  mode select
  D9  Ø2                     2  printer on
FFDA  6C   JMP (ARGSV)     1Ø8  l
  DB  14                    2Ø  default col
  DC  Ø2                     2  printer on
FFDD  6C   JMP (FILEV)     1Ø8  l
  DE  12                    18  def gphs col
  DF  Ø2                     2  printer on
FFEØ  6C   JMP (RDCHV)     1Ø8  l
  E1  1Ø                    16  clear gphs.
  E2  Ø2                     2  printer on
FFE3  C9   CMP #&ØD        2Ø1  LIST
  E4  ØD                    13  carr. return
FFE5  DØ   BNE OSWRCH      2Ø8  PAGE(right)
  E6  Ø7                     7  beep
FFE7  A9   LDA #&ØA        169  LEN
  E8  ØA                    1Ø  down 1 line
FFE9  2Ø   JSR OSWRCH       32      space
  EA  EE                   238  ON
  EB  FF                   255
FFEC  A9   LDA #&ØD        169  LEN
  ED  ØD                    13  carr. return
FFEE  6C   JMP (WRCHV)     1Ø8  l
  EF  ØE                    14  page on
  FØ  Ø2                     2  printer on
FFF1  6C   JMP (WORDV)     1Ø8  l
  F2  ØC                    12  clear text
  F3  Ø2                     2  printer on
FFF4  6C   JMP (BYTEV)     1Ø8  l
  F5  ØA                    1Ø  down 1 line
  F6  Ø2                     2  printer on
FFF7  6C   JMP (CLIV)      1Ø8  l
  F8  Ø8                     8  back space
  F9  Ø2                     2  printer on
FFFA  ØØ   BRK               Ø  nothing
FFFB  ØD   ORA &D9CD        13  carr. return
  FC  CD                   2Ø5  SAVE
  FD  D9                   217  CLOSE
FFFE  1C   ?                28  text window
FFFF  DC   ?               22Ø  DATA
```

Example 3 Part of the top of the Operating System ROM showing the vectored jumps to service routines.

● In addition to the normal hexadecimal and mnemonic display, it had to provide the name of Operating System routines and vectors, from a look up table, whenever these appeared as an operand. When this was done it would be easy for the user to name the routines he himself found, while poking around in the ROM area, in the same table. The program would then do some of the work of identifying where jumps and branch instructions were going to.

● The program had to cope not only with ASCII characters but also with:

1 The Tables of data so that the disassembler did not slip out of sync when it came to the end of a table. There really is no way round this except to display each location on a single line — if an instruction is wrong then the user will have enough space to alter it.

2 The tokenised keywords which are used by the interpreter.

3 The special way that the Beeb stores program line numbers.

These last two would allow the disassembly of BASIC programs in their stored format. As the program is unable to distinguish between different types of data, I decided to display the information in as many forms as possible. The intelligent user does all the thinking and the dumb micro does all the tedious manipulation. If your machine isn't too user friendly then it may be the other way round!

● Adding a bit of colour. This not only looks good but is functional as it helps to distinguish between fields of data. Mode 7 seemed appropriate as I only needed to display text.

● Adding a printing facility (which presented a slight problem as my Epson printer did not like teletext control codes, but more about that later).

● Incorporating a certain amount of user control over the display, ie single step, scrolled, paged and so on while the program was running.

● Allowing the use of pseudo variables like PAGE as input to the 'start address' and 'end address' prompts.

TABLE 1

Bits —	7	6	5	4	3	2	1	0
Byte 1	0	1	128s	$\overline{64}$s	0	$\overline{16384}$s	0	0
Byte 2	0	1	32s	16s	8s	4s	2s	1s
Byte 3	0	1	8192s	4096s	2048s	1024s	512s	256s

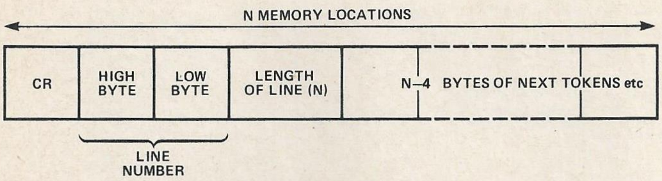


Fig 1 How the memory locations are stored in a line of n length.

```
806A  4C   JMP &8A80        76  L
  6B  80                   128  AND
  6C  8A                   138  TAB(
806D  41   EOR (&4E,X)      65  A
  6E  4E                    78  N
806F. 44   ?                68  D
8070  80   ?               128  AND
8071  00   BRK               0  nothing
8072  41   EOR (&42,X)      65  A
  73  42                    66  B
8074  53   ?                83  S
8075  94   STY &00,X       148  ABS
  76  00                     0  nothing
8077  41   EOR (&43,X)      65  A
  78  43                    67  C
8079  53   ?                83  S
807A  95   STA &00,X       149  ACS
  7B  00                     0  nothing
807C  41   EOR (&44,X)      65  A
  7D  44                    68  D
807E  56   LSR &41,X        86  V
  7F  41                    65  A
8080  4C   JMP &96          76  L
  81  96                   150  ADVAL
  82  00                     0  nothing
8083  41   EOR (&53,X)      65  A
  84  53                    83  S
8085  43   ?                67  C
8086  97   ?               151  ASC
8087  00   BRK               0  nothing
8088  41   EOR (&53,X)      65  A
  89  53                    83  S
808A  4E   LSR &98          78  N
  8B  98                   152  ASN
  8C  00                     0  nothing
808D  41   EOR (&54,X)      65  A
  8E  54                    84  T
808F  4E   LSR &99          78  N
  90  99                   153  ATN
  91  00                     0  nothing
8092  41   EOR (&55,X)      65  A
  93  55                    85  U
8094  54   ?                84  T
8095  4F   ?                79  O
8096  C6   DEC &10         198  AUTO
  97  10                    16  clear gphs.
```

Example 5 The start of the interpreter's look up table of keywords and their tokens.

TOKENS AND BASIC STORAGE

Whenever a line of BASIC is typed into the BBC machine and the RETURN key is pressed, the resident interpreter changes the line into a more condensed format to save user memory space. The line is stored in the format shown in Fig 1.

Any keywords that are present in the line are changed into hexadecimal codes called tokens (see pages 483-484 of the User Guide for a full list of tokens). For example, every time the keyword PRINT appears in a program it is represented by the byte &F1. To make things more complex, all line numbers after GOTO, GOSUB and RESTORE are stored in a special way. The line number is replaced with byte 141 (or &8D in hex) and three bytes of code to represent the value of the line number. These bytes can be calculated from Table 1. The bits with a bar above their value are zero if the line number does contain their value and one if it does not. This type of approach speeds up the execution of GOTO and RESTORE and so on and is one of the reasons why BBC BASIC is so fast.

A more detailed explanation of these principles and of the way in which the Beeb stores variables can be found in either **The BBC Micro Revealed** by Jeremy Rushton or **The BBC Micro : An Expert Guide** by Mike James. Both books contain a lot of useful information but Mike James does not cover the topic of coded line numbers and Jeremy Rushton has made a slight mistake in describing the format of BASIC lines. His book describes the line number following the carriage return as being in the order LSB MSB, when in fact it is the other way round. We could also mention here that the Acom User Guide is not infallible. There is a mistake in the 6502 instruction set on page 508. It lists JMP (indirect addressing) as &8C when it should be &6C; the code &8C represents STY (absolute addressing).

USING THE PROGRAM

When running the program you will be presented with the prompt 'start address'. You may enter the address in decimal or in hexadecimal if it is prefixed by '&'. It will also accept the psuedo variables PAGE, LOMEN, HIMEN, BASIC (&8000) and MOS (&C000). If you press RETURN instead then the start address will default to &0000. You will then be asked for an 'end address'. The same rules apply but the default is now &FFFF. The program will not accept an end address which is less than the start address. You will then be asked if you require a hard copy, reply with a Y or N answer. When the program is running, the following keys will produce the effects listed below.

N	No heading
H	Restores heading
R	Reduces height of heading
E	Enlarges height of heading
C	Makes program scroll continuously
S	Makes program scroll single step
P	Puts program in paged mode (use shift keys to move on)

The program will stop when it reaches the specified end address or the Escape key is pressed. You will be asked if you want another disassembly — N will end the program and Y will take you back to the start address prompt with all format preserved. In this way, you can wander round the memory without ever having to restart the program.

FEATURES

● Notice how error handling is used at the start of the program to trap the errors generated by EVAL when the user gives silly inputs. It also is used to inform the user of any possible programming errors at line 1890 yet doesn't allow

```
 10 REM DISASSEMBLER / DETOKENISER
 20 REM Copyright (C) Wayne Henderson
 30 REM written FEB. 1984
 40 *TV 255,0
 50 MODE7
   :PROCdriver
   :Format$="H"
   :Scroll$="P"
   :VDU14
   :CLS
 60 ON ERROR GOTO 1880
 70 VDU133
   :INPUT"  Start address ",Loc$
 80 IF Loc$="PAGE" L%=PAGE
     :GOTO160
 90 IF Loc$="LOMEM" L%=LOMEM
     :GOTO160
100 IF Loc$="HIMEM" L%=HIMEM
     :GOTO160
110 IF Loc$="BASIC" L%=&8000
     :GOTO160
120 IF Loc$="MOS" L%=&C000
     :GOTO160
130 IF Loc$="" L%=0
     :GOTO160
140 ON ERROR GOTO 60
150 L%=EVAL Loc$
160 ON ERROR GOTO 1880
170 IF L%<0 OR L%>&FFFF THEN 60
180 VDU134
   :INPUT"  End   address ",End$
190 IF End$="PAGE" End%=PAGE
     :GOTO270
200 IF End$="LOMEM" End%=LOMEM
     :GOTO270
210 IF End$="HIMEM" End%=HIMEM
     :GOTO270
220 IF End$="BASIC" End%=&8000
     :GOTO270
230 IF End$="MOS" End%=&C000
     :GOTO270
240 IF End$="" End%=&FFFF
     :GOTO270
250 ON ERROR GOTO 160
260 End%=EVAL End$
270 ON ERROR GOTO 1880
280 IF End%<0 OR End%>&FFFF THEN 160
290 IF End%<L% GOTO60
         ELSE NC%=L%
        :Ln%=L%-4
        :GT%=L%-4
300 PRINT
   :VDU130
   :PRINT"  Hard copy 'Y' or 'N' ? ";
310 A$=GET$
   :IF A$<>"Y" AND A$<>"N" THEN 310
         ELSE PRINTA$
        :IF A$="Y" CALL&ADC
          :VDU2
          :printout=1
             ELSE printout=0
320 PRINT'
   :REPEAT
     :PROCuser
330   IF Format$="H" AND L% MOD 32=0 PROCheading
340   IF Scroll$="S" ANY$=GET$
350   PROClocation
     :PROChex
     :PROCmnemonic
     :PROCdec
     :PROCascii_basic
     :PROCupdate
     :UNTIL NC%>&FFFF OR L%> End%
360 PRINT'
   :VDU130
   :PRINT"  END OF DISASSEMBLY"
370 PRINT
   :CALL&AE9
   :printout=0
   :VDU3,131
   :PRINT"ANOTHER DISASSEMBLY 'Y' or 'N' ? ";
380 *FX21,0
390 A$=GET$
   :IF A$<>"Y" AND A$<>"N" GOTO380
400 IF A$="Y" PRINT'
     :GOTO60
         ELSE PRINT
        :ON ERROR OFF
410 END
420 DEF PROCmnemonic
430 MN$="   "
   :OP$=""
   :LE%=1
   :IF NC%<>L% VDU 131
     :PRINTSPC(13);
     :ENDPROC
440 IF H1%=&FF MN$="?"
     :MC%=1
         ELSE RESTORE 1480
        :FOR X=0TOH1%
          :READ MN$,MC%
          :NEXT
450 RESTORE 1680
   :FOR X=1 TO MC%
     :READ LE%
     :NEXT
   :PROCoperand
   :PROCroutine
   :PROCvector
   :PROCsymbols
   :VDU131
   :PRINT" ";MN$;" ";OP$;SPC(22-POS);
   :ENDPROC
460 DEF PROCupdate
470 IF L%=NC% NC%=L%+LE%
480 IF H1%=&0D AND L%>Ln%+3 AND L%<>&DFF PROCline_no
     :Ln%=L%+1
490 IF H1%=&8D AND L%>GT%+3 PROCG
     :GT%=L%+1
500 L%=L%+1
   :ENDPROC
510 DEF PROCline no
520 LIN%=H2%*256+H3%
   :L1$="(LINE NO.)"
   :IF H2%=&FF L1$="PROGRAM END"
530 L2$="(."+STR$(LIN%)+STRING$(7-LEN(STR$(LIN%))," ")+")"
   :L3$=STR$(H4%)
   :L3$=L3$+STRING$(4-LEN(L3$)," ")+" bytes"
   :ENDPROC
540 DEF PROCoperand
550 IF LE%=1 ENDPROC
560 IF LE%=2 OP%=H2%
570 IF LE%=3 OP%=H2%+(H3%*256)
580 IF MC%<>11 THEN 600
590 IF H2%>127 OP%=L%+H2%-254
         ELSE OP%=L%+H2%+2
600 OP$=STR$~(OP%)
   :IF LEN(OP$)=3 OR LEN(OP$)=1 OP$="0"+OP$
610 OP$="&"+OP$
   :ENDPROC
620 DEF PROCroutine
630 IF LE%=1 ENDPROC
640 IF OP$="" OR OP%<&FFB9 ENDPROC
```

```
650 RESTORE 1660
   :FOR X=1TO19
     :READ R$,R%
     :IF R%=OP% OP$=R$
660    NEXT
   :ENDPROC
670 DEF PROCvector
680 IF LE%=1 ENDPROC
690 IF OP$="" OR OP%<&200 OR OP%>&234 ENDPROC
700 RESTORE 1640
   :FOR X=1TO27
     :READ R$,R%
     :IF R%=OP% OP$=R$
710    NEXT
   :ENDPROC
720 DEF PROCsymbols
730 IF MC%=1 OR MC%=3 OR MC%=4 OR MC%=11 ENDPROC
740 IF MC%=2 OP$="A"
     :ENDPROC
750 IF MC%=5 OP$="#"+OP$
     :ENDPROC
760 IF MC%=6 OR MC%=10 OP$=OP$+",X"
     :ENDPROC
770 IF MC%=7 OR MC%=13 OP$=OP$+",Y"
     :ENDPROC
780 IF MC%=8 OP$="("+OP$+",X)"
     :ENDPROC
790 IF MC%=9 OP$="("+OP$+"),Y"
     :ENDPROC
800 IF MC%=12 OP$="("+OP$+")"
     :ENDPROC
810 DEF PROClocation
820 H1%=?L%
   :H2%=?(L%+1)
   :H3%=?(L%+2)
   :H4%=?(L%+3)
   :IF L%=NC% THEN 850
830 L$=STR$~(L% MOD 256)
   :IF LEN(L$)<2 L$="  0"+L$
        ELSE L$="  "+L$
840 VDU133
   :PRINTL$;
   :ENDPROC
850 L$=STR$~(L%)
860 IF LEN(L$)=1 L$="000"+L$
870 IF LEN(L$)=2 L$="00"+L$
880 IF LEN(L$)=3 L$="0"+L$
890 VDU133
   :PRINTL$;
   :ENDPROC
900 DEF PROCdec
910 Dec$=STR$(H1%)
   :Dec$=STRING$(3-LEN(Dec$)," ")+Dec$
   :VDU130
   :PRINTDec$;
   :ENDPROC
920 DEF PROCascii_basic
930 Bas$=""
   :IF L%>=PAGE AND L%<LOMEM AND (L%<Ln%+3 OR L%<GT%+3) PROCline_string
     :VDU129
     :PRINT:Bas$
     :ENDPROC
940 IF H1%=127 VDU129
     :PRINT"delete"
     :ENDPROC
950 IF H1%=&FF VDU129
     :PRINT
     :ENDPROC
960 IF H1%=32 VDU135
     :PRINT;"    space"
     :ENDPROC
970 IF H1%>127 PROCkeyword
     :ENDPROC
980 IF H1%<32 PROCcontrol
     :ENDPROC
990 Bas$=CHR$(H1%)
   :VDU135
   :PRINTBas$
   :ENDPROC
1000 DEF PROChex
1010 Hex$=STR$~(H1%)
    :IF LEN(Hex$)<2 Hex$="0"+Hex$
1020 VDU 134
    :PRINTHex$;
    :ENDPROC
1030 DEF PROCkeyword
1040 RESTORE 1690
    :FOR X=&80 TO H1%
      :READ Bas$
      :NEXT
    :IF H1%=&8D VDU129
         ELSE VDU135
1050 PRINTBas$
    :ENDPROC
1060 DEF PROCcontrol
1070 REMRESTORE 1460

    :X=-1
    :REPEAT
    :READ Asc$
    :X=X+1
    :UNTIL X=H1%
1080 IF H1%<&10 RESTORE 1780
         ELSE RESTORE 1790
1090 X=-1
    :REPEAT
      :READ Bas$
      :X=X+1
      :UNTIL (H1%<&10 AND X=H1%) OR (H1%>&F AND X+&10=H1%)
1100 VDU129
    :PRINTBas$
    :ENDPROC
1110 DEF PROCG
1120 B1%=H2%-64
    :B2%=H3%-64
    :B3%=H4%-64
    :T%=0
1130 IF B1%>=32 T%=T%+128
      :B1%=B1%-32
1140 IF B2%>=32 T%=T%+32
      :B2%=B2%-32
1150 IF B3%>=32 T%=T%+8192
      :B3%=B3%-32
1160 IF B2%>=16 T%=T%+16
      :B2%=B2%-16
1170 IF B3%>=16 T%=T%+4096
      :B3%=B3%-16
1180 IF B1%<16 T%=T%+64
         ELSE B1%=B1%-16
1190 IF B2%>=8 T%=T%+8
      :B2%=B2%-8
1200 IF B3%>=8 T%=T%+2048
      :B3%=B3%-8
1210 IF B2%>=4 T%=T%+4
      :B2%=B2%-4
1220 IF B3%>=4 T%=T%+1024
      :B3%=B3%-4
1230 IF B1%<4 T%=T%+16384
         ELSE B1%=B1%-4
1240 IF B2%>=2 T%=T%+2
      :B2%=B2%-2
1250 IF B3%>=2 T%=T%+512
      :B3%=B3%-2
1260 IF B2%=1 T%=T%+1
```

```
1270 IF B3%=1 T%=T%+256
1280 L1$="(         )"
    :L2$="( "+STR$(T%)+STRING$(7-LEN(STR$(T%))," ")+")"
    :L3$="(         )"
    :ENDPROC
1290 DEF PROCline_string
1300 IF L%=Ln% OR L%=GT% Bas$=L1$
     :ENDPROC
1310 IF L%=Ln%+1 OR L%=GT%+1 Bas$=L2$
     :ENDPROC
1320 IF L%=Ln%+2 OR L%=GT%+2 Bas$=L3$
     :ENDPROC
        ELSE ENDPROC
1330 DEFPROCuser
1340 USER$=INKEY$(0)
    :IF USER$="P" VDU14
     :Scroll$=USER$
1350 IF USER$="C" OR USER$="S" VDU15
     :Scroll$=USER$
1360 IF USER$="H" OR USER$="N" Format$=USER$
1370 IF USER$="R" printout=1
1380 IF USER$="E" printout=0
1390 *FX 21,0
1400 ENDPROC
1410 DEF PROCheading
1420 IF printout=1 THEN 1450
1430 IF (L%<PAGE OR L%>TOP) FOR X=1TO2
        :VDU141,133
        :PRINT"Loc";
        :VDU 134
        :PRINT"Hex";
        :VDU 131
        :PRINT"Mnemonic     ";
        :VDU130
        :PRINT"Dec";
        :VDU135
        :PRINT"A";
        :VDU129
        :PRINT"S";
        :VDU135
        :PRINT"C";
        :VDU129
        :PRINT"I";
        :VDU135
        :PRINT"I"
        :NEXT
      :ENDPROC
1440 IF L%>=PAGE AND L%<=TOP FOR X=1TO2
        :VDU141,133
        :PRINT"Loc";
        :VDU 134
        :PRINT"Hex";
        :VDU 131
        :PRINT"Mnemonic     ";
        :VDU130
        :PRINT"Dec";
        :VDU135
        :PRINT"B";
        :VDU129
        :PRINT"A";
        :VDU135
        :PRINT"S";
        :VDU129
        :PRINT"I";
        :VDU135
        :PRINT"C"
        :NEXT
      :ENDPROC
1450 IF (L%<PAGE OR L%>TOP) VDU133
      :PRINT"Loc ";
      :VDU 134
      :PRINT"Hex";
      :VDU 131
      :PRINT"Mnemonic     ";
      :VDU130
      :PRINT"Dec";
      :VDU135
      :PRINT"A";
      :VDU129
      :PRINT"S";
      :VDU135
      :PRINT"C";
      :VDU129
      :PRINT"I";
      :VDU135
      :PRINT"I"
      :ENDPROC
1460 IF L%>=PAGE AND L%<=TOP VDU133
      :PRINT"Loc ";
      :VDU 134
      :PRINT"Hex";
      :VDU 131
      :PRINT"Mnemonic     ";
      :VDU130
      :PRINT"Dec";
      :VDU135
      :PRINT"B";
      :VDU129
      :PRINT"A";
      :VDU135
      :PRINT"S";
      :VDU129
      :PRINT"I";
      :VDU135
      :PRINT"C"
      :ENDPROC
 1470 ENDPROC
 1480 DATA BRK,1,ORA,8,?,1,?,1,?,1,ORA,4,ASL,4,?,1,PHP,1,ORA,5,ASL,2,?,1,?,1,ORA
,3,ASL,3,?,1
 1490 DATA BPL,11,ORA,9,?,1,?,1,?,1,ORA,10,ASL,10,?,1,CLC,1,ORA,7,?,1,?,1,?,1,OR
A,6,ASL,6,?,1
 1500 DATA JSR,3,AND,8,?,1,?,1,BIT,4,AND,4,ROL,4,?,1,PLP,1,AND,5,ROL,2,?,1,BIT,3
,AND,3,ROL,3,?,1
 1510 DATA BMI,11,AND,9,?,1,?,1,?,1,AND,10,ROL,10,?,1,SEC,1,AND,7,?,1,?,1,?,1,AN
D,6,ROL,6,?,1
 1520 DATA RTI,1,EOR,8,?,1,?,1,?,1,EOR,4,LSR,4,?,1,PHA,1,EOR,5,LSR,2,?,1,JMP,3,E
OR,3,LSR,3,?,1
 1530 DATA BVC,11,EOR,9,?,1,?,1,?,1,EOR,10,LSR,10,?,1,CLI,1,EOR,7,?,1,?,1,?,1,EO
R,6,LSR,6,?,1
 1540 DATA RTS,1,ADC,8,?,1,?,1,?,1,ADC,4,ROR,4,?,1,PLA,1,ADC,5,ROR,2,?,1,JMP,12,
ADC,3,ROR,3,?,1
 1550 DATA BVS,11,ADC,9,?,1,?,1,?,1,ADC,10,?,1,?,1,SEI,1,ADC,7,?,1,?,1,?,1,ADC,6
,?,1,?,1
 1560 DATA ?,1,STA,8,?,1,?,1,STY,4,STA,4,STX,4,?,1,DEY,1,?,1,TXA,1,?,1,STY,3,STA
,3,STX,3,?,1
 1570 DATA BCC,11,STA,9,?,1,?,1,STY,10,STA,10,STX,13,?,1,TYA,1,STA,7,TXS,1,?,1,?
,1,STA,6,?,1,?,1
 1580 DATA LDY,5,LDA,8,LDX,5,?,1,LDY,4,LDA,4,LDX,4,?,1,TAY,1,LDA,5,TAX,1,?,1,LDY
,3,LDA,3,LDX,3,?,1
 1590 DATA BCS,11,LDA,9,?,1,?,1,LDY,10,LDA,10,LDX,13,?,1,CLV,1,LDA,7,TSX,1,?,1,L
DY,6,LDA,6,LDX,7,?,1
 1600 DATA CPY,5,CMP,8,?,1,?,1,CPY,4,CMP,4,DEC,4,?,1,INY,1,CMP,5,DEX,1,?,1,CPY,3
,CMP,3,DEC,3,?,1
 1610 DATA BNE,11,CMP,9,?,1,?,1,?,1,CMP,10,DEC,10,?,1,CLD,1,CMP,7,?,1,?,1,?,1,CM
P,6,DEC,6,?,1
 1620 DATA CPX,5,SBC,8,?,1,?,1,CPX,4,SBC,4,INC,4,?,1,INX,1,SBC,5,NOP,1,?,1,CPX,3
,SBC,3,INC,3,?,1
 1630 DATA BEQ,11,SBC,9,?,1,?,1,?,1,SBC,10,INC,10,?,1,SED,1,SBC,7,?,1,?,1,?,1,SB
C,6,INC,6,?,1
 1640 DATA USERV,&200,BRKV,&202,IRQ1V,&204,IRQ2V,&206,CLIV,&208,BYTEV,&20A,WORDV
,&20C,WRCHV,&20E,RDCHV,&210,FILEV,&212,ARGSV,&214,BGETV,&216,BPUTV,&218,GBPBV,&2
1A,FINDV,&21C,FSCV,&21E,EVNTV,&220,UPTV,&222
 1650 DATA NETV,&224,VDUV,&226,KEYV,&228,INSV,&22A,REMV,&22C,CNPV,&22E,IND1V,&23
0,IND2V,&232,IND3V,&234
 1660 DATA OSCLI,&FFF7,OSBYTE,&FFF4,OSWORD,&FFF1,OSWRCH,&FFEE,OSNEWL,&FFE7,OSASC
I,&FFE3,OSRDCH,&FFE0,OSFILE,&FFDD,OSARGS,&FFDA,OSBGET,&FFD7,OSBPUT,&FFD4,OSGBPB,
&FFD1,OSFIND,&FFCE,NVRDCH,&FFCB,NVWRCH,&FFC8,GSREAD,&FFC5,GSINIT,&FFC2
 1670 DATA OSEVEN,&FFBF,OSRDRM,&FFB9
 1680 DATA 1,1,3,2,2,3,3,2,2,2,2,3,2
 1690 DATA AND,DIV,EOR,MOD,OR,ERROR,LINE,OFF,STEP,SPC,TAB(,ELSE,THEN,coded line,
 ,PTR(left )
 1700 DATA PAGE(left ),TIME(left ),LOMEM(left ),HIMEM(left ),ABS,ACS,ADVAL,ASC,A
SN,ATN,BGET,COS,COUNT,DEG,ERL,ERR
 1710 DATA EVAL,EXP,EXT,FALSE,FN,GET,INKEY,INSTR(,INT,LEN,LN,LOG,NOT,OPENIN,OPEN
OUT,PI
 1720 DATA POINT(,POS,RAD,RND,SGN,SIN,SQR,TAN,TO,TRUE,USR,VAL,VPOS,CHR$,GET$,INK
EY$
 1730 DATA LEFT$,MID$,RIGHT$,STR$,STRING$,EOF,AUTO,DELETE,LOAD,LIST,NEW,OLD,RENU
MBER,SAVE, ,PTR(right)
 1740 DATA PAGE(right),TIME(right),LOMEM(right),HIMEM(right),SOUND,BPUT,CALL,CHA
IN,CLEAR,CLOSE,CLG,CLS,DATA,DEF,DIM,DRAW
 1750 DATA END,ENDPROC,ENVELOPE,FOR,GOSUB,GOTO,GCOL,IF,INPUT,LET,LOCAL,MODE,MOVE
,NEXT,ON,VDU
 1760 DATA PLOT,PRINT,PROC,READ,REM,REPEAT,REPORT,RESTORE,RETURN,RUN,STOP,COLOUR
,TRACE,UNTIL,WIDTH,
 1770 DATA NUL,SOH,STX,ETX,EOT,ENQ,ACK,BEL,BS ,HT ,LF ,VT ,FF ,CR ,SO ,SI ,DLE,D
C1,DC2,DC3,DC4,NAK,SYN,ETB,CAN,EM ,SUB,ESC,FS ,GS ,RS ,US
 1780 DATA nothing,printer only,printer on,printer off,sep. curs's,join curs's,e
nable VDU,beep,back space,fwd. space,down 1 line,up 1 line,clear text,carr. retu
rn,page on,page off
 1790 DATA clear gphs.,def text col,def gphs col,def log. col,default col,disabl
e VDU,mode select,reprog.char.,grph.window,"Plot K,x,y",def't window,escape,text
 window,grph. origin,home cursor,move cursor
 1800 DEF PROCdriver
 1810 V%=&20E
     :X%=?V%
     :Y%=V%?1
     :wrchv=X%+256*Y%
 1820 FOR PASS=4 TO 6 STEP 2
 1830   P%=&AB0
 1840   [
       :OPT PASS
       :PHA
       :CLC
       :CMP#&80
       :BCC end
       :LDA#&1
       :JSR wrchv
       :LDA#&20
       :JSR wrchv
       :LDA#&1
       :JSR wrchv
       :LDA#&20
       :JSR wrchv
       :LDA#&3
       :JSR wrchv
       :PLA
       :JSR wrchv
       :LDA#&2
       :JMP wrchv
       :.end PLA
       :JMP wrchv
1850    PHA
       :LDA#&B0
       :STA V%
       :LDA#&0A
       :STA V%+1
       :PLA
       :RTS
       :PHA
       :LDA#X%
       :STA V%
       :LDA#Y%
       :STA V%+1
       :PLA
       :RTS
       :]
1860    NEXT
1870 ENDPROC
1880 *FX21,0
1890 IF ERR<>17 REPORT
       :PRINT" at ";ERL
       :Z=GET
1900 GOTO360
```

the program to crash. Good error handling can greatly enhance a program and is a boon for the programmer and non specialist user.

● The small machine code routine set up by the PROCdriver shows how interrupting the operating system vectors can get you out of an awkward situation. Here I have interrupted the writer character routine vector WRCHV in order to prevent the teletext control codes going to the printer. If this is not done, my Epson printer becomes quite demented. All I have done is to intercept codes greater than &7F, send two spaces to the printer, turn off the printer, send the code to the screen and turn the printer on again. Two spaces are sent to the printer in order to space out the data and make it more legible. With 80 character per line printers, this leaves plenty of room for hand-written comments.

EXPANSION POSSIBILITIES

There is room for expansion . . . For example, you could get rid of the decimal numbers in the display and change PROCmnemonic to cope with names up to 10 letters long. The user could write a PROC similar to PROCroutine or PROCvector and incorporate his own names, for modules he finds in memory, in the data lists. When the disassembler comes to the start of a valid module, the variable NC% must be set to the start of the location of the start of the module, as NC% holds the location of the next valid op-code. If this is not done, the first byte of the program may be interpreted by the program as part of the operand of the op-code it met. The disassembler would then be out of synchronisation until it met the next op-code with no operand.

BACKNUMBERS

MARCH 1983
Colour Genie reviewed, Epson HX-20 review, PEEKing the Spectrum, Into Atari's BASIC, Terminology translated.

APRIL 1983
Froglet on the BBC Micro, PC-1251 hand-held review, Valley Variations, Galaxy reviewed, Micro Database, Lower case UK101.

MAY 1983
Spectrum Book Survey, Oric-1 Review, Going FORTH Again, Jupiter Ace review.

JUNE 1983
Interrupt handling, Rubic simulation on the Spectrum, Beating the RS232 Blues, Lynx review, Indexer.

JULY 1983
Atari renumber, 16-bit micros, Bomb-proof Tandy, Olivetti Praxis 30 review, Ikon Hobbit tape drive review.

Computing today
MACHINE CODE MUSIC ON THE APPLE:
6502 routines can be adapted for other systems
In-depth review of the Electron – a quantum jump for Acorn?
Monitor review for the Spectrum
MULTI-TASKING ZX81-FORTH– Can it do the job?
Sounding off on the Commodore 64
Getting to grips with Epson printer graphics
Debugging random number software with non-random sequences
Two useful utilities for PET programmers
The chips are down when you play stud poker against the BBC Micro
If you're baffled by machine language you need our new series on Easycode
Hardware project and software routines for a Spectrum Centronics interface
DIY arcade games on the Spectrum – now there's SCOPE for improvement
MEMOTECH'S MTX500 ON THE TEST BENCH: A black pearl from the peripheral people
Learn how to pull machine code apart with our BBC OS12 ROM disassembly feature
Have a Brain Storming session on your CP/M system
More of the same – Recursion on four Apple languages

AUGUST 1983
Speeding up the Sharp, Premier Dragon disc drive, Sord M5 review, BBC String Store, Planetfall.

SEPTEMBER 1983
FELIX knowledge shops, Software protection, Torch disc pack, ZX81 Backgammon, Dragon character generator, Three Tandy computers.

OCTOBER 1983
Slingshot game, Sharp MZ-700 review, Sharp MZ-3541 review, Z80 Disassembler, A better TRSDOS, Improved VIC-20 editor.

NOVEMBER 1983
BBC Word Processor, ZX LPRINT review, Laser 200 review, Writing Adventures, Learning FORTH Part 1, PET tape append.

DECEMBER 1983
MIKRO assembler review, Getting More from the 64 Part 1, Adventures part 2, Curve-fitting, BBC Touch Typing Tutor.

JANUARY 1984
TRS-80 programmer's aid, Apple music, Electron review, TRS-80 screen editor, calendar program.

FEBRUARY 1984
Using MX-80 graphics, Colour Genie monitor, non-random random numbers, ZX81-FORTH, Program recovery on the Commodore 64.

MARCH 1984
Easycode part 1, BBC poker, Spectrum SCOPE review, Genie utilities, Spectrum Centronics interface.

APRIL 1984
Memotech MTX500 review, Genie BASIC extensions, Brainstorm review, Disassembly techniques, Recursion.

If you've lost, lent or had stolen one of those precious back copies of Computing Today then now is your chance to fill the gap in your collection. The list of issues given here represents the few remaining copies that we have available to help complete your library of all that's good in features, programs and reviews.

If you want one of these issues, it's going to cost you £1.40 (including postage and packing) but we think that's a small price to pay for the satisfaction you'll get. Ordering could hardly be made simpler — just fill in the form, cut it out (or send a photocopy) together with your money to:

Backnumbers,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

If you wait until next month to do it, the chances are that we'll have run out of the very issue you wanted!

A Computing today
Reader Service

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £.................................

NAME...

ADDRESS ....

....

....

.... POSTCODE ....

Signature ....

I enclose a cheque/PO for £.....(Payable to ASP Ltd)
I wish to pay by credit card

Access □ Barclaycard □

□□□□□□□□□□□□□□□□□□□□

Insert Card No.

If you wish to pay by Access or Barclaycard, just fill in your card number and sign the form, do not send your card.

Please allow 21 days for delivery.

EXTENDING THE 64'S BASIC PART 5

Tony Cross

Finally, and sadly, we arrive at the last of our articles on improving the Commodore 64 by adding new keywords. OLD and DELETE are on offer this month — if you want any more you'll have to write them yourselves (let us see them too!).

In the earlier parts of this series I have described the various ROM routines as they relate to particular features of the Commodore 64's BASIC (floating point numbers, strings and so on). In this final part I'm going to tidy up the loose ends by looking at the remaining routines that didn't fit into any of these pigeon holes.

PRINT ROUTINES

There are three 'general purpose' print routines in the ROM, for printing both strings and numbers. (I've called them print routines but they actually output to the current device). PRTSTG, at address $AB1E, prints the string pointed to by the A/Y registers (high byte in Y, low byte in A). The string *must* be terminated by a null byte (which is not printed), and all the registers are modified. PRTNUM, at address $BDCD, prints the integer number in the A/X registers (high byte in A, low byte in X). All the registers are modified by PRTNUM. PRTFPA, at address $AABC, prints the floating point number in FPA1. All the registers are modified by PRTFPA.

The PRTSTG routine can also be used to print special 'non-standard' error messages. This is done by simply calling PRTSTG to print the text of the error message, and then calling the PRTERR routine at address $A437. PRTERR prints the "ERROR" message plus an "IN (line number)" clause if the error occured while a program was running.

For example, the following simple program will print the message "COMMAND ERROR":

```
          LDA     #<ERRMSG
          LDY     #>ERRMSG
          JSR     PRTSTG
          JMP     PRTERR
ERRMSG    BYT     'C,'O,'M,'M,'A,'N,'D,$00
```

Note: PRTERR should only be used for 'fatal' error messages because it resets the stack and program pointers, which is why it's jumped to and not called in the program above.

TEXT SCANNING ROUTINES

There may be occasions when you want to know how many characters there are to the end of the current program line (for stepping over keywords like REM, for example). The following two routines will help you do this.

EOLSCN, at address $A909, scans from the current TXTPTR location looking for the next end-of-line character (null). On return, the Y register contains the number of characters to the end of the line. The value of TXTPTR is *not* changed but all the registers are modified.

EOSSCN, at address $A906, scans from the current TXTPTR location looking for an end of statement character (colon or null. Note: colons in quoted strings are ignored). On return, the Y register contains the number of characters to the end of the statement. The value of TXTPTR is *not* changed but all the registers are modified.

Should you want to actually move TXTPTR to the end of the line/statement (or to any location in fact) then SETEND is the routine you need. SETEND, at address $A8FB, adds the offset in the Y register to the current TXTPTR value. The A register is modified but the X and Y registers are not.

For example, to move TXTPTR to the end of the current statement/line use the following two calls:

```
     .
     .
     .
JSR     EOSSCN
JSR     SETEND
     .
     .
     .
```

OPERATIONS ON THE PROGRAM TEXT

In the first part of this series I described how BASIC text is stored as a linked list, and there are two routines in the ROM which operate on the BASIC text.

FNDLIN, at address $A613, searches the program text for a line with a particular line number. The line number in question is specified in locations $14/$15 (high byte in $15, low byte in $14). On return, the carry flag is set if the line has been found and locations $5F/$60 contain the address of the start of the wanted line (high byte in $60, low byte in $5F). If the line was not found (carry flag reset), then locations $5F/$60 contain the address of the start of the next higher line number. All the registers are modified by FNDLIN.

LNKPTR, at address $A533, re-links the link pointer bytes in the program text. It is mainly used by the Editor when inserting or deleting program lines, but it can be called whenever you need to re-link the program text.

SOME NEW KEYWORDS

The final two routines (OLD and DELETE) both operate on the program text. As its name implies, OLD restore a program that has been NEWed. It works by restoring BASIC's pointers to point to the program currently in memory.

DELETE start line, end line

DELETE erases a block of program lines. The lines erased are

CORRECTION

Please note that there were two errors in last month's article (Part 4). On page 68, second column, line 7, the address of the ASCII routine is actually $B78B. On page 71, second column, last line, the 'number of elements' bytes should be $00 $0D.

from 'start line' to 'end line' inclusive. For example, a DELETE 30,50 statement will delete all lines of code between (and including) lines 30 to 50.

THE OLD KEYWORD

The NEW keyword works by resetting BASIC's pointers to point to the start of the program text area. In addition, the first two link pointer bytes are set to null. So, although the program is still in memory, BASIC can no longer see it. Restoring a NEWed program is therefore a two stage process: first the link pointers are restored, and then the end of program pointers are replaced.

The OLD keyword routine restores the link pointer bytes by calling LNKPTR. Before this can be done, however, the null in the high byte of the first link pointer must be removed (because it indicates the end of the program). OLD does this by writing a $01 byte in there (you can in fact put any value in there as long as it's not null).

To find the end of the program OLD calls FNDLIN. However, at this stage we have no idea of what the highest line number might be. So, to guarantee finding the highest line, OLD searches for a line with the highest possible line number, that is, 65535.

The address that FNDLIN returns is two bytes short of the end of the program, so OLD increments this address by two. This modified address is then copied into the end of program and variable pointers.

With all the relevant pointers replaced, the NEWed program is now completely restored.

THE DELETE KEYWORD

Basically, DELETE calls FNDLIN to get the address of the 'start line' and 'end line'. It then copies the rest of the program (from 'end line+1') up to the address of 'start line' (ie over the lines to be deleted).

The 'start line' number is extracted using the NUMEXP/POSINT technique that we've looked at before. After calling FNDLIN the carry flag is tested to make sure that the line exists. If it doesn't then a special error message is given (see later). If the line exists then its address is saved on the stack.

The 'end line' number is then extracted, and FNDLIN is called to check that the line exists (by testing the carry flag). Now, since the 'end line' must also be erased, what we really need is the address of the next line. This can easily be found by incrementing the line number by one ('end line+1') and then calling FNDLIN to find its address.

Listing 1. The OLD keyword.

```
 10 033C          !###################################
 20 033C          !#                                 #
 30 033C          !#           OLD KEYWORD           #
 40 033C          !#                                 #
 50 033C          !#     VERSION 1.0 -- 10/02/84     #
 60 033C          !#                                 #
 70 033C          !# COPYRIGHT (C)  A.L.CROSS 1984  #
 80 033C          !#                                 #
 90 033C          !###################################
100 033C          !
110 033C          !
120 C5C3          *=$C5C3
130 C5C3          !
140 C5C3          !
150 C5C3          !VARIABLES AND EQUATES
160 C5C3          !
170 C5C3          LNKPTR        =  $A533
180 C5C3          FNDLIN        =  $A613
190 C5C3          PRTSTG        =  $AB1E
200 C5C3          !
210 C5C3 E000     OLD           CPX #$00                  !CHECK STATEMENT FLAG
220 C5C5 F003                   BEQ DOOLD
230 C5C7 4C08AF                 JMP $AF08
240 C5CA A901     DOOLD         LDA #$01
250 C5CC A8                     TAY
260 C5CD 912B                   STA ($2B),Y               !OVERWRITE NULL BYTE
270 C5CF 2033A5                 JSR LNKPTR                !RE-LINK PROGRAM
280 C5D2 A9FF                   LDA #$FF
290 C5D4 8514                   STA $14                   !SET MAX LINE NUMBER
300 C5D6 8515                   STA $15
310 C5D8 2013A6                 JSR FNDLIN                !FIND LAST LINE ADDRESS
320 C5DB A902                   LDA #$02                  !ADD 2 TO LAST LINE
330 C5DD 18                     CLC                       !ADDRESS
340 C5DE 655F                   ADC $5F
350 C5E0 852D                   STA $2D                   !SET NEW END OF PROGRAM
360 C5E2 852F                   STA $2F                   !AND VARIABLES
370 C5E4 8531                   STA $31
380 C5E6 A900                   LDA #$0
390 C5E8 6560                   ADC $60
400 C5EA 852E                   STA $2E
410 C5EC 8530                   STA $30
420 C5EE 8532                   STA $32
430 C5F0 60                     RTS
```

Listing 2. The DELETE keyword.

```
 10 033C          !###################################
 20 033C          !#                                 #
 30 033C          !#         DELETE  KEYWORD         #
 40 033C          !#                                 #
 50 033C          !#    VERSION 1.0 -- 10/02/84      #
 60 033C          !#                                 #
 70 033C          !# COPYRIGHT (C)  A.L.CROSS 1984  #
 80 033C          !#                                 #
 90 033C          !###################################
100 033C          !
110 033C          !
120 C5F1          *=$C5F1
130 C5F1          !
140 C5F1          !
150 C5F1          !VARIABLES AND EQUATES
160 C5F1          !
170 C5F1          LNKPTR        =  $A533
180 C5F1          FNDLIN        =  $A613
190 C5F1          NUMEXP        =  $AD8A
200 C5F1          POSINT        =  $B7F7
210 C5F1          TSTCOM        =  $AEFD
220 C5F1          PRTSTG        =  $AB1E
230 C5F1          PRTERR        =  $A465
240 C5F1          !
250 C5F1 E000     DELETE        CPX #$0                   !CHECK STATEMENT FLAG
260 C5F3 F003                   BEQ DODEL
270 C5F5 4C08AF                 JMP $AF08
280 C5F8 208AAD   DODEL         JSR NUMEXP                !GET START LINE
290 C5FB 20F7B7                 JSR POSINT
300 C5FE 2013A6                 JSR FNDLIN                !GET LINE ADDRESS
310 C601 9066                   BCC LINERR
320 C603 A560                   LDA $60                   !STACK ADDRESS
330 C605 48                     PHA
340 C606 A55F                   LDA $5F
350 C608 48                     PHA
360 C609 20FDAE                 JSR TSTCOM
370 C60C 208AAD                 JSR NUMEXP                !GET END LINE
380 C60F 20F7B7                 JSR POSINT
390 C612 2013A6                 JSR FNDLIN                !CHECK THAT LINE EXISTS
400 C615 9052                   BCC LINERR
410 C617 E614                   INC $14                   !INC LINE NUMBER
420 C619 D002                   BNE GETEND
430 C61B E615                   INC $15
440 C61D 2013A6   GETEND        JSR FNDLIN                !GET END LINE+1
450 C620 68                     PLA                       !RECOVER START ADDRESS
460 C621 8514                   STA $14
470 C623 68                     PLA
480 C624 8515                   STA $15
490 C626 38                     SEC
500 C627 E560                   SBC $60                   !CHECK THAT START
501 C629 F004                   BEQ CHKLOW                !ADDRESS IS LESS
510 C62B 9009                   BCC ADRSOK                !THAN END ADDRESS
520 C62D B03A                   BCS LINERR
530 C62F A514     CHKLOW        LDA $14
540 C631 38                     SEC
550 C632 E55F                   SBC $5F
560 C634 B033                   BCS LINERR
570 C636 A000     ADRSOK        LDY #$0
580 C638 A52E     LOOP          LDA $2E
590 C63A C560                   CMP $60                   !END OF PROGRAM?
600 C63C D006                   BNE NOTEND
610 C63E A52D                   LDA $2D
620 C640 C55F                   CMP $5F
630 C642 F012                   BEQ PRGEND
640 C644 B15F     NOTEND        LDA ($5F),Y               !MOVE ONE BYTE
650 C646 9114                   STA ($14),Y               !DOWN
660 C648 E65F                   INC $5F
670 C64A D002                   BNE DODEST
680 C64C E660                   INC $60
690 C64E E614     DODEST        INC $14
700 C650 D0E6                   BNE LOOP
710 C652 E615                   INC $15
720 C654 D0E2                   BNE LOOP
730 C656 A514     PRGEND        LDA $14                   !SET NEW PROG END
740 C658 852D                   STA $2D
750 C65A 852F                   STA $2F
760 C65C 8531                   STA $31
770 C65E A515                   LDA $15
780 C660 852E                   STA $2E
790 C662 8530                   STA $30
800 C664 8532                   STA $32
810 C666 4C33A5                 JMP LNKPTR
820 C669 A973     LINERR        LDA #<ERRMSG
830 C66B A0C6                   LDY #>ERRMSG
840 C66D 201EAB                 JSR PRTSTG
850 C670 4C65A4                 JMP PRTERR
860 C673 0D0A4C   ERRMSG        BYT $0D,$0A,'L,'I,'N,'E,$20
870 C67A 4E554D                 BYT 'N,'U,'M,'B,'E,'R,$00
```

A check is then made to ensure that 'end line+1' is greater than (or equal to) 'start line'. A special error is given if not (see later).

It is then simply a case of copying the rest of the program, from 'end line+1', up to the address of 'start line'. When this has been completed, locations $14/$15 (which were used as a pointer during the copying process) contain the new end of program address. This address is written into the end of program and variable pointers. Finally, LNKPTR is called to restore the link pointers.

A special error message is printed if a non-existent line number is specified, or if 'end line+1' is less then 'start line'. The PRTSTG/PRTERR technique is used to print the message "LINE NUMBER ERROR".

AND FINALLY

Sadly that brings us to the end of this series on the Commodore 64 BASIC. I hope you've found it both informative and useful, and I'll look forward to seeing some of your keyword routines in future issues of Computing Today.

PROCOPINION

Peter Green

More ramblings from the editor. This month he's actually dipped his hand into his wallet, and has some thoughts on Acorn, Sinclair, and a certain software company.

I have finally done it. I have finally spent some of my own money on a home computer. Up until now I have never felt the need for any personal outlay in this area, for several reasons. Not the least of these is my mortgage; and with a large number of machines of various types that we have in the office, it's always been possible to borrow one for a while when I've needed to test software or write some programs.

More importantly, I'm something of a perfectionist when it comes to items that hit my pocket. And I've always found that there's something about every home computer that I didn't like. The BBC and Electron have a nice BASIC but are very tight on memory. The Spectrum is capable of some amazing feats of graphics given its price, but the keyboard is almost impossible to work with. The Commodore 64 has excellent sound and graphics, provided you can work out how to use them and don't mind a machine that has disc drives that are as slow as cassettes. And so on, and so forth. There's always something I'm not happy with.

Well, the Amstrad isn't a perfect machine either, but it's close enough, and at the right price, to make me reach for the chequebook.

NICE ONE, ALAN

There isn't any doubt in my mind that Alan Sugar, company chairman of Amstrad, has got a winner on his hands. People who have come into the offices and seen the computer here have all remarked on what a nice machine it is. Several contributors to *Computing Today*, as well as staff here, have expressed a desire to buy one (several of us already have). A colleague tells me her local Rumbelows is selling Amstrads as fast as they can get them out of their warehouse.

There will be a full bench test of the Amstrad in the next issue of *Computing Today*, but I can let you know a little bit about it to be going on with. First, it's available *now*. It was the availability of the Dragon 32 at a time when Spectrums and BBCs were hard to come by that fuelled its success, and I think the same will be true regarding the Amstrad and the QL. Amstrad have reversed the industry trend of broken promises and receding delivery dates with a vengeance.

When the computer was announced, it was stated that it would be on sale in the shops in July. It *was* on sale in July. I ordered mine direct from the company on a form stating 28 days delivery. I sent my cheque on a Tuesday and the equipment was delivered the following Thursday; that's seven working days. When Amstrad say that the disc drives will be available in the Autumn, I believe them: they haven't put a foot wrong so far.

The whole concept of the machine is a nice change for people fed up with untangling masses of cable. There is a monitor unit containing the power supply for the keyboard, and the keyboard unit itself containing the data recorder. One mains cable for the whole lot, no separate power supply to trip over or trailing cassette leads all over the desk: this alone makes the use of the machine a pleasure.

The BASIC is very rich, and has some unusual features that are very powerful, such as the ability to transfer program control in response to interrupts. There is a WHILE-WEND construct (rather than REPEAT-UNTIL), arrays may be erased once they are of no further use, there are extensive sound and graphics commands, and useful things such as conversion of a string to all upper case or lower case letters.

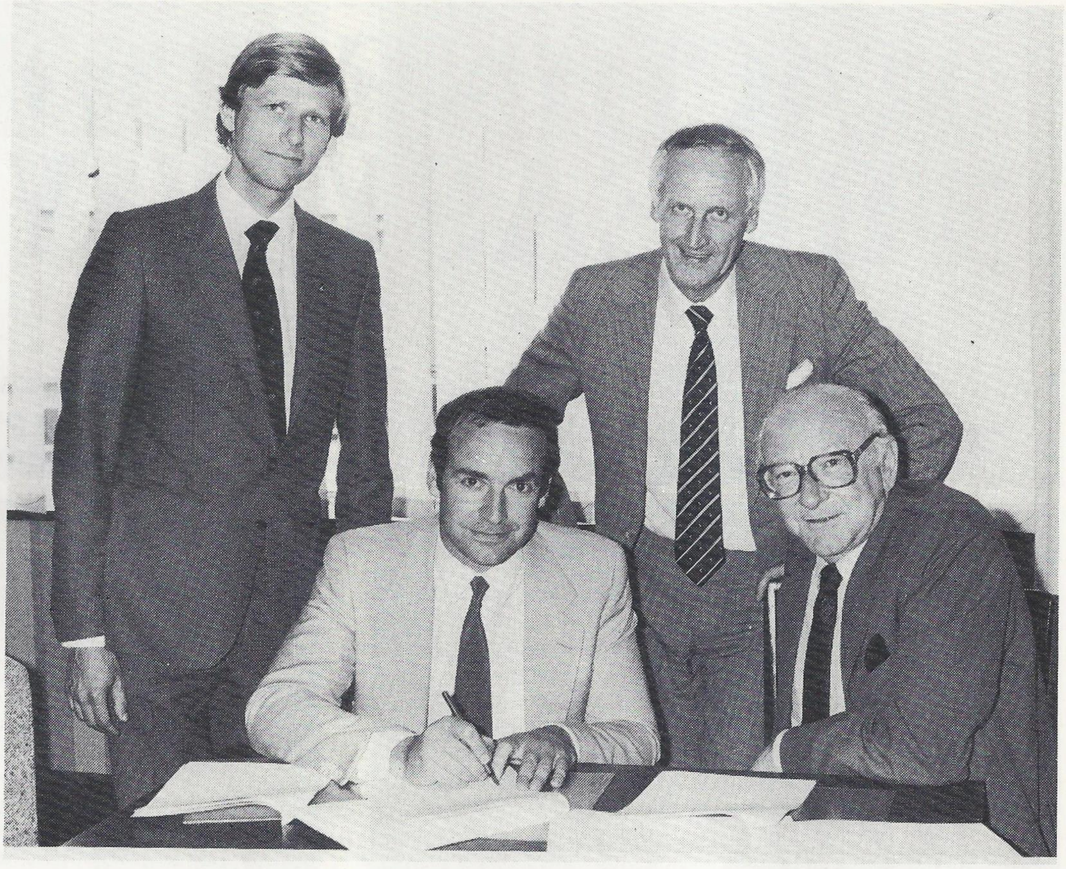
One of the things I like least is that Amstrad BASIC is fussy about spaces. It insists that all BASIC keywords have spaces separating them or it won't recognise them. The reason for this is that Amstrad have plumped for unlimited variable names, including names that begin with reserved words. For example, you could have a variable called TOTAL, which most BASICs would not allow because TO is a keyword. Having spent a lot of time writing in BBC BASIC over the last few months, and training myself to leave out all unnecessary spaces in order to conserve memory, I now find I have to relearn putting them in again. C'est la vie . . .

Actually the easy way to do this is to type your programs in lower case letters only. The editor will convert anything it recognises as a keyword to capitals when listing, so you can scan through the program when you've finished, checking for errors quite easily.

There are three screen modes, corresponding to the BBC's Modes 0 to 2. They all take the same amount of memory, but make a trade-off between colours available (maximum 16) and characters per line (maximum 80). The screen, therefore, doesn't encroach on the user memory like the BBC — at power-up you have 42.5K free, and it doesn't change with mode.

Amstrad promised that games software would be available when the machines were, and it is, although to be

Alan Sugar and his sweet little computer . . .



Meanwhile, back at the BBC, it's another four-year contract for Acorn. From left to right we have Dr. Herman Hauser and Chris Curry of Acorn; Bryon Parkin, Managing Director of BBC Enterprises Ltd; and the BBC's Chairman, Bill Cotton.

fair I have to point out that what we've seen so far is pretty dire. However, this is set to change quite rapidly as the large software houses are very interested in the machine. For example, Jet Set Willy is billed as being 'available soon', and I'm sure many other popular titles will be following in its footsteps. The fact that the Amstrad cassette recorder has a protected save facility should go a long way to cutting down on software piracy, making the programmers a lot happier. (I wish Amstrad would explain why the Welcome tape supplied with the machine is protected: surely it would make sense to let users see how the demo programs were written? Ho hum . . .)

So the future looks bright on the games front, and when the disc drives arrive later this year the door to CP/M will be open, providing business users with thousands of off-the-shelf programs to run. (A tip: if you are planning mainly business use, I suggest the monochrome monitor. Eighty column text on the colour monitor is not kind to the eyes after a while.)

I think the CPC464 has an outstanding career ahead of it.

QUEUE HELL

Meanwhile the sorry Sinclair story drags on. Back at the press launch (January 12th, remember?), I almost decided to buy one. I'm glad I didn't. At the time I went into print saying that the QL would run and run: but it wasn't the Comedy of Errors that I had in mind when I said that. Sir Clive has done great damage to the stockpile of goodwill and respect that people had for Sinclair, and the computer press in particular has been running increasingly hostile articles recently.

If the premature launch was done in an attempt to get the QL considered for the BBC contract when it fell due for renewal, it was badly misjudged and has come to nothing. People are cancelling their orders, and I know of one long-standing customer who, having finally received his machine, had it break down totally after only two weeks. I feel sorry for Sinclair, but they really have no-one to blame but themselves.

Meanwhile the position of this magazine remains unchanged. If a complete, undongled, production line QL becomes available to us, we will review it. We are not interested in reviewing pre-production prototypes, which is what most customers are getting.

IT'S ACORN AGAIN

It probably won't come as much of a surprise to learn that BBC has signed an agreement with Acorn Computers to extend their manufacture and distribution of the BBC Microcomputer for a further four-year period. There was much speculation in the press as to whether Acorn were going to be axed, and Sir Clive in particular has made no secret of his desire to land the contract, but did anyone seriously think the BBC would change horses in mid-stream? Consider: the BBC makes schools programs. There have been 350,000 BBCs sold to date, and a lot of them are in schools and universities. The Corporation would have been mad to throw away that user base.

At Sinclair, apparently, they are putting a brave face on things and muttering that their chance will come the next time round when extensions to an existing machine will no longer be sufficient. The attitude seems to ignore the fact that the whole design concept of the BBC Micro is of a flexible I/O device capable of almost indefinite expansion.

DON'T DO AS I DO . . .

Some weeks ago a letter was circulated among the computer press by software house Microdeal. The subject was piracy, and the letter made accusations about computer clubs being formed simply to copy commercial software. So you might be interested in the following press release which we have just received:

"Activision Inc instituted proceedings in the High Court in London on 6 July 1984 against Microdeal Limited on the grounds that Microdeal's computer game 'Cuthbert in the Jungle' was a copy of Activision's game 'Pitfall'. Activision claimed damages for infringement of their copyright in the program and visual presentation of 'Pitfall' and applied to the Court for an Injunction to prevent Microdeal selling 'Cuthbert in the Jungle'.

"The proceedings were resolved late last week when Microdeal gave undertakings to the Court that they would not make any further copies of 'Pitfall' or sell any more copies of their game 'Cuthbert in the Jungle'.

"Activision views the infringement of copyright very seriously and will not hesitate to take action again should the occasion arise," says Geoff Heath, Managing Director of Activision in the UK.

MACHINE CODE THE EASY WAY

Peter Green

New Generation have turned from the likes of Trashman and 3D Tunnel to produce some very nice educational software. We've put their Complete Machine Code Tutor under the microscope — here's our opinion.

There are a great many books around that deal with the subject of machine code programming, both as a general treatise on one type of microprocessor or as a specific work which relates to one particular computer. The drawback with any type of book, whatever its nature, is that programming is best learnt by doing, not reading. There really is no substitute for sitting at the keyboard, typing in numbers and watching what happens next.

While it is possible to POKE bytes of machine code directly into memory, this is not exactly the easiest of systems that you can have. It is very tedious to use, even if you are using a loader-type program that accepts the data byte by byte at the keyboard and automatically puts it into consecutive memory locations. It is not at all easy to keep track of where you are in the program, or what the code you've already entered means. You can't check what the microprocessor is doing when you execute the code. And the code itself is likely to contain a lot of bugs since converting to hex by hand is not a job suited to the average human brain (hackers excluded, of course).

More to the point, it's boring.

ASSEMBLY POINT

It is at this point that you realise an assembler is needed. A decent assembler will allow you to type in the machine code mnemonics, such as LD A,H (LoaD the Accumulator with the contents of the H register), instead of insisting on 7C. It will allow you to identify memory locations by label rather than by absolute address, to make program writing even simpler and easier to follow. It will calculate relative jumps for you and warn you if you try to exceed the maximum jump allowed. And so on.

Unfortunately, of the leading machines in the home computer field, only the BBC Micro contains a built-in assembler as standard. And in this case, the accompanying explanation of how to program in machine code has to fit into just one chapter of the User Manual, so it's understandably brief. Most instructions and addressing modes are only given a cursory treatment.

Into this gap have stepped New Generation, with their range of Complete Machine Code Tutors. Initially this consists of a range of four: for the BBC Model B, the 48K Spectrum, the Commodore 64 and the Atari.

WHAT YOU GET

In each of the versions you get two cassette tapes, the tutor requiring all four sides. The first side contains a comprehensive editor/assembler plus the first block of lessons and examples, with the other examples and lessons spread over the remaining sides. Spectrum owners get 100K of data, with 70K on the other machines.

Although the structure of each Tutor is broadly similar, naturally the Spectrum version is completely different in content because it alone uses the Z80 processor (the other three machines being 6502-based). The other main difference is that the Spectrum version allows a much greater amount of memory to be used to store your own program: 6912 bytes on the Spectrum, but only 1K on the BBC.

I chose to concentrate on the Spectrum tape, although I looked at the BBC version for comparison (the two other tapes were not available at the time of review but have since been released). The Spectrum tutor has 33 lessons, each of the others has 27: the Z80 is a more complicated processor with extra instructions, so more lessons are needed to cover all its features. Although it's capable of separate use, the beginner is advised not to use the editor/assembler first, so we too will start with the lessons. The editor/assembler must be loaded before they can be run, however.

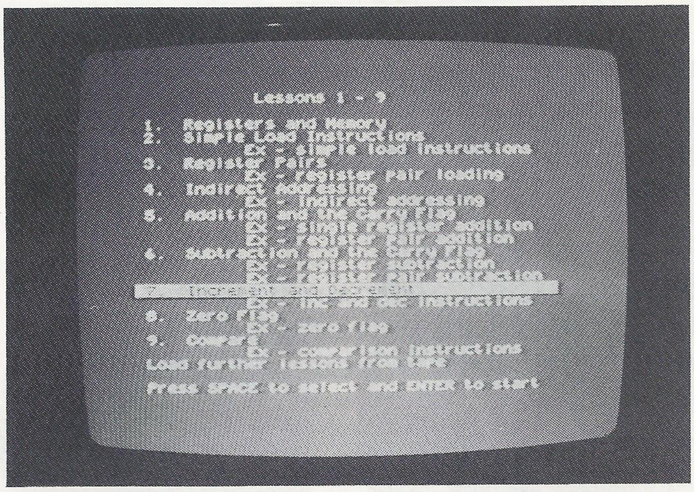
HERE BEGINS THE LESSON

Initially you are asked whether you want to load a batch of examples. Once they are in, a menu appears for that batch showing the subject of each lesson and the associated examples. Selection has been made foolproof — a highlighted cursor is moved down the menu by pressing Space, followed by Enter to run the chosen section (confusingly, this is the opposite of the BBC version).

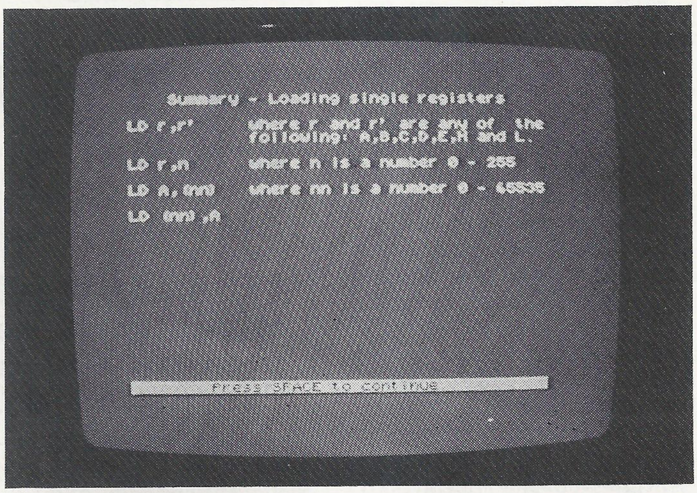
Once a lesson has been selected, a block of text appears on the screen describing the action of the group of machine code instructions that you've picked. So far, not too different to an ordinary book. The difference comes at the end of the lesson (which may run to two or three screenfuls of text), because the example part of the program sets up the assembler with a demonstration program which features the instructions just covered. Again, a highlighted bar indicates the current instruction, and a description of its action is displayed below the program. Beneath this are displayed the current contents of the flags register, each of the other on-chip registers and any relevant memory locations, plus a portion of the assembler's stack area. When you RUN the demo, you actually step through it in single-step mode, with the new contents of the registers appearing each time. In this way you can try to work out what should happen, then check your guess immediately.

This sort of feedback is obviously going to speed up the learning, but the Tutor program goes further. It is possible to edit the example program and see what effect this has on the way the program runs, or even wipe it completely and write your own. This is completely safe: the assembler has a comprehensive set of error messages built in so that you cannot crash the machine.

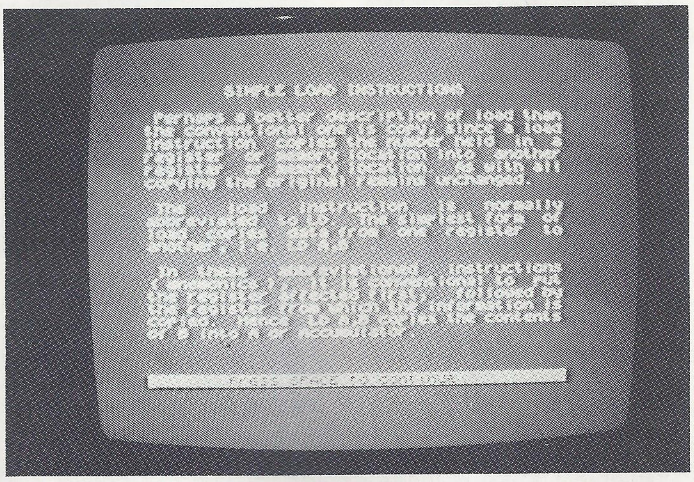
Among the usual sorts of error such as "Offset too big", there are errors specific to the simulator. These include "You are about to affect memory not allocated to you" (ie you're mucking about outside the safe limits set by the simulator), and "You are trying to run code in an allocated storage area" — the simulator will prevent you causing the program counter to jump to an



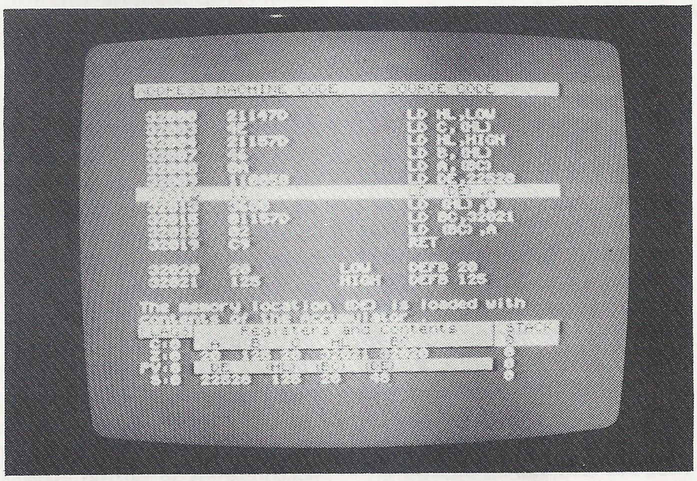
```
Lessons 1 - 9

1. Registers and Memory
2. Simple Load Instructions
          Ex - simple load instructions
3. Register Pairs
          Ex - register pair loading
4. Indirect Addressing
          Ex - indirect addressing
5. Addition and the Carry Flag
          Ex - single register addition
          Ex - register pair addition
6. Subtraction and the Carry Flag
          Ex - register subtraction
          Ex - register pair subtraction
7. Increment and Decrement
          Ex - inc and dec instructions
8. Zero Flag
          Ex - zero flag
9. Compare
          Ex - comparison instructions
Load further lessons from tape

Press SPACE to select and ENTER to start
```



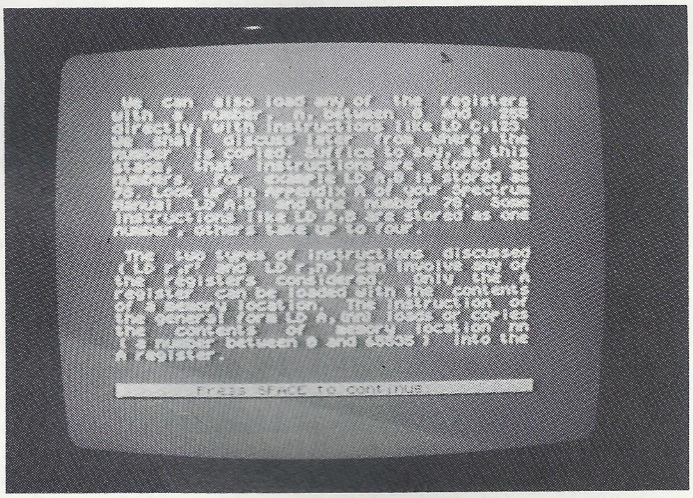
```
Summary - Loading single registers

LD r,r'      where r and r' are any of the
             following: A,B,C,D,E,H and L.

LD r,n       where n is a number 0 - 255

LD A,(nn)    where nn is a number 0 - 65535

LD (nn),A


         Press SPACE to continue
```



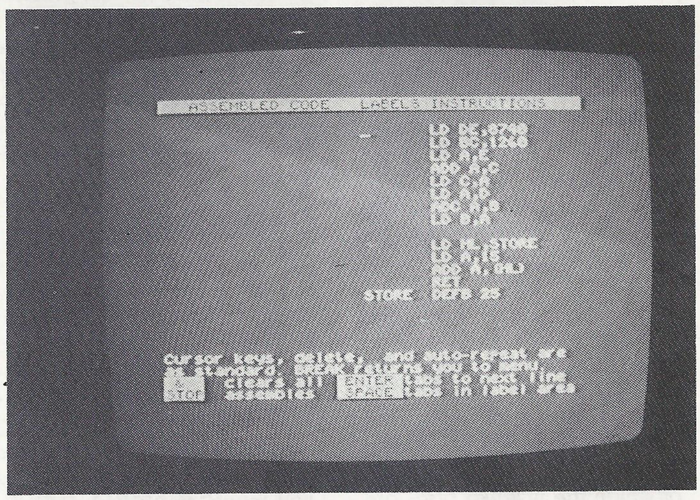
```
SIMPLE LOAD INSTRUCTIONS

Perhaps a better description of load than
the conventional one is copy, since a load
instruction copies the number held in a
register or memory location into another
register or memory location. As with all
copying the original remains unchanged.

The load instruction is normally
abbreviated to LD. The simplest form of
load copies data from one register to
another, i.e. LD A,B

In these abbreviated instructions
(mnemonics) it is conventional to put
the register affected first, followed by
the register from which the information is
copied. Hence LD A,B copies the contents
of B into A or Accumulator.

         Press SPACE to continue
```



```
[illegible]

32000    211470               LD HL,LOW
32003    4E                   LD C,(HL)
32004    211570               LD HL,HIGH
32007    46                   LD B,(HL)
[illegible]
32013    211570               LD BC,32021
[illegible]
32019    C9                   RET

32020    20          LOW      DEFB 20
32021    125         HIGH     DEFB 125

The memory location (HL) is loaded with
contents of the Accumulator
          Registers and Contents
[illegible]
A/F      22520   125   20   40
```



```
We can also load any of the registers
with a number (n) between 0 and 255
directly with instructions like LD C,155.
We shall discuss later from where the
number is copied. Suffice to say at this
stage, that instructions are stored as
numbers. For example LD A,B is stored as
78. Look up in Appendix A of your Spectrum
Manual LD A,B and the number 78. Some
instructions like LD A,B are stored as one
number, others take up to four.

The two types of instructions discussed
(LD r,r' and LD r,n) can involve any of
the registers considered. Only the A
register can be loaded with the contents
of a memory location. The instruction of
the general form LD A,(nn) loads or copies
the contents of memory location nn
(nn is a number between 0 and 65535) into the
A register.

         Press SPACE to continue
```



```
ASSEMBLED CODE     LABELS  INSTRUCTIONS
                           LD DE,4700
                           LD BC,1268
                           LD A,E
                           ADD A,C
                           LD C,A
                           LD A,D
                           ADC A,B
                           LD B,A

                           LD HL,STORE
                           LD A,16
                           ADD A,(HL)
                           RET
                   STORE   DEFB 25


cursor keys, delete, and auto-repeat are
as standard. BREAK returns you to menu.
[illegible]  clears all  ENTER tabs to next line
STOP  assembles     SPACE tabs in label area
```

address which has been allocated as storage (a machine code 'variable'). In other words, the Tutor prevents you from treating data as instructions, so you can't write self-modifying code. This is fair enough, because it's not very elegant and can cause all kinds of trouble if you aren't careful.

The screen formatting on the Spectrum is a pleasure to see, especially when you think of the horrors perpetrated by some commercial companies. Text is yellow against a blue background, with the highlighted cursor appearing in cyan. The actual lessons are displayed tidily on the screen, and I spotted no spelling errors or examples of poor grammar.

The BBC version of the Tutor has an extra refinement, in that you can choose the Spectrum-type display of the simulator with the registers along the bottom of the screen, or switch to a vertical format with the registers down the left-hand side. This does mean that there is only room for the assembly language mnemonics to be shown, however, there's not enough space on a line to give the hexadecimal equivalents of the instructions.

CONCLUSIONS

New Generation have got a very nice product here. It's well thought out, easy to use, aesthetically pleasing, and (as far as I could tell) bug-free and crash-proof. I always tend to shudder a little when I see cassette software costing more than a tenner, and the Machine Code Tutor is priced at £14.95, but if you want hands-on experience of machine code in a beginner's environment then it provides an ideal tool.

Overall then, an excellent idea, well executed. How about producing a version which allows proper, longer machine code programs to be written, debugged and SAVEd permanently? Drop the price a little (most of the hard work is already done) and you'll probably clean up, gentlemen.

ADVENTURES

Peter Rabett

For such a small machine, the Spectrum packs a mighty graphics punch once you've learnt how to use it properly. A number of Adventure writers are putting that to good use.

Back when adventure was invented, people programmed on mainframes and home micros were still just a twinkle in their designer's eye. You didn't communicate with these things using a VDU, you typed your commands in on a teletype, and received output the same way. Thus the first Adventure was, of necessity, text-based, and in order to maintain the complexity of that early, classic effort, adventure programs for microcomputers have tended to follow suit.

However, technology marches on, and with startling speed the graphics capabilities of even the cheapest home micro have become quite sophisticated, with high resolution and colour just about standard. Human nature and programmers being what they are, it was inevitable that people would want to exploit these facilities. So we are seeing a new breed of adventure appearing more and more frequently — the graphics adventure.

DRAWING THE LINE

The problem with writing about graphic adventures is, where do you draw the line (figuratively, not literally!). The distinction between arcade games and adventures is not getting pretty blurred, with elements of logic and strategy binding the two. Graphics can be approached in sevral ways.

The most basic approach is to take a normal text adventure and just add the occasional user-defined graphic character here and there. This method is used in Incentive Software's Ket trilogy, of which the first two parts are now available.

The next step is to add a small graphical depiction of each (or some) of the locations to the text display. This 'window' is normally at the top of the screen and is quite popular now on Spectrum adventures. The program that started the ball rolling was Melbourne House's Hobbit, of course, and it's still among the best. The Phipps Associates adventures reviewed a few months ago in this magazine use the technique, and more recent examples that I've seen include Invincible Island by Richard Shepherd Software, and the Odyssey of Hope by Software Communications. The more venerable Mysterious Advantures by Brian Howarth, which are sold for the Spectrum by Digital Fantasia, have a variation on this theme — you can toggle between a graphics display of your current location, or a plain text display for speed (and extra information).

Finally there are adventures that are almost totally graphics-based, and it's hard to know where these end and arcade games begin. Things like the War of the Worlds from CRL and Lothlorien's Special Operations are obviously adventures, but you can make a good case for including something like Ultimate's latest game, Sabre Wulf.

SABRE WULF

The reason why I feel this deserves the benefit of the doubt is that, although basically a maze-running game, it does involve elements of traditional adventuring. There is a large playing area to map out (256 locations on a 16 by 16 grid), and as you move about you encounter other creates and artifacts which you can pick up (this being graphics only, you pick things up automatically by walking over them). Some (most!) items are useless in that they only give you extra points, but some things are worth their weight in gold. The manual is rather vague about the objectives and playing of the game (in fact it's a dreadful bit of prose), so like a real adventure you have to observe and work out what's happening (how do you get extra lives, for example?).

Fun though this game is — the animation is superb, especially the Wulf of the title which trots along, crouches and pounces — it is relatively easy to solve. I managed to crack it in a week and half of occasional evening play.

KET

The Ket trilogy has two parts on sale so far, The Mountains of Ket and The Temple of Vran. This is almost entirely a text adventure series, and I only include them here because your inventory, when called up, has little drawings of your belongings down the side of the list: also, I enjoyed playing them very much!

You take the role of a condemned prisoner seeking a reprieve by saving your country from the vicious attacks of baddies Vran and Delphia. In doing so, you will not only require the usual adventuring skills of patience and logic, but a little economic sense will help too. How do you buy goods worth eight gold pieces when you only start with two?

As an added bonus, players scoring 100% will get a special message displayed on the screen. According to Incentive, it is "advantageous" to make a note of this. I've managed it on Ket, but the Temple of Vran is eluding me. I'm sure I know how to cross the quicksand, I just can't find the right way to express it . . . Recommended.

INVINCIBLE ISLAND

This program by Richard Shepherd Software has a graphics window at the top displaying a rudimentary view of each location you visit. Drawing is relatively fast since the scenes are mostly line drawings with an occasional fill, but it would be nice to disable them or only have them displayed on the first entry into a location. An average sort of game, with some rather strange logic in places. When I climb down a well, I must admit to a slight feeling of surprise when I find myself in a valley. Oh well — better than watching Coronation Street, I suppose.

I tried playing Virgin Games' The Island to compare it with the previous game, but it steadfastly refused to load, despite all my coaxing.

THE GOLDEN BATON

It seems that this game has been around almost as long as Classic Adventure, and it turns up in a variety of guises on several machines. It's the original adventure written by Brian Howarth, who now has a list of about 10 adventures to his name. The nice thing about the Spectum version, which I obtained from Digital Fantasia, is that the graphical scenes, one for each location, can be turned off at will simply by pressing Enter. From then on you are playing a normal text adventure, which speeds up the playing time considerably.

I think that this game would be a good one to start off with if you are new to Adventuring. It's easy enough to get quite a way into the puzzle-solving, so beginners can learn the ropes (hint?) before tackling something more devious.

THE ODYSSEY OF HOPE

This is a brand new adventure from Software Communications Ltd. The time is Ancient, the place is Greece and the object is to find Hope, after

TEMPLE of VRAN
INCENTIVE
MOUNTAINS of KET
Invincible Island by Peter Cooke
An intriguing split screen graphic and text adventure for the 48K Spectrum
RICHARD SHEPHERD SOFTWARE
martech
THE ODYSSEY OF HOPE
LOTHLORIEN
SPECIAL OPERATIONS
Games
THE ISLAND
Jeff Wayne's Video Game Version of The War of the Worlds
CRL
ZX - SPECTRUM 48K

Pandora's box has been opened. The instruction program is rather clever — real pages of text are displayed on the screen, which actually 'turn over' to reveal the page below as you 'read the book'. I liked it.

The main program is another text-below, graphics-above type of game, with a scene for each location. Some of these are quite intricate, and when you come across the dead cow in the field, the program provides a buzzing noise: to represent the flies, I suppose! Again, some of the puzzles are a bit illogical — why would a python turn into a flute when you kill it? — but some of the features are rather clever and it is quite nice to look at. Worth considering.

SPECIAL OPERATIONS

This Lothlorien game has you cast as the leader of a Special Operations Squad, who has to pick a team of four men and lead them into German occupied territory to infiltrate a secret establishment. You can choose various levels of difficulty, both as to the type of operation (from just finding the camp to actually stealing the research materials), and playing time. The game simulates real-time operation, with activities such as marching cross-country making your clock advance at the same rate as it would in real life.

The initial phase, picking the team, is text-based and involves you 'interviewing' a large group of men to identify the most useful combinations of skills (each man has two). After that, play moves onto a series of maps.

The main map is actually not a true representation of the ground area, only stylised; you can only be sure of features lying adjacent to your current position (slightly further if you have a scout in the team) — a nice real-life touch. Some of the features can be further investigated; for example, potholes may be explored or caves penetrated. In this case the action is displayed on a small subsidiary 'map' at the side of the screen., Again, as in real life you can only 'see' in straight lines, so as you move through the tunnels, extra passages reveal themselves.

Encountering a German patrol leads to combat on another small map, and involves a sequence of rounds where you and the enemy position your men, aim at the opponents and shoot (hopefully around the obstacles). This continues until one side or the other is wiped out. So far l ve become quite proficient at getting all my men killed!

This is a complex game which will require a lot of thought, planning and skill to succeed at — it should be ideal for strategy-minded wargame players.

THE WAR OF THE WORLDS

Inspired by the Jeff Wayne record of the same name, this adventure is almost totally graphic. After the Martian invasion, you are left in a desolate London with the task of tracking down your fiancee. The game follows the album quite closely — the artilleryman makes an appearance, and you have to visit six locations in the right order (the same order as in the record) on each of the six 'playing days' (this is another 'real-time' game). The Spectrum even makes a creditable attempt to play some of the music from the album.

The format of the game is similar to Valhalla: you move your stick figure across a scrolling background landscape, mapping out the London suburbs and trying to find food, drink, shelter and the special locations. Failing to eat and drink will impair your physical and mental well-being, and if you don't find a house with an open door before night falls you die of exposure.

Other hazards include fleeing mobs which sweep you away, Martian redweed, and the occasional War machine complete with heat rays. When you run into one of these, a new screen is drawn with the alien enemy towering over you. At this point you have to make a beeline for the edge of the screen, before you are reduced to a heap of charred remnants. (Don't try running underneath; CRL tell me your chances of survival are 1 in 80!). Of course, running away in panic also means you are lost . . .

The graphics are rather nice in places — the War machine is accurate and the title screens have been digitised from the album artwork. I do feel that the background representations of the houses and streets are a bit flat, though.

This game can be described as value for money, since it is going to take you a while to solve it, especially as the war machines pop up in different places and so every game is different However, some more screen prompts would have been nice. "Press any key to continue" is so much better than just sitting there and waiting for the player to guess what to do.

A tip when playing: having enjoyed the title screens and introduction the first time, make a note of the tape counter on your cassette and wind past them for subsequent games. If you don't load the game in direct in this way, it takes 10 minutes from inserting the cassette to starting to play. Another tip: if you haven't got it already, buy the album as well. It's rather good, and playing the game and the record at the same time will provide a complete audio-visual experience!

TRS~80 ARRAYSAVE

D Garvin

This Universal Array Save and load program will save any type of array with a single POKE and USR call and be up to 80 per cent faster in execution. If you're frustrated by the amount of time that it takes to save and load data arrays in BASIC then this is manna from heaven.

This program was written for the TRS-80 (and the Video Genie) Level II in order to simplify and reduce the time taken to save data arrays on tape to an absolute minimum. The end result is that UASL will reduce the time taken to save an array by up to 80% as compared with using the PRINT#-1 statement in BASIC. Any type of dimension of array (string or numeric) may be saved with a single POKE and USR call — UASL will determine the size and type of array to be saved.

UASL occupies 349 bytes and resides in low memory from 4300H to 443CH below the BASIC program storage area. It will work with any size of memory but cannot be used with Level III or DOS.

UASL is presented as both an assembler listing and as a BASIC loader. If an assembler is available, simply type in the listing as printed and produce a SYSTEM tape. UASL will execute automatically on loading, resetting the BASIC pointers and setting the MSB for the USR call.

For those without an assembler, the BASIC loader should be used. Do *not* attempt to run this program until a copy has been made on tape as before it can be run, the BASIC pointers must be reset. To do this, simply enter the following directly from the keyboard:

```
POKE 16548,60: POKE 16549,68:
NEW
```

This will probably result in an apparent syntax error but don't panic — all is well and this can be safely ignored. With the pointers reset, the program can be loaded from tape and run. The loader will set the MSB for the USR call and will end with READY prompt.

```
10 REM ** BASIC LOADER FOR UASL
20 IF PEEK(16548)<>60 THEN PRINT "YOU MUST RESET BASIC POINTER BY:
   POKE 16548,60: POKE 16549,68: NEW AND RELOAD THIS PROGRAM AGAIN":
   END
30 FOR N=17152 TO 17467: READ D: POKE N,D
40 X=X+D: NEXT
50 IF X<>34750 THEN PRINT "ERROR IN DATA": END
60 POKE 16527,67: REM ** SET MSB FOR THE USR CALL
70 CLS: END
80 DATA 255,255,62,201,50,226,65,33,60,68,34,164,64,205,74,27,62,67,
   50,143,64,195,25,26,205,127,10,62,0,50,0,67,24,8,205,127,10,62,1,
   50,0,67,229,229,253,225,6,0,58,1,67,203,39,198,3,79,175,237,66,94
   ,35,86,175
90 DATA 235,11,11,237,66,235,175,1,4,0,237,66,126,254,3,202,149,67,
   213,58,0,67,254,1,40,30,205,18,2,205,135,2,209,123,205,100,2,122,
   205,100,2,225,126,205,100,2,35,27,123,178,32,246,205,248,1,201,
   205,18,2,205
100 DATA 150,2,205,53,2,95,205,53,2,87,225,225,205,53,2,119,27,35,
    123,178,32,246,205,248,1,201,58,0,67,254,1,40,85,225,229,213,213
    ,193,237,91,214,64,205,18,2,205,135,2,123,205,100,2,122,205,100,
    2,209,225,253,126
110 DATA 0,205,100,2,253,126,1,205,100,2,253,126,2,205,100,2,253,126
    ,0,254,0,40,16,253,70,0,253,110,1,253,102,2,126,205,100,2,35,16,
    249,27,27,27,123,178,40,8,253,35,253,35,253,35,24,200,205,248,1,
    201,213,205
120 DATA 18,2,205,150,2,205,53,2,95,205,53,2,87,237,83,214,64,209,
    225,205,53,2,71,253,112,0,205,53,2,111,253,117,1,205,53,2,103,
    253,116,2,205,53,2,71,253,112,0,205,53,2,111,253,117,1,205,53,2,
    103,253,116,2,120,254,0,40,7,205,53,2,119,35,16,249,27,27,27,123
    ,178,202,55,68,253
130 DATA 35,253,35,253,35,24,207,205,248,1,201,0
```

Listing 1. The BASIC loader.

UASL is very easy to use and requires a single line of BASIC to either save or load an array. The first requirement is to set up the USR call — the entry point for saving an array is 4318H and the load entry point is 4322H. the MSB for the USR call is set by USAL if a SYSTEM tape is used or by the BASIC loader. Therefore, to set up for saving an array simply POKE 16526 with 24 or to load an array, POKE 16526 with 34. The number of dimensions in the array must be POKEd into 17153. UASL is then called with USR (VARPTR(ARAY)). For example, to save a two-dimensional string array use:

```
10000 POKE 16526,24: POKE 17153,2:
      X=USR(VARPTR(A$(0,0)))
```

To load the array use:

```
10010 POKE 16526,34: POKE 17153,2:
      X=USR(VARPTR(A$(0,0)))
```

Having set up the USR call and the number of dimensions, any number of arrays of the same dimension may be saved in quick succession by repeated USR calls. Any type of array (string or numeric) of any size or number or dimension may be saved or loaded with UASL. The only restriction being that the array' to be loaded must have been dimensioned to the same size as that which has been saved. The reduction in time taken to save or load an array will be up to 80% — the greatest improvement being with large, double precision or string arrays.

THE WORKS

- Lines 130 to 200 initialize the routine by restoring the SYSTEM command, setting the new BASIC pointers, setting the USR call MSB and returning to BASIC.
- Lines 210 to 250 are the respective entry points for the save and load sections of the program and the FLAG is set to either one or zero.
- Lines 280 to 550 calculate the size and type of array. The VARPTR function gives the start address fo the array data and is passed to UASL via the USR call and transferred to HL in lines 210 to 250. The start of the array is preceded by, a sequence of two bytes per dimension, a single byte giving the "depth" of the the dimension, a single byte

Listing 2. Assembler listing for UASL.

```
              00100        ORG  4302H
              00110 FLAG   EQU  4300H
              00120 DIMEN  EQU  4301H
4302 3EC9     00130 INIT   LD   A,201
4304 32E241   00140        LD   (41E2H),A   ;restore SYSTEM command
4307 213C44   00150        LD   HL,BASIC
430A 22A440   00160        LD   (40A4),HL   ;set new BASIC pointer
430D CD4A1B   00170        CALL 1B4AH       ;call NEW to set pointers
4310 3E43     00180        LD   A,43H
4312 328F40   00190        LD   (16527),A   ;set USR MSB
4315 C3191A   00200        JP   1A19H       ;return to BASIC
4318 CD7F0A   00210 ESAVE  CALL 0A7FH       ;SAVE entry point
431B 3E00     00220        LD   A,0
431D 320043   00230        LD   (FLAG),A    ;set FLAG to zero
4320 1808     00240        JR   CONT
4322 CD7F0A   00250 ELOAD  CALL 0A7FH       ;LOAD entry point
4325 3E01     00260        LD   A,1
4327 320043   00270        LD   (FLAG),A    ;set FLAG to 1
432A E5       00280 CONT   PUSH HL          ;save VARPTR on stack
432B E5       00290        PUSH HL          ;twice!
432C FDE1     00300        POP  IY          ;set IY to VARPTR
432E 0600     00310        LD   B,0
4330 3A0143   00320        LD   A,(DIMEN)   ;get dimension of array
4333 CB27     00330        SLA  A           ;multiply by two
4335 C603     00340        ADD  A,3         ;and add three
4337 4F       00350        LD   C,A         ;put result in C
4338 AF       00360        XOR  A
4339 ED42     00370        SBC  HL,BC       ;HL now points to no.
              00380                         ;of bytes in the array
433B 5E       00390        LD   E,(HL)      ;no. of bytes in the
433C 23       00400        INC  HL          ;array now in
433D 56       00410        LD   D,(HL)      ;DE
433E AF       00420        XOR  A
433F EB       00430        EX   DE,HL       ;no. of bytes to HL
4340 0B       00440        DEC  BC
4341 0B       00450        DEC  BC          ;BC=DIMEN*2-1
4342 ED42     00460        SBC  HL,BC
4344 EB       00470        EX   DE,HL       ;DE now contains no. of
              00480                         ;bytes in the array from
              00490                         ;VARPTR to end.
4345 AF       00500        XOR  A
4346 010400   00510        LD   BC,4
4349 ED42     00520        SBC  HL,BC       ;HL points to array type
434B 7E       00530        LD   A,(HL)      ;get array type
434C FE03     00540 CONT2  CP   3           ;is it a string?
434E CA9543   00550        JP   Z,STRING    ;yes!
4351 D5       00560        PUSH DE          ;save no. of bytes
4352 3A0043   00570        LD   A,(FLAG)    ;get flag
4355 FE01     00580        CP   1           ;is it LOAD?
4357 281E     00590        JR   Z,LOAD      ;yes!
4359 CD1202   00600        CALL 0212H       ;switch on cassette
435C CD8702   00610        CALL 0287H       ;output header
435F D1       00620        POP  DE          ;no of bytes in DE
4360 7B       00630        LD   A,E
4361 CD6402   00640        CALL 0264H       ;write E to tape
4364 7A       00650        LD   A,D
4365 CD6402   00660        CALL 0264H       ;write D to tape
4368 E1       00670        POP  HL          ;VARPTR back in HL
4369 7E       00680 SAVE   LD   A,(HL)      ;array byte to A
436A CD6402   00690        CALL 0264H       ;write byte to tape
436D 23       00700        INC  HL          ;point next byte
436E 1B       00710        DEC  DE          ;decrement byte count
436F 7B       00720        LD   A,E
4370 B2       00730        OR   D
4371 20F6     00740        JR   NZ,SAVE     ;all done?
4373 CDF801   00750        CALL 01F8H       ;yes! stop tape
4376 C9       00760        RET              ;return to BASIC
4377 CD1202   00770 LOAD   CALL 0212H       ;start tape
437A CD9602   00780        CALL 0296H       ;read header from tape
437D CD3502   00790        CALL 0235H       ;read byte
4380 5F       00800        LD   B,A
4381 CD3502   00810        CALL 0235H       ;read byte
4384 57       00820        LD   D,A         ;DE now contains no. of
              00830                         ;bytes in the array
4385 E1       00840        POP  HL          ;VARPTR to HL and sort
4386 E1       00850        POP  HL          ;out the stack!
4387 CD3502   00860 LOAD1  CALL 0235H       ;read byte
438A 77       00870        LD   (HL),A      ;store byte in array
438B 1B       00880        DEC  DE          ;decrement byte count
438C 23       00890        INC  HL          ;point to next byte
438D 7B       00900        LD   A,E
438E B2       00910        OR   D
438F 20F6     00920        JR   NZ,LOAD1    ;all done?
4391 CDF801   00930        CALL 01F8H       ;yes! switch off tape
4394 C9       00940        RET              ;return to BASIC
4395 3A0043   00950 STRING LD   A,(FLAG)    ;get flag
4398 FE01     00960        CP   1           ;is it LOAD?
439A 2856     00970        JR   Z,LOAD2     ;yes!
439C E1       00980        POP  HL          ;VARPTR to HL
439D E5       00990        PUSH HL          ;save VARPTR again
439E D5       01000        PUSH DE          ;save no. of bytes
439F D5       01010        PUSH DE          ;twice!
43A0 C1       01020        POP  BC          ;byte count to BC
43A1 ED5BD640 01030        LD   DE,(40D6H)  ;get string space pointer
43A5 CD1202   01040        CALL 0212H       ;switch on cassette
43A8 CD8702   01050        CALL 0287H       ;write header
43AB 7B       01060        LD   A,E         ;write byte
43AC CD6402   01070        CALL 0264H       ;to tape
43AF 7A       01080        LD   A,D
43B0 CD6402   01090        CALL 0264H       ;string space pointer now
              01100                         ;on tape
43B3 D1       01110        POP  DE          ;recover byte count to DE
43B4 E1       01120        POP  HL          ;VARPTR to HL
43B5 FD7E00   01130 GO2    LD   A,(IY)      ;length of string to A
43B8 CD6402   01140        CALL 0264H       ;write length to tape
43BB FD7E01   01150        LD   A,(IY+1)    ;get LSB of address
43BE CD6402   01160        CALL 0264H       ;write to tape
43C1 FD7E02   01170        LD   A,(IY+2)    ;get MSB of address
43C4 CD6402   01180        CALL 0264H       ;address now on tape
43C7 FD7E00   01190        LD   A,(IY)
43CA FE00     01200        CP   0           ;null string?
43CC 2810     01210        JR   Z,GO3       ;yes!
43CE FD4600   01220        LD   B,(IY)      ;length to B
43D1 FD6E01   01230        LD   L,(IY+1)
43D4 FD6602   01240        LD   H,(IY+2)    ;address to HL
43D7 7E       01250 LOOP3  LD   A,(HL)
43D8 CD6402   01260        CALL 0264H       ;write byte to tape
43DB 23       01270        INC  HL          ;point to next byte
43DC 10F9     01280        DJNZ LOOP3       ;any more?
43DE 1B       01290 GO3    DEC  DE
43DF 1B       01300        DEC  DE
43E0 1B       01310        DEC  DE          ;decrement byte count
              01320                         ;three times
43E1 7B       01330        LD   A,E
43E2 B2       01340        OR   D
43E3 2808     01350        JR   Z,FINI      ;end of array?
43E5 FD23     01360        INC  IY          ;no!
43E7 FD23     01370        INC  IY
43E9 FD23     01380        INC  IY          ;IY set to next element
43EB 18C8     01390        JR   GO2         ;continue loop.
43ED CDF801   01400 FINI   CALL 01F8H       ;stop tape
43F0 C9       01410        RET              ;return to BASIC
43F1 D5       01420 LOAD2  PUSH DE          ;save byte count
43F2 CD1202   01430        CALL 0212H       ;switch on tape
43F5 CD9602   01440        CALL 0296H       ;read header
43F8 CD3502   01450        CALL 0235H       ;read byte
43FB 5F       01460        LD   E,A
43FC CD3502   01470        CALL 0235H       ;read byte
43FF 57       01480        LD   D,A         ;DE contains string space
              01490                         ;pointer
4400 ED53D640 01500        LD   (40D6H),DE  ;reset pointer
4404 D1       01510        POP  DE          ;recover no. of bytes
4405 E1       01520        POP  HL          ;recover VARPTR
4406 CD3502   01530 GO5    CALL 0235H       ;read byte
4409 47       01540        LD   B,A         ;length of string to B
440A FD7000   01550        LD   (IY),B      ;store length in array
440D CD3502   01560        CALL 0235H       ;read LSB address
4410 6F       01570        LD   L,A
4411 FD7501   01580        LD   (IY+1),L    ;store LSB
4414 CD3502   01590        CALL 0235H       ;read MSB address
4417 67       01600        LD   H,A         ;HL contains address
4418 FD7402   01610        LD   (IY+2),H    ;store MSB
441B 78       01620        LD   A,B
441C FE00     01630        CP   0           ;null string?
441E 2807     01640        JR   Z,GO6       ;yes!
4420 CD3502   01650 LOOP5  CALL 0235H       ;read byte
4423 77       01660        LD   (HL),A      ;store in string area
4424 23       01670        INC  HL          ;point to next location
4425 10F9     01680        DJNZ LOOP5       ;any more?
4427 1B       01690 GO6    DEC  DE
4428 1B       01700        DEC  DE          ;decrement byte count
4429 1B       01710        DEC  DE          ;three times
442A 7B       10720        LD   A,E
442B B2       01730        OR   D
442C CA3744   01740        JP   Z,FINI2     ;end of data?
442F FD23     01750        INC  IY          ;no!
4431 FD23     01760        INC  IY
4433 FD23     01770        INC  IY          ;IY set to next element
4435 18CF     01780        JR   GO5         ;continue loop
4437 CDF801   01790 FINI2  CALL 01F8H       ;stop tape
443A C9       01800        RET              ;return to BASIC
443B 00       01810        NOP
443C 00       01820        NOP              ;BASIC will start here
41E2          01830        ORG  41E2H       ;set SYSTEM autostart
41E2 C30243   01840        JP   INIT
0000          01850        END
```

giving the total number of dimensions, a pair of bytes giving the total number of bytes in the array (including those containing the size and number of the dimensions), a two byte array name, and finally a single byte for the array type — three if it is a string array and two, four, or eight for an integer, single precision and double precision array respectively. Once you know the number of dimensions in the array, HL can be pointed to the number of bytes in the array with the following expression:

(Array start address — (number of dimensions *2+3)

The number of bytes is put into DE and then (number of dimensions*2+1) is subtracted, the end result being that DE contains the number of bytes in the array. Subtracting four from HL will now point HL to the array type. If this is three it is a string array otherwise it is numeric. Lines 560 to 940 save and load numeric arrays. Numeric arrays are stored in memory as a sequence of two, four or eight bytes per element depending on the type of array (integer, single or double precision). The total number of bytes to be saved is first written to tape followed by each byte in turn. Loading the array from tape is simply a reversal of this procedure.

● Lines 950 to 1800 save and load string arrays. String arrays are stored in memory as three bytes per element. The first byte is the length of the string followed by the address at which the string is actually stored in the string area. The pointer for the next available string space is held at 40D6H and the pointer is written to tape. This is used to reset the string storage pointer on loading the array from tape to ensure that string data from the tape will not be overwritten by subsequent string operations. This is then followed by the length of the string, the address of the string and the string itself or each element for the array. Loading of string arrays is simply a reversal of the procedure.

● Lines 1830 to 1850 set up the autostart following the loading of the SYSTEM tape.

CHARTBUSTERS

BRITAIN'S SOFTWARE

CHARTS

by the ASP Market Research Group

ARCADE

1	Sabre Wulf	Ultimate	Spectrum (-)
2	Beach Head	US Gold	CBM (2)
3	Loco	Alligata	CBM 64 (3)
4	Psytron	Beyond	Spectrum (5)
5	Omega Race	Commodore	CBM 64 (4)
6	Trashman	Quicksilva	CBM 64 (7)
7	Oblivion	Bug-Byte	Spectrum (6)
8	Son of Blagger	Alligata	CBM 64 (9)
9	Trashman	New Generation	Spectrum (10)
10	Munch Mania	Mastertronic	CBM 64 (-)

NON-ARCADE

1	Mugsy	Melbourne House	Spectrum (1)
3	Flight Path 737	Anirog	CBM 64 (2)
3	Lords of Midnight	Beyond	Spectrum (4)
4	Sphinx Adventure	Acornsoft	BBC (3)
5	The Inferno	R. Shepherd	Spectrum (6)
6	Mastermind	Commodore	CBM 64 (5)
7	Valhalla	Legend	CBM 64 (7)
8	The Fall of Rome	APS	Spectrum (9)
9	Quizmaster	Commodore	CBM 64 (10)
10	Solo Flight	US Gold	CBM 64 (8)

Compiled with the assistance of Britain's leading software distributors, including: Pinnacle, SDL, PCE, Websters, PCS and Software Centre.

SPECTRUM

1	Sabre Wulf	Ultimate (1)
2	Jet Set Willy	Software Projs (-)
3	Stop the Express	Sinclair (9)
4	Match Point	Sinclair (-)
5	Mugsy	Melbourne Hse (-)
6	Atic Atac	Ultimate (-)
7	Scuba Dive	Martech (-)
8	Fighter Pilot	Digital Integratn (-)
9	Night Gunner	Digital Integratn (-)
10	Manic Miner	Software Projs (-)

COMMODORE 64

1	Manic Miner	Software Projs (-)
2	Beachhead	Centresoft (-)
3	Solo Flight	Centresoft (-)
4	Colossus Chess	CDS (-)
5	Chuckie Egg	A & F (-)
6	Snooker	Visions (-)
7	Superpipeline	Tasket (-)
8	Megawarz	Paramount (-)
9	Chinese Juggler	Microsoft (-)
10	Revelation	Softek (-)

DRAGON 32

1	Buzzard Bait	Microdeal (-)
2	Cuthbert in Space	Microdeal (-)
3	Hungry Horace	Melbourne Hse (-)
4	Bug Diver	Mastertronic (-)
5	Dragon Chess	Oasis Software (-)
6	Mr Dig	Microdeal (-)
7	Sprite Magic	Knight (-)
8	Chuckie Egg	A & F (-)
9	Transylvanian Twr	R. Shepherd (-)
10	Hunchback	Ocean (-)

Compiled by W. H. Smith and Websters. Figures in brackets are last week's positions.

VIC-20

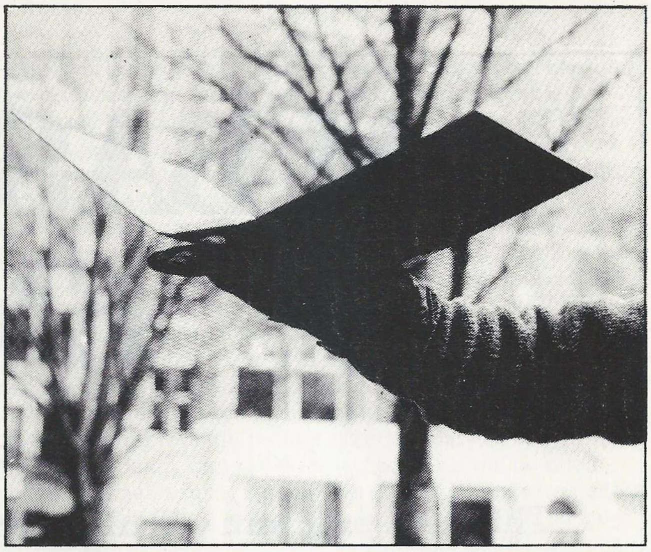
1	Computer War	Creative Sparks (-)
2	Tank Commander	Creative Sparks (-)
3	Chariot Race	Microantics (-)
4	Snooker	Visions (-)
5	Flight 015	FerrantiĆraig (-)
6	Games Designer	Galactic (-)
7	Wizard and the Princess	Melbourne House (-)
8	Grandmaster Chess	Audiogenic (-)
9	Duck Shoot	Mastertronic (-)
10	Tower of Evil	Creative Sparks (-)

BBC

1	Aviator	Acornsoft (-)
2	JCB Digger	Acornsoft (-)
3	Linkwood French	Acornsoft (-)
4	Linkwood German	Acornsoft (-)
5	Ghouls	Micropower (-)
6	Fortress	Pace (-)
7	Snooker	Acornsoft (-)
8	Snooker	Visions (-)
9	Twin Kingdom Valley	Bug-Byte (-)
10	Tree of Knowledge	Acornsoft (-)

ZX81

1	Krypton Ordeal	Novus (4)
2	Black Crystal	Carnell (9)
3	Planet Raider	Novus (8)
4	Walk the Plank	Novus (5)
5	Sabotage	Sinclair (10)
6	Space Raiders	Sinclair (-)
7	Flight Simulation	Sinclair (-)
8	Chess 1K	Sinclair (-)
9	Reversi	Sinclair (-)
10	City Patrol	Sinclair (-)



In days gone by, falconry was the sport of gentlemen and kings — this noble and time-honoured tradition is not so prevalent in these technological times, and it is quite a pity, too. Just imagine the pride you'd feel standing in your own back yard while your very own hunting falcon swooped down upon unsuspecting dogs, cats and Ford Sierras.

For a limited time only, Computing Today is offering you the chance to experience the thrill of commanding your own bird of prey, with the new CT Hunting Falcon/Magazine Binder. Swift of wing, sure of eye and made of genuine vinyl and cardboard, the Computing Today Hunting Falcon/Magazine Binder is the spitting image of the hunting birds of old to anybody suffering from cataracts. Release it from your arm, and it dives just like a traditional hawk. If it lands on a small animal, it will probably stun it. Also, when you tire of the sport, and would rather hunt hedgehogs with your Ford Sierra, your CT Hunting Falcon converts into a useful magazine binder that holds a full year's supply of Computing Today. The new CT Hunting Falcon/Magazine Binder will cost you not a farthing more than the old binder alone used to: just £5.00. This includes postage and packing, so your falcon won't have to tire itself out flying to your abode.

Cut out and send to:

COMPUTING TODAY HUNTING FALCON/MAGAZINE BINDER,
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB,
ENGLAND

I am enclosing my (delete as necessary)
Cheque/Postal Order/International Money
Order for £ . . . .
(made payable to ASP Ltd)
OR
Debit my Access/Barclaycard*
(*delete as necessary)

Insert card no. □□□□□□□□□□□□□□□□

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss) . . . .
delete accordingly
Address . . . .

. . . .

. . . .

Signature . . . .

Date . . . .

M BASIC DATA ENTRY

C. Bowden

This neat program will allow you to enter data without tears and with the minimum fuss.

With the advent of the mass-produced microcomputer, many people are now using systems at work or at home, and are writing programs for all sorts of diverse tasks. Many of these users are largely self taught, perhaps with some assistance from books and local courses. As a result, there is a tendency to adopt 'pet' methods of carrying out certain tasks when there is often a much better way of proceeding. When the better way is found, one wonders how the old way was ever acceptable.

One activity that is common to almost all data processing is the need to enter, display, update or correct records. To do this in BASIC, one can resort to the obvious method of prompting for each item of data in turn, combined with some method of input error trapping and data correction. This usually results in much scrolling of the screen display, particularly when errors are made. It is also very easy to lose track of just where you are, unless previous prompts and data are displayed. It is then difficult to see the wood for the trees. Alteration of data is also tedious, as 'Menu' lists of options need to be displayed, and multiple 'Prompts' are again needed.

After writing a number of programs in this way myself, it eventually became apparent that a much better method of data entry was possible, which permitted all of the data fields and associated prompts to be displayed on the screen and edited by cursor movement in a 'circular' manner and without scrolling. This could continue until the data was satisfactory and it could then be accepted for saving in a simple manner.

This article describes the essential details of a BASIC program that allows full on-screen editing of data fields to be carried out. The BASIC used is MBASIC Vn5, running under CP/M 2.2 but the method should be applicable to most BASICs with modification. It should also be possible to modify it for other machines, although there are a number of screen-related to hardware features that MUST be supported for the method to be used. These are discussed below, and information is given later on allow customisation for other machines that provide similar screen related commands.

The software necessary to do this task was developed as a self supporting unit that can be easily modified and added into programs. All data fields are displayed on the screen, together with a 'field description' prompt. Data is entered within a ">. . . . . <" type display. The cursor is automatically placed at the first character position within a data field when it is entered. Photo 1 is the screen display seen during editing as produced by this example program. The NAME field in the example display is shown partly completed and the underscore (__) character next to the I represents the cursor waiting for the next character or control key entry. Any period (.) characters represent unused character positions in the fields.

Essentially, the program provides the following features:

- Display does not scroll during editing of data fields.
- All input fields are displayed on the screen simultaneously with prompts and markers.
- When field is 'full', cursor automatically skips to the next field.
- Cursor may be moved up/down to any field in a 'circular' manner.
- Any data in any field may be altered. The data saved will be that displayed when the editing of the display is ended.
- Field editing is by Cursor right/left keys, overtyping, Backspace and Space.
- The Cursor cannot be moved out of fields in a 'sideways' direction.
- When the operator is satisfied, edit is ended by a simple command.
- The lower area of the screen is available for prompts and any other relevant messages (depending on number of data fields/prompts).

For the ease of use, the program listing has been split into sections and copious REM statements have been added. The main entry point is at line 310. This would be entered as a subroutine from the main program, the exit back to the main program would be from the RETURN in line 570. Lines 10 to 270 are part of the setting up process and should be at the start of any program, and only executed *once*. Lines 610 to 780 process any keystrokes, with the additional help of the subroutine lines 820 to 990.

Lines 1030 to 1130 are optional, and are intended to provide a HELP menu to the operator, as shown in Photo 1. If screen space is required for other purposes, this could easily be omitted and written instruction can be provided as an alternative. Line 1170 is another cursor positioning subroutine, used by the HELP subroutine, and also by the data verification routine lines 500 to 570.

IN OPERATION

In the sample listing, several assumptions have been made. It has been assumed that five data items will be recorded, and will comprise fields for DAY, MONTH, YEAR, NAME and STREET. It has further been assumed that these five fields will have lengths of 9, 8, 4, 8 and 12 characters respectively. The variable Y (line 40) is set to the number of data items. A number of arrays are defined in line 50 and 60 and all are set to the value of Y.

It should be noted that if it is intended to use a complicated version of this program which will run much faster, the arrays should be DIM'ed explicitly or the compiler will give error messages.

The array L stores the lengths of the data fields, array C stores the screen column at which the prompt will be printed and R stores the screen row that will be used. The array CC holds the screen column at which the Data field for each section starts.

The string array P$ stores the strings that will be printed on the screen as input prompts. It should be noted that the number of characters between the '>' and '<' characters equal the length of the data field. The '.' can be omitted if desired and space characters used instead but the dots provide a useful indication of the field length and position.

Any PRINT command in BASIC causes the operand to be sent of the screen. If the operand is a normal ASCII character it will be printed, but if it is a command to the screen handling software, the necessary action is carried out. On my system, the command to clear the screen (in BASIC) is PRINT CHR$(26). Many of the screen commands are ESCape codes, followed by a number of additional characters. Thus, Clear to End of Line is ESC followed by a "*", and Clear to End of Screen is ESC;"%". The purpose of line 270 is therefore to define 'shorthand' versions of the three 'Clear' commands so as to shorten any lines that use them later.

The command to position the cursor is slightly more complex, consisting of ESC;"=" and then two characters defining the screen position. The first one defines the ROW, the second defines the COLUMN. For my system it is necessary to add an offset of 32 to these two values. Other cards may require different offsets. Thus line 1170 is simply a general cursor positioning subroutine. It is only necessary to declare row R, and column C before calling 1170. The next BASIC 'PRINT' command will then take place at the defined position.

Examination of lines 240 and 250 will show them to be similar to 1170, but the required cursor positions are calculated from values stored in the arrays C, CC and R. The positions are calculated using a FOR-NEXT loop equal in size to the number of data items and the resulting values are stored in two cursor position arrays — CRT1$ and CRT2$. The array CRT1$ () holds the screen position at which the prompts in array P$() are to be printed, whilst the array CRT2$() holds the positions of the first character of each field. The reason that the values are calculated and stored in the arrays is to avoid the slowing up that would occur if the positions had to be calculated each time. It also shortens some later lines.

If lines 110 and 130 are examined in conjunction with lines 230-260 and 340, it will be seen that the prompts P$(1) to P$(3) will be printed starting at column 5 in rows 4,5 and 6, but P$(4) and P$(5) will be printed in column 45 of rows 4 and 5. Thus there are two data items on screen lines 4 and 5 in this example program, and one data item displayed on screen line 6. The effect is shown in Photo 1. By adjusting the values held in these arrays, altering lines 40 to 60 and the data in arrays P$, it is thus easy to move the fields about the screen and alter them.

When the program is run, line 310 ensures that MBASIC does not output any CRLF's of its own which could make a mess of the screen display. If this line is omitted, MBASIC will output a CRLF to the screen every so often depending on the value of WIDTH, which can completely ruin the screen display.

Line 320 clears the screen and leaves the cursor at the top left of the screen. The FOR-NEXT loop of lines 330 to 350 causes the cursor to be placed at each of the positions stored in array CRT1$ in turn and the appropriate prompt stored in array P$ to be PRINTed. In addition, array DD$ is cleared. (This will be dealt with later). The net result is that the five prompt strings are printed at the positions as define in lines 110 and 130. (The subroutine called from Line 390 is optional and only PRINTS any HELP messages if these are deemed necessary to help the operator.

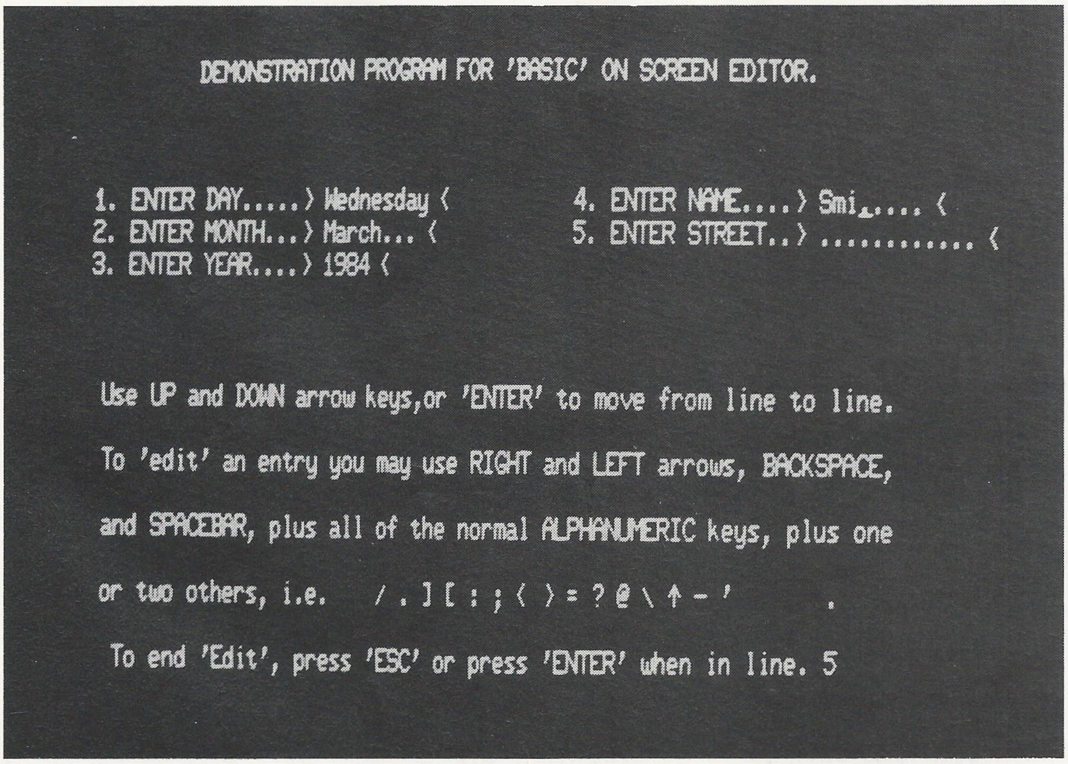
The core of the program is to be found in the loop between lines 400 and 460. The value of X set by the FOR-NEXT loop in line 400 is used to indicate the current data item. The variable W in line 410 is used to store a copy of the current value of X. The cursor is then positioned at the start of the 'current' data field as defined by CRT2$(X) and the program can then call the subroutine at line 610, which processes all keys that are pressed. This will be described in more detail later.

The cursor editing feature of this program allows the cursor to move to other data items in either direction, so it is necessary to adjust the value of X if the cursor is moved to a different data item.

The following examples will illustrate how W and X are used to ensure that the cursor moves as desired, while all data fields are continually stored. The way in which UP/DOWN arrow keys are detected and any necessary alterations are made to variable W will be described in more detail when the key processing routines are described later. For the moment it will be assumed that the necessary adjustments are made.

Suppose that the value of W and X is 5 and the cursor is in the field of data item 5. Pressing the DOWNARROW key should take the cursor to data field 1 due to the 'circular' nature of the program. Any subsequent editing should then take place on item 1 so the value of X must be adjusted accordingly. Before X is changed to 1 though, any data associated with item 5 must be saved, since it may have been altered when the cursor was on item 5.

The action is as follows: the DOWN key will be detected and the key process software will adjust W to 0, then RETurn to line 420. Ignoring line 420 for the moment, line 430 will print the most recently edited version of item DD$(5) on the screen. DD$(5) will of course hold the most recent version of field 5. Since Z is not 27 (ESC), line 450 has no effect. Line 460 will set X equal to W (0); NEXT X will then set X to 1. Execution can then proceed from line 410. W will be set to equal X (1), and editing will take place on data field 1, as required.

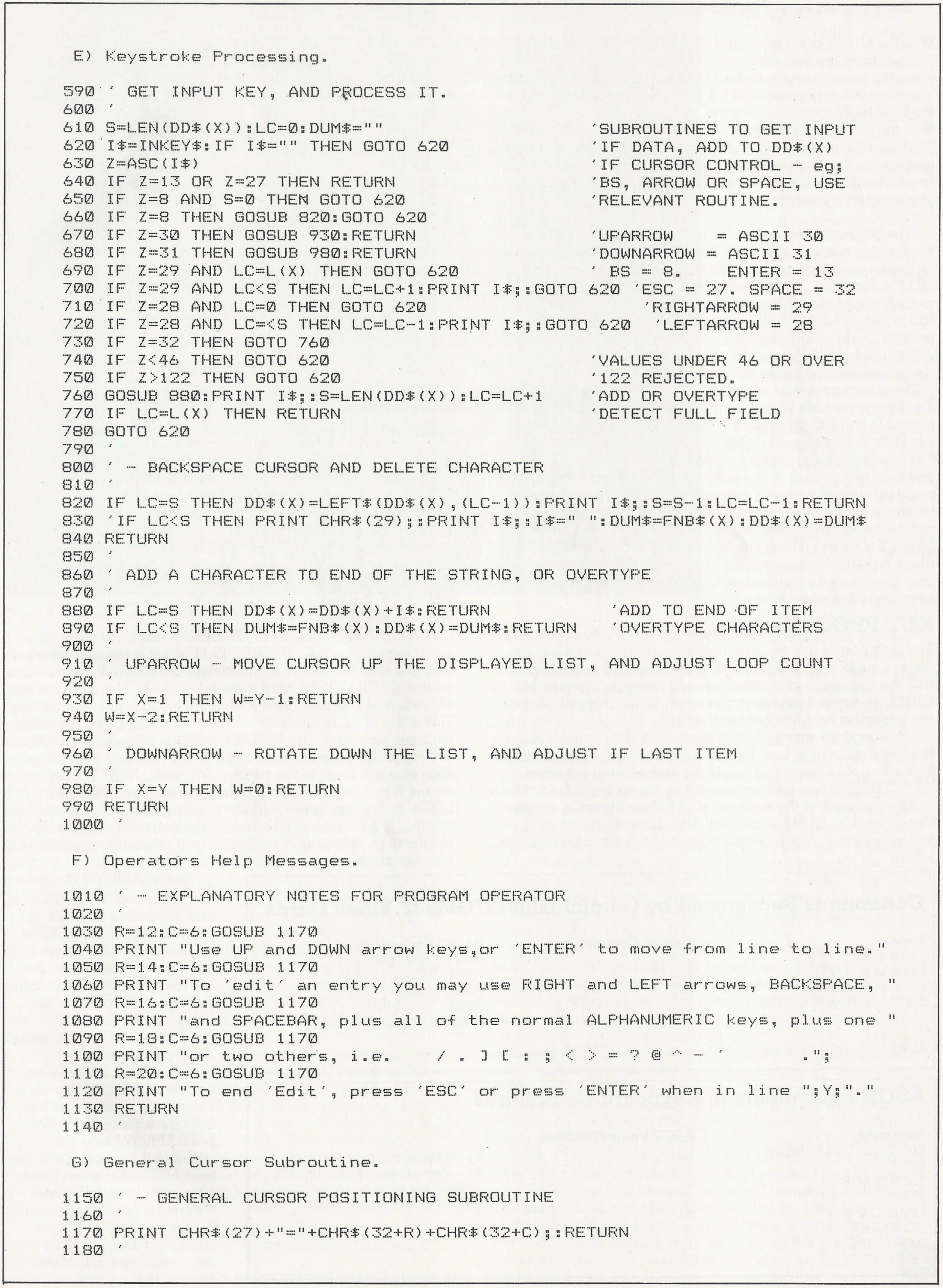


```
           DEMONSTRATION PROGRAM FOR 'BASIC' ON SCREEN EDITOR.



    1. ENTER DAY.....) Wednesday (           4. ENTER NAME....) Smi_.... (
    2. ENTER MONTH...) March... (            5. ENTER STREET..) ............ (
    3. ENTER YEAR....) 1984 (



    Use UP and DOWN arrow keys,or 'ENTER' to move from line to line.

    To 'edit' an entry you may use RIGHT and LEFT arrows, BACKSPACE,

    and SPACEBAR, plus all of the normal ALPHANUMERIC keys, plus one

    or two others, i.e.   / . ] [ : ; ( ) = ? @ \ ↑ - '       .

     To end 'Edit', press 'ESC' or press 'ENTER' when in line. 5
```

Photo 1 Display as produced by example program.

Listing 1 The MBASIC program.

```
 A) General setting up.

10 ' - SCREEN EDITOR FOR MBASIC AND CURSOR ADDRESSABLE V.D.U.
20 ' Copyright - C. Bowden                           Jan 84.
30 '
40 Y=5                                        'SET  NUMBER  OF  DATA ITEMS
50 DIM C(Y),L(Y),R(Y),S(Y),CC(Y),P$(Y)
60 DIM DD$(Y),CRT1$(Y),CRT2$(Y)
70 DEF FNB$(X)=LEFT$(DD$(X),LC)+I$+RIGHT$(DD$(X),(S-LC-1))
80 '
90 ' - SET UP ALL CONSTANTS INTO ARRAYS FOR EASY ACCESS
100 '
110 C(1)=5:C(2)=5:C(3)=5:C(4)=45:C(5)=45       'SET PRINT COLUMN
120 L(1)=9:L(2)=8:L(3)=4:L(4)=8:L(5)=12        'SET INPUT STRING LENGTHS
130 R(1)=4:R(2)=5:R(3)=6:R(4)=4:R(5)=5         'SET SCREEN ROWS
140 CC(1)=24:CC(2)=24:CC(3)=24:CC(4)=64:CC(5)=64 'SET DATA INPUT COLUMNS
150 P$(1)="1. ENTER DAY     > ......... <"     'DEFINE SCREEN PROMPTS
160 P$(2)="2. ENTER MONTH   > ........ <"
170 P$(3)="3. ENTER YEAR    > ....  <"
180 P$(4)="4. ENTER NAME    > ........ <"
190 P$(5)="5. ENTER STREET > ............ <"
200 '
210 ' - PRESET SCREEN POSITIONS AND SHORTEN SOME SCREEN COMMANDS
220 '
230 FOR X=1 TO Y                              'PRESET POSITIONS FOR
240 CRT1$(X)=CHR$(27)+"="+CHR$(32+R(X))+CHR$(32+C(X)) ' SCREEN
250 CRT2$(X)=CHR$(27)+"="+CHR$(32+R(X))+CHR$(32+CC(X))
260 NEXT X
270 CLS$=CHR$(26):CLN$=CHR$(27)+"*":CLE$=CHR$(27)+"%" 'SCREEN COMMANDS
280 '

 B) Main Edit Module.

290 ' - MAIN PROGRAM. CLEAR SCREEN, PRINT PROMPTS, CLEAR INPUT ARRAYS
300 '
310 WIDTH 255                                'STOP BASIC ADDING CRLF'S
320 PRINT CLS$                               'CLEAR SCREEN
330 FOR X=1 TO Y                             'PRINT PROMPT STRINGS AND
340 PRINT CRT1$(X);:PRINT P$(X);:DD$(X)=""   'CLEAR INPUT ARRAY
350 NEXT X
360 '
370 ' - POSITION CURSOR AT INPUT ITEM X, GET AND PROCESS KEYSTROKES
380 '
390 GOSUB 1030                               'PRINT 'HELP' MESSAGES

 C) Edit Loop.

400 FOR X=1 TO Y                             'NOW LOOP, SETTING CURSOR
410 W=X:PRINT CRT2$(X);:GOSUB 610            'TO INPUT POSITION, AND
420 IF X=Y AND LC=L(X) THEN W=0              'READING KEYS. ADD KEYS
430 PRINT CRT2$(X);DD$(X);                   'TO DATA STRINGS, FILTER
440 '                                        'CURSOR AND COMMAND KEYS
450 IF Z=27 THEN GOTO 500                    'UNTIL "ESC" (OR "ENTER"
460 X=W:NEXT X                               'FROM LAST LINE.)
470 '

 D) Verify Edit.

480 ' VERIFY THAT DISPLAYED DATA IS O.K. - LOOP IF NOT
490 '
500 R=12:C=6:GOSUB 1170:PRINT CLE$;
510 PRINT "Any Key except 'Y' or 'y' will Re-enter 'EDIT' mode. ";
520 R=14:C=6:GOSUB 1170
530 INPUT "Is the Data correct - ";RES$       'VERIFY ENTERED DATA
540 IF RES$="Y" OR RES$="y" THEN GOTO 560
550 GOTO 390                                 'IF NOT O.K., DO AGAIN
560 WIDTH 80                                 'IF O.K. RETURN TO PROG.
570 RETURN                                   'END OF EDITING ROUTINE
580 '
```



```
 E) Keystroke Processing.

590 ' GET INPUT KEY, AND PROCESS IT.
600 '
610 S=LEN(DD$(X)):LC=0:DUM$=""                     'SUBROUTINES TO GET INPUT
620 I$=INKEY$:IF I$="" THEN GOTO 620               'IF DATA, ADD TO DD$(X)
630 Z=ASC(I$)                                      'IF CURSOR CONTROL - eg;
640 IF Z=13 OR Z=27 THEN RETURN                    'BS, ARROW OR SPACE, USE
650 IF Z=8 AND S=0 THEN GOTO 620                   'RELEVANT ROUTINE.
660 IF Z=8 THEN GOSUB 820:GOTO 620
670 IF Z=30 THEN GOSUB 930:RETURN                  'UPARROW     = ASCII 30
680 IF Z=31 THEN GOSUB 980:RETURN                  'DOWNARROW = ASCII 31
690 IF Z=29 AND LC=L(X) THEN GOTO 620              ' BS = 8.    ENTER = 13
700 IF Z=29 AND LC<S THEN LC=LC+1:PRINT I$;:GOTO 620 'ESC = 27. SPACE = 32
710 IF Z=28 AND LC=0 THEN GOTO 620                      'RIGHTARROW = 29
720 IF Z=28 AND LC=<S THEN LC=LC-1:PRINT I$;:GOTO 620   'LEFTARROW = 28
730 IF Z=32 THEN GOTO 760
740 IF Z<46 THEN GOTO 620                          'VALUES UNDER 46 OR OVER
750 IF Z>122 THEN GOTO 620                         '122 REJECTED.
760 GOSUB 880:PRINT I$;:S=LEN(DD$(X)):LC=LC+1      'ADD OR OVERTYPE
770 IF LC=L(X) THEN RETURN                         'DETECT FULL FIELD
780 GOTO 620
790 '
800 ' - BACKSPACE CURSOR AND DELETE CHARACTER
810 '
820 IF LC=S THEN DD$(X)=LEFT$(DD$(X),(LC-1)):PRINT I$;:S=S-1:LC=LC-1:RETURN
830 'IF LC<S THEN PRINT CHR$(29);:PRINT I$;:I$=" ":DUM$=FNB$(X):DD$(X)=DUM$
840 RETURN
850 '
860 ' ADD A CHARACTER TO END OF THE STRING, OR OVERTYPE
870 '
880 IF LC=S THEN DD$(X)=DD$(X)+I$:RETURN             'ADD TO END OF ITEM
890 IF LC<S THEN DUM$=FNB$(X):DD$(X)=DUM$:RETURN     'OVERTYPE CHARACTERS
900 '
910 ' UPARROW - MOVE CURSOR UP THE DISPLAYED LIST, AND ADJUST LOOP COUNT
920 '
930 IF X=1 THEN W=Y-1:RETURN
940 W=X-2:RETURN
950 '
960 ' DOWNARROW - ROTATE DOWN THE LIST, AND ADJUST IF LAST ITEM
970 '
980 IF X=Y THEN W=0:RETURN
990 RETURN
1000 '

 F) Operators Help Messages.

1010 ' - EXPLANATORY NOTES FOR PROGRAM OPERATOR
1020 '
1030 R=12:C=6:GOSUB 1170
1040 PRINT "Use UP and DOWN arrow keys,or 'ENTER' to move from line to line."
1050 R=14:C=6:GOSUB 1170
1060 PRINT "To 'edit' an entry you may use RIGHT and LEFT arrows, BACKSPACE, "
1070 R=16:C=6:GOSUB 1170
1080 PRINT "and SPACEBAR, plus all of the normal ALPHANUMERIC keys, plus one "
1090 R=18:C=6:GOSUB 1170
1100 PRINT "or two others, i.e.    / . ] [ : ; < > = ? @ ^ - '       .";
1110 R=20:C=6:GOSUB 1170
1120 PRINT "To end 'Edit', press 'ESC' or press 'ENTER' when in line ";Y;"."
1130 RETURN
1140 '

 G) General Cursor Subroutine.

1150 ' - GENERAL CURSOR POSITIONING SUBROUTINE
1160 '
1170 PRINT CHR$(27)+"="+CHR$(32+R)+CHR$(32+C);:RETURN
1180 '
```

There are two ways of ending the edit:

- If the ENTER key is pressed during edit, it will cause the cursor to move on to the next data item in numerical order. If this is done when the cursor is on the highest numbered data item, eg 5, then when line 460 is reached NEXT X will make X equal to 6 and then the FOR-NEXT loop will terminate, ending the edit.
- If the ESC key is pressed while editing, the value of Z will be 27. This will cause the program to RETurn from the key processing routines, and will be detected by line 450. Thus the edit is terminated irrespective of the position of the cursor, although all displayed data is saved.

The program lines 500 to 570 provide a means of reversing the decision to end an edit. Line 500 positions the cursor in column 6 on screen line 12 and then clears to end of screen, removing the 'HELP' information. The operator is then prompted to enter 'Y' or 'y' to confirm end of edit, in which case the program proceeds to line 560 to restore MBASIC 'WIDTH' to 80 and returns to the main program. Entry of any other key will loop back to line 390 for printing of the 'HELP' messages again, followed by the repositioning of the cursor at the start of the first data item.

The observant reader may be wondering why the program uses the dummy variable W, and why the array variable DD$(X) is printed in line 430. After all, any editing is immediately placed into DD$(X) and echoed on the screen, which seems to make line 430 pointless. It also seems pointless to store a copy of X in W, and to transfer it back in line 460, when the current value of X could be changed directly in lines 910 to 990. This is perfectly true in the example listing. In a more sophisticated program, it may be necessary to carry out some more processing on DD$(X) *before* X is altered. This is true of the program displayed in photo 2, where right justification and padding of the data occurs. Extra lines to do this can be inserted 420 and 430. If this is done, the extra code described is needed.

KEY PROCESSING

The only part of the program not yet considered in detail are the key stroke processing subroutines, lines 610 to 990. In line 610 the number of characters already stored in the variable DD$(X) is counted and stored in variable 'S'. This will be zero if the particular field has not been edited.

A second counter — LC, is set to zero. This counter is used to store the count of how far along the data field the cursor is. The value of LC cannot exceed the count S, but note that 'SPACE' characters may be present at the end of a field, which will be included in the total value of S. In addition, a dummy string variable, DUM$ is cleared. This dummy is used as a temporary store during certain editing functions to be described later.

Incidentally, it is easy to use this routine to update previously entered records. The data should be read into the array DD$() before calling the Edit software. The logic around line 340 should be altered such that when used in this way, the array DD$() is NOT cleared. Also arrays CRT2$() and DD$() should be printed at this time, so that the screen displays the data that is to be edited.

Line 620 simply loops until a key is pressed. Any key pressed is then stored in the string variable I$. The ASCII value of the key is computed and stored in variable Z, to permit easy filtering of command keys or data keys. Line 640 traps ENTER or ESC keys, either of which will cause an immediate RETurn to the main edit loop at line 420.

Line 650 stops the BACKSPACE key from having an effect if the cursor is in the first column of the data field. If B/S is pressed from further along the field, the operation is valid so the program uses subroutines at 820 to 840. If LC equals S (line 820) then the cursor is on the last character of the field and a simple B/S is performed. String DD$(X) is shortened by one character, S and LC are decremented by one, and the B/S is PRINTed on the screen as a 'space', causing the last character displayed to be deleted.

If the cursor is part way along the data item (ie LC<S — line 830) then the cursor is moved back one character and a B/S is printed on screen, causing a space to appear within the displayed field. LC and S are not adjusted. The variable I$ is then set to a 'SPACE' character, and the DEFINED FUNCTION FNB$(X) is used (line 70). This function is used to replace characters within the string DD$(X). Thus a SPACE character will be placed into DD$(X) at the correct position. The program will then return to 660, and then to 620 for another key. At this stage, the cursor is at the space created within DD$(X), and any other non-control character typed will be placed into this space as described below for character insertion. As an alternative, the cursor can be moved elsewhere, leaving a space in the string.

Line 670 will process any UPARROW keys, via the subroutine at line 930. UPARROW must not terminate the whole edit, but only the edit of the particular data item. If the cursor is on item 1, UPARROW must take the cursor to item 5. Since line 460 will add one to X, then X must be set to 4 before returning. This is the effect of line 930, where W is set to Y−1 (ie 4 in this example program). The RETURN in 930 is followed by another in 670, so the program goes back to line 430 and then on to 460, where X is set to the value of W, (ie 4). NEXT X then makes X = 5, which is the required value. The FOR-NEXT LOOP is thus not terminated, and editing continues in item 5 as desired. If X is not equal to 1, then the correct result can be obtained by reducing W by 2. The part played by variable W has already been described.

Commands Recognised by Gemini GM812/GM832 Video Cards

Action	Gemini Command	BASIC Equivalent
Clear Screen	01A Hex	PRINT CHR$(26)
Clear to End of Line	ESC,"*"	PRINT CHR$(27) ; "*"
Clear to End of Screen	ESC,"%"	PRINT CHR$(27);"%"
Set Cursor Row-R, Col-C	ESC, "=",R+32,C+32	PRINT CHR$(27);"="; CHR$(32+R);CHR$(32«C)

Many commands used are similar to Lear-Seigler ADM-3A commands.

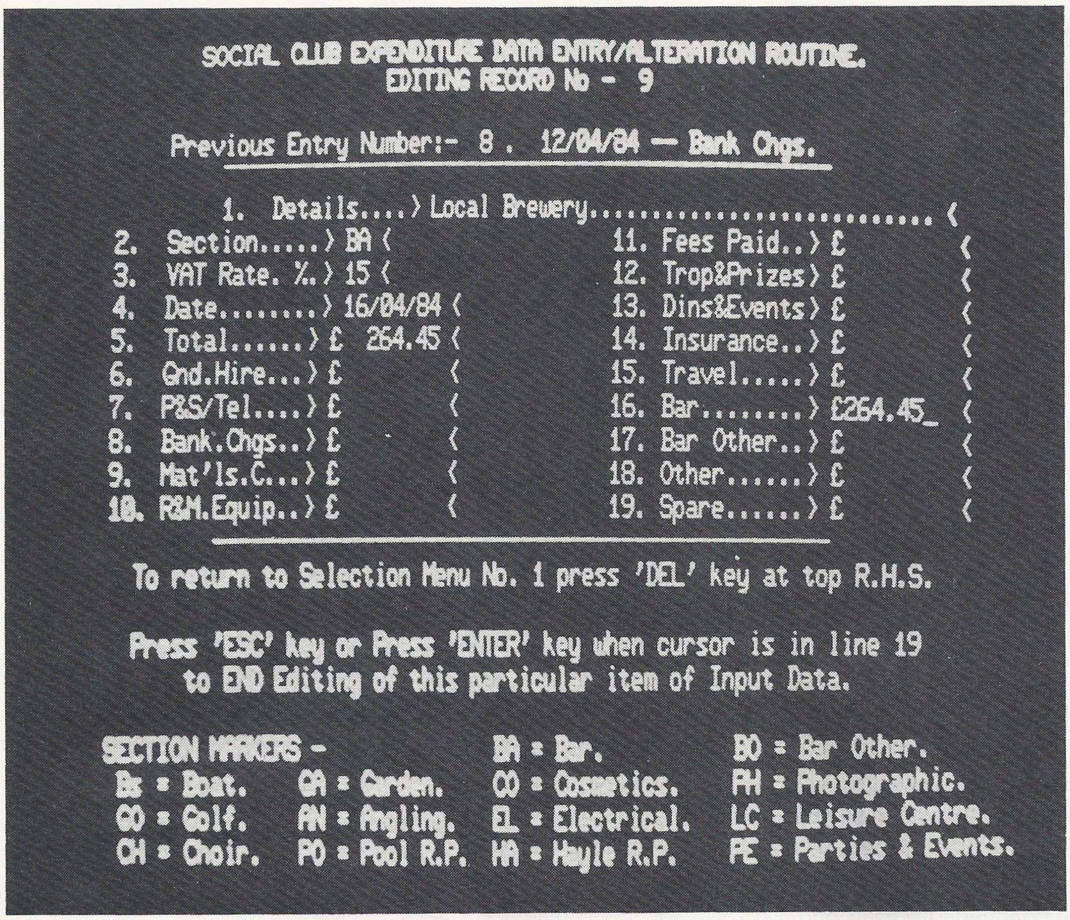
ASCII Equivalents to Keyboard Commands

Keystroke	ASCII Value (Decimal)
ENTER/RETURN/NEWLINE	13
ESC	27
BACKSPACE	08
SPACE	32
UPARROW KEY	30
DOWNARROW	31
RIGHTARROW	29
LEFTARROW	28

If your keyboard does not have arrow keys, then you may be able to use some other combination to generate these or other codes.

DOWNARROW keystrokes are processed in line 680, and subroutine 980,990. Adjustments similar to those for the UPARROW key are made, if the cursor is in the last (highest numbered) data field, as described earlier. Otherwise no action is needed as NEXT X will make X the correct value.

RIGHTARROW keys are prevented from having an effect if the cursor is on the last character in the field — LEFT ARROW keys are inhibited if the cursor is in the first character position (lines 690, 710). Line 700 allows rightwards cursor movement if the cursor is not at the right-hand end of a data item, while line 720 allows



```
        SOCIAL CLUB EXPENDITURE DATA ENTRY/ALTERATION ROUTINE.
                       EDITING RECORD No -  9

          Previous Entry Number:-  8 .  12/04/84 — Bank Chgs.

        1.  Details....> Local Brewery............................ <
   2.  Section.....> BA <                  11. Fees Paid..> £          <
   3.  VAT Rate. %..> 15 <                 12. Trop&Prizes> £          <
   4.  Date........> 16/04/84 <            13. Dins&Events> £          <
   5.  Total......> £  264.45 <            14. Insurance..> £          <
   6.  Gnd.Hire...> £          <           15. Travel.....> £          <
   7.  P&S/Tel....> £          <           16. Bar........> £264.45_   <
   8.  Bank.Chgs..> £          <           17. Bar Other..> £          <
   9.  Mat'ls.C...> £          <           18. Other......> £          <
   10. R&M.Equip..> £          <           19. Spare......> £          <

    To return to Selection Menu No. 1 press 'DEL' key at top R.H.S.

    Press 'ESC' key or Press 'ENTER' key when cursor is in line 19
         to END Editing of this particular item of Input Data.

   SECTION MARKERS -              BA = Bar.          BO = Bar Other.
    Bs = Boat.    GA = Garden.    CO = Cosmetics.    PH = Photographic.
    GO = Golf.    AN = Angling.   EL = Electrical.   LC = Leisure Centre.
    CH = Choir.   PO = Pool R.P.  HA = Hayle R.P.    PE = Parties & Events.
```

Photo 2 Typical screen during editing of a real program.

leftward movement within a data item. In each case, the cursor command is PRINTed on the screen, so that the cursor is seen to move in response to the keystroke and the cursor position counter LC is adjusted. After RIGHT/LEFT arrow keys have been processed, the program goes back to line 620 for the next keystroke.

SPACE characters are allowed within data items (in this example program, also at the end of data items). Line 730 'traps' spaces, and diverts via line 760 to the subroutines at 880, 890 which allow overtyping or addition of characters. If LC is equal to S, then the cursor is at the end of the string, and so the SPACE is added to the string variable DD$, and then counters LC and S are adjusted. The character is also PRINTed. If the cursor is not at the end of the string (LC<S — line 890), then the DEFINED FUNCTION FNB$(X) is used as before to place the 'SPACE' within the string.

At this stage, all valid commands have been filtered out, so lines 740 and 750 trap any other non-valid characters. In this program, any characters within the range 46 to 122 are acceptable as data, so line 760 will send them to be processed exactly as for the 'SPACE' character just described.

If the last character added did not fill the data field, then line 780 takes the program back to line 620 for the next keystroke.

If the character that was added to a data item has resulted in the length of the item filling the alloted field, then line 770 will detect the full field and will terminate data entry for that item. The program will return to line 430, to process the data as described. The cursor will move on to the next higher data field, due to the action of NEXT X in line 460. Note, however, that if this should happen in the highest numbered data field (5 in our example), the FOR-NEXT loop could terminate and end the edit with the usual prompts of lines 500 etc. Line 420 prevents this from occuring.

I hope that you will adjust this program to your own requirements — you can add many extra features that will improve its performance.

In one Social Club Accounts program there are around twenty data fields to define different sectional activites, descriptive information, dates, and a number of financial items. Money amounts are entered as string data, and are then automatically right justified in the field, with one or two trailing zeros added after the decimal point, depending on the pence value, to give a neat tabulation. The 'Total' in brackets as seen in Photo 2 has been justified in this way, but the 'Bar' field has not, since the cursor is still in the field.

The use of a Section marker allows separate accounts to be printed for different sections, and VAT is calculated at the rate entered in the VAT field. The account is printed on a MX100 printer using compressed print and 232 columns.

When a new record is entered, the program asks if the data is the same as for the last entry, and if so, automatically fills in the data field, including "/" characters. In addition, the date, description and record number of the last entered record are displayed at the top of the screen as an aid to the operator. Another feature allows use of the DEL key to end editing without saving the displayed data. Photo 2 is a 'dump' of the screen display during editing with this program. The system I use at work consists of a GEMINI Galaxy 2 running CP/M 2.2, with twin 400K discs and 256K virtual disc, and an EPSON MX100 printer. At home I use a similar system — a hybrid NASCOM-GEMINI 80BUS multiboard computer, with twin 350K discs, 256K virtual disc, video card, programmable keyboard and a number of other I/O and Colour cards, and an EPSON FX80 printer. Several operating systems are available, but CP/M 2.2 is the normal system used. (On both systems Sys/Gemini CBIOSs are used, which have a number of unusual features that greatly improve the 'user friendliness' of CP/M. In particular these BIOSs allow FULL 'on screen' CURSOR editing of many CP/M Utility command lines, and in particular MBASIC lines including line numbers. This feature is most unusual on CP/M machines, and makes editing a simple job.)

The video card (Gemini IVC) is an intelligent video card with its own Z80, and appears to the host as three I/O ports. This card plugs into the bus, so access is very rapid. It recognises a large number of commands, but those concerned with the subject of this article are few in number and probably common to most CP/M machines.

If your machine has cursor addressing and a few common screen commands such as 'Clear to end of line', 'Clear to end of screen', 'Clear screen', then you can use the type of routine that is described in this article. Below are listed the commands required by the Gemini IVC card for the various operations. These can be altered for similar video systems. The software assumes 25 by 80 screen format, and this should be taken into account for other formats. Ideally the keyboard should also have 'arrow' keys that can be used to control cursor movement. The necessary codes can often be generated by other key combinations if this is not the case. There is no reason that codes other than 28 to 31 be used for this purpose, unless it clashes with some other allocation for the chosen code.

CLUB CALL

Fiona Eldridge

OXON TI USERS

29 Kestrel Crescent
Blackbird Leys
Oxford OX4 5DY

Contact: Peter Brooks (Organiser)
Tel: Oxford (0865) 717985

This newly-formed group is dedicated to users of the Texas Instruments range of home computers (both TV standards). Membership is not restricted to Oxfordshire users only (although they are subsidised) and communication is via a monthly newletter, TI-LINES.

TI-LINES aims to be a veritable panacea to TI users and provide detailed information on subjects such as the use of control and function keys in TI BASIC and the application of pitch-excited linear predictive coding in the speech synthesiser! TI-LINES has also run a BASIC tutorial and A TMS9900 assembly language course, some of the material being presented in TI.MES, the quarterly newsletter of TI-99/4A Exchange, a national user group. The newsletter may be unique in that it can be read onto audio tape for the benefit of blind or partially sighted users.

For TI users in Oxfordshire, membership is 12 x 12½p stamps, for those outside the country, membership costs £10 pa and new members receive a starter pack of the first two issues. For further information, contact Peter Brooks on the number above.

NORTH-WEST TRS-80 USERS GROUP

40 Cowless
Westhoughton
Bolton
BL5 3EG

Contact: D F Franklin (Secretary)

This group aims to promote the use of TRS-80, Dragon and Video-Genie (non-colour) computers in NW England. Meetings are held on the last Wednesday of every month at 7.30 pm at the Lancashire Aeroclub HQ, Barton Airport. There are usually 60 to 80 members present and a general get-together is followed by a lecture or demonstration.

Subscription costs £8 pa which include copies of Rem 80, a bi-monthly newsletter with articles, hints and tips written by members, access to two hire libraries and a repair and modification service.

NORFOLK GENIE AND TRS-80 USER GROUP

Contact: P A Golder
Tel: Swanton Morley (0362-83) 491

NOGATUNG is a group dedicated to the Video Genie and TRS-80 range of computers. A subscription fee of £5 pa includes meetings on the third Wednesday of each month at the Crome Community Centre in Norwich, lectures and help and advice.

Membership is increasing all the time and includes students, doctors, businessmen and servicemen with a wide range of interest and experience in the field of computing. For further details, give Paul Golder a ring on the number above.

INDEPENDENT COLECO ADAM USERS CLUB

PO Box 9
Towcester
Northants NN12 7QG

Contact: David Winnett
Tel: Towcester (0327) 50705

This is a newly-formed user group dedicated to the Coleco Adam computer system. As well as area meetings around the country, the club aims to provide members with news, reviews, articles, program listings, hints and tips and general advice via a regular newsletter (ten issues pa).

Users, owners and potential owners are welcome to write to the club at the address above and prospective members are asked to include an sae if they would like membership details.

PENCOED AMATEUR COMPUTER CLUB

7 Duffryn
Pencoed
Bridgend
Mid Glamorgan CF35 6JL

Contact: B Pugh (Secretary)
Tel: Pencoed (0656) 86231

A well established group, the Pencoed Amateur Computer Club was formed in 1982 with the aim of 'familiarising and educating people in the use of home computers'. Meetings are on every other Wednesday at the Pavillion, Felindre Road, Pencoed between 6 and 9 pm where there are demonstrations of computers and peripherals and talks on applications as well as a library of books, educational tapes and magazines available on loan to members. Future meetings include visits to computer and chip factories and lectures from professionals on computer related subjects.

New members are welcome and subscription is £6 pa or £9 pa for family membership.

GRAVESEND COMPUTER CLUB

58 Apseldene
Hever Farm Estate
Singlewell
Gravesend
Kent DA12 5EE

This club has recently moved premises to the Council Tenants Club, Whitehill Lane, Gravesend. Meetings are every Thursday at 7 pm and wide variety of home computers are available. For further details contact the club secretary at the above address.

LYNX USERS GROUP

209 Kenton Lane
Kenton
Harrow
Middx HA3 8TL

Contact: RB Jones
Tel: 01-907 3406

This is a new user group/magazine which has taken over from NILUG as a spokespiece for Lynx owners. If you're interested in promoting and retaining interest in the Lynx, contact R B Jones at the address above.

BLOXWICH COMPUTER CLUB

64 Nursery Road
Bloxwich
Walsall
W Mids WS3 2DU

Contact: Mo Warden (Chairman)

This is a local computer group meeting every Wednesday at the frank F Harrison Comprehensive, Leamore Lane, Bloxwich at 7 pm. Meetings are held in the computer room and members have access to the school's range of Tandy and BBC machines as well as computers brought to the meetings.

There is no membership fee but members are asked to pay an admission fee of 25p per session. There is a small but enthusiastic membership and new members are welcome to attend.

EAST LONDON AMATEUR COMPUTING CLUB

Contact: Jim Turner
Tel 01-558 3681

This club meets on the second and fourth Tuesday of the month (temporarily) at the Leytonstone Branch Library at 7 pm. The first meeting of the month is usually a lecture or tutorial and the second meeting is a 'free' evening where members can exchange news, views and advice on hardware and software. There is always a great variety of computers at meetings, both commercial and home made and interest varies from games to high level languages such as FORTH, Pascal and COBOL as well as BASIC and machine code.

In cooperation with the Youth Outreach Service in Leytonstone, some members taught computing during the 1983 school holidays and will be doing so again this year. Other computer-related community services are also planned.

The club has about 85 regular members ranging from schoolchildren to pensioners and beginners to experts. Subscription cost £6 pa for adults or £4 for juniors and pensioners. Visitors are always welcome and the only charge is for refreshments. For further details, ring Jim Turner on the number above.

CLUB TIO SINCLAIR

Blas Cabrera, 67
Arrecife
Las Palmas
Spain

Contact: Tony Gubern Soyka

If you are interested in "knowing everything about your ZX machine" then this club is for you! Members are welcomed from all levels of experience and for a subscription fee of 3000 pesetas (about £15) will receive regular bulletins with news, features on Machine Code, BASIC and Spectrum UDGs and listings and an ad section whereby members can buy and sell anything of interest. Members can also enjoy discounts on programs.

Anyone interested in jouning the club should contact Tony at the above address.

BRITISH OSBORNE OWNERS' GROUP

Flat 19, Rowan House
Mitton Road
Handsworth
Birmingham B20 2JR

Contact: Gaynor Anglesea (Secretary)

Boog is a national group completely independent of Osborne and exists to offer help, advice and a vast library of public domain software available to members for a small hire fee. Members also receive a 10% discout on software, hardware and other services from Osborne UK and there are other "advantageous deals" available with servicing companies.

As well as regular national meetings, there are a number of local groups being established throughout the country. Membership costs £18 pa and enquiries should be addressed to Mrs Gaynor Angelsea at the address above. Software library enquiries to Robin Auld, 44 Granshaw Close, Kings Norton, Birmingham B38 8RA.

CLUB 64

65 Upper Drumcondra Road
Dublin 9
Contact: Brendan Conroy

Although Club 64 is a Dublin based CBM64 sotware user group, the majority of members come from England and Northern Ireland as well as Scotland, Canada, USA, New Zealand, Australia and the Republic of Ireland. As yet the membership is very much male dominated but it is hoped that it will become more equally balanced in the future. Club 64 aim to supply public domain software on disc at the lowest possible cost — this means £12 membership pa plus £4 per disc (between 10 and 25 programs). The price includes postage and packing to anywhere in the UK, a surcharge of £1 per disc covers postage to other countries. If enough new members join, they hope to be able to issue at least two new discs per month.

Club 64 is particulary interested in serving the needs of users who cnnot attend club meetings whether it be through living in remote areas, abroad or disablement. As a large selection of educational software is available to the group, teachers and educational organisations are invited to contact them. For any further details, contact Brendan at the above address above.

ZX EXCHANGE

4 Hurkur Crescent
Eyemouth
Berwickshire
Scotland TD14 5AP

Contact: Nick Godwin
Tel: Eyemouth (0390) 50965

ZX Exchange is a worldwide group of users dedicated to the ZX81, TS-1000 and TS-1500 who exchange programming expertise and information about all aspects of the "little wonder". With contacts in the USA, Hong Kong, Mexico, Australia, Eire, The Netherlands and Britain, communication is via a bimonthly newspaper, ZX Broadsheet. Apart from routines, the newletter also regularly includes Exchange Profile, a column in which people can say something about themselves, their equipment and their interests. Intermittently, the newletter also features a magazine list, giving information about publications that regularly deal with the ZX81 and reviews of hardware and software.

Details of ZX Exchange and a trial issue of the newsletter can be obtained by sending 60p or four International Relply Coupons to Nick Godwin at the address above.

If you are interested in using your computer in connection with model railways, why not write to Ian Bertram (including a sae) at 16 Tynedale Avenue, Whitley Bay, Tyne and Wear NE26 3BA. Ian is hoping to set up a user group and would welcome suggestions or ideas for articles to put in a members' newsletter. For a more rapid reply, anyone on Micronet can contact Ian on Mbx 632526429.

If you would like details of your/ club user group on these pages, write to me at:

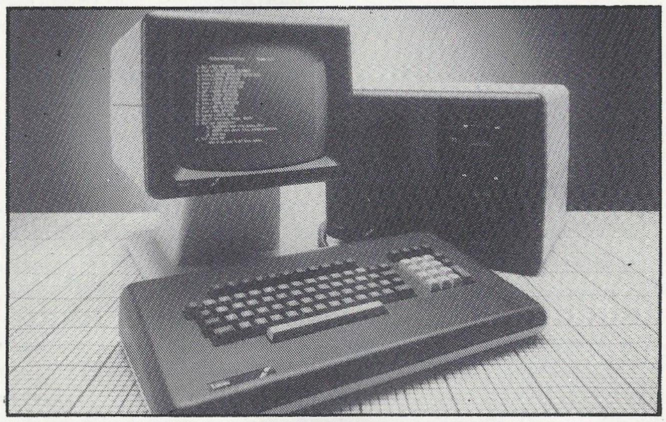
Club Call
1 Golden Square
London W1R 3AB

LUCAS MICRO

LUCAS LX

MEMORY	64K RAM expandable to 256K
LANGUAGE	Microsoft BASIC
CASSETTE	300 or 1200 baud
DISC	Single or twin 5¼ floppy disc drives DOS CP/M 2.2 (supplied) or NAS-DOS
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☑ FUNCT ☑
DISPLAY	TV ☑ MONITOR ☑ SUPPLIED ☑
INTERFACE	PARA ☑ SERIAL ☑ BUS ☑
GRAPHICS	BLOCK ☑ USER ☑ LINE □ RES 392 by 256 COLOUR 8 TEXT 80 by 25

Notes. The Lucas LX is a Z80A microcomputer aimed more at the professional and business user. Hence 5Mb Winchester disc interfacing is provided. Popular printers may be used with the RS232 serial interface, and a Centronics interface is also provided. There is an additional parallel interface connector for providing up to 16 on/off signals. The monitor supplied as standard is a 12″ monochrome version: a colour monitor is also available. The high res colour graphics may be 392 by 256 in eight colours, or 784 by 256 in two colours. A wide range of applications software is available via the CP/M operating system, including Wordstar, Supercalc, and Calcstar.



ACT MICRO

APRICOT xi

MEMORY	256K RAM
LANGUAGE	Microsoft BASIC with MS-DOS Personal BASIC with CP/M
CASSETTE	No
DISC	Built-in 3½″ floppy plus built-in 5 or 10Mb Winchester disc
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☑ FUNCT ☑
DISPLAY	TV □ MONITOR ☑ SUPPLIED ☑
INTERFACE	PARA ☑ SERIAL ☑ BUS □
GRAPHICS	BLOCK ☑ USER ☑ LINE ☑ RES 800 by 400 COLOUR – TEXT 80 by 25 or 132 by 50
SOUND	No

DEALER

DORSET

PARKSTONE COMPUTER SYSTEMS
Main Nascom Dealer & Service Centre
Also Dealers for Gemini, Transtec, Genie
18 Station Road, Lower Parkstone,
Poole, Dorset BH14 8UB.
Tel: PARKSTONE (0202) 746555

DEVON

PLYMOUTH'S DEALER nascom
S & R BREWSTER LIMITED
86-88 Union Street, Plymouth PL1 3HG
Tel: 0752 665011 Open: 6 days

LANCASHIRE

EV COMPUTING
700 Burnage Lane,
Manchester M19.
Tel: 061-431 4866
80-BUS SOLUTIONS

SUFFOLK

M. D. W. ELECTRONICS
47 Woodbridge Rd. East, Ipswich
IP4 5QN.
Tel: (0473) 78295
Nascom, Nasbus, Gemini, Multiboard Stockists.

WALES

BYTE-RITE
Llandaff R & TV Ltd, 24-26 High St.,
Llandaff, Cardiff. Tel: 0222 563760
Nascom, Gemini, Lucas Logic, C.A.D.,
Networks, Dragon, Peripherals.

WARWICKSHIRE

BUSINESS & LEISURE MICROCOMPUTERS
16 The Square, Kenilworth,
Warwickshire CV8 1EB
Tel: 0926 512127
We are the largest Lucas Nascom dealer in the UK and carry the greatest variety of products for this range of computers.

LONDON

POSEIDON COMPUTER SERVICES LTD.
of Hampton
S.W. London dealer for Sirius I
Demonstations by appointment. Most types of printers supplied.
Bespoke Software a speciality
Competitive prices. Full UK delivery
Tel: 01-941 1147/5986 Telex: 8954665

WALES

SIGMA SYSTEMS LTD
266 North Road, Cardiff
Tel: 0222 621414
Main dealer and Service for
ACT, SIRIUS, APRICOT,
IBM, COMMODORE & DIGITAL

CBM MICRODEALER

CHESHIRE

Automated Business Equipment Ltd — Stockport — 061-432 0708
CBM specialist for industrial accounts, management graphics and planned maintenance.

HERTFORDSHIRE

Alpha Business Systems Ltd
Church Street, Industrial Area, Ware,
Herts. Tel: 0920 68926
Specialists in recommending and supplying complete systems for small businesses.

LONDON

Da Vinci Computer Store
APRICOT
SIRIUS
COMMODORE
IBM
Approved Dealer
112 Brent Street,
Hendon,
NW4 2DT
Tel: 01-202-2272

MAYFAIR MICROS
for a full range of Commodore equipment at very competitive prices. Blenheim House, Podmore Road, London SW18 1AJ.
01-629-2487

STAFFS

COMPUTER CABIN
The store with everything for the enthusiast. Official Commodore Dealer and service and information centre.
24 The Parade, Silverdale, Newcastle,
Staffs. Tel: 0782 636911.

TYNE & WEAR

KEY COMPUTER SERVICES LIMITED
Micro Computer solutions to everyday business problems.
Osborne House, 28 Osborne Road,
Newcastle upon Tyne NE2 2AJ.
Telephone: (0632) 815157
Directors: J. Sowerby, V. Shepernson.

WALES

SIGMA SYSTEMS LTD
266 North Road, CARDIFF
Tel: 621414
Also authorised dealer for CBM Digital, IBM, ACT, Sirius & Apricot.

WEST MIDLANDS

MICRO BUSINESS CENTRE LTD
Wolverhampton Computer Centre, 17-19 Lichfield Street, Wolverhampton,
West Midlands WV1 1EA. Tel: (0902)
29907 or 29021
Complete range of Commodore products always available.

YORKSHIRE

YORKSHIRE ELECTRONICS
Commodore Appointed
Commerical Systems Dealer
Full range of peripherals and Software available.
Caxton House, 17 Fountain St.,
Morley, West Yorkshire.
Tel: 0532 522181

COMMODORE 715B

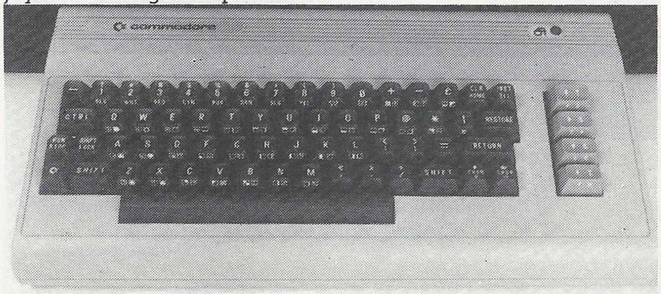
MEMORY	256K 20K ROM
LANGUAGE	Commodore BASIC
CASSETTE	None
DISC	Single or dual floppy disc drives
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☑ FUNCT ☑
DISPLAY	TV □ MONITOR SUPPLIED ☑
INTERFACE	PARA ☑ SERIAL ☑ BUS □
GRAPHICS	BLOCK ☑ USER □
	LINE □ RES 800 by 25
	COLOUR No TEXT 80 by 25
SOUND	Three Channels

Notes. The Commodore 715B is the top model in the 700 range of business machines. It is built round the 6509 processor, but there is a second processor (8088) option. The machine has been designed to meet the IEC specifications. The black-and-white monitor screen is integral and features tilt and swivel. The keyboard may be detached.

COMMODORE 64

MEMORY	64K RAM 26K ROM
LANGUAGE	PET BASIC
CASSETTE	300 baud
DISC	extra DOS
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC □ FUNCT ☑
DISPLAY	TV ☑ MONITOR SUPPLIED □
INTERFACE	PARA ☑ SERIAL ☑ BUS ☑
GRAPHICS	BLOCK ☑ USER ☑
	LINE □ RES 80 by 25
	COLOUR 16 TEXT 40 by 25
SOUND	Three channels

Notes. The Commodore 64 is a 6510 based micro that can also use Pascal, COMAL, LOGO, FORTH and PILOT. Programs can be loaded from cassette recorder or disc drives, both extra, or cartridges. The various peripherals include printer, joysticks and games paddles.

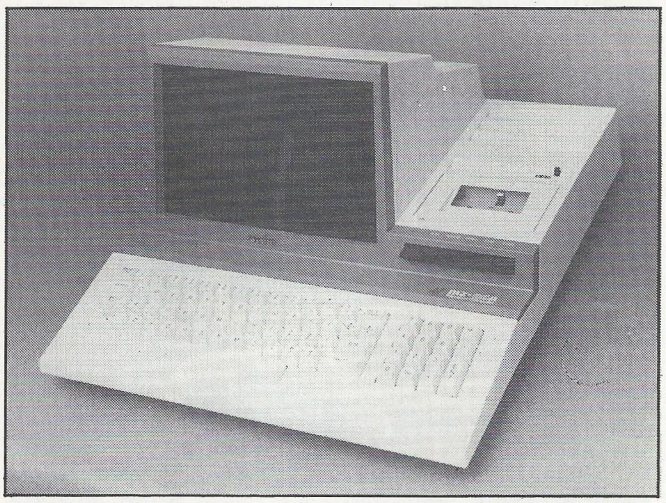


SHARP MICRODEALER

SHARP MZ-80A

MEMORY	48K RAM	4K ROM		
LANGUAGE	Microsoft BASIC			
CASSETTE	1200 baud (built-in)			
DISC	extra	DOS		
KEYBOARD	QWERTY☑	CURSOR☑	NUMERIC☑	FUNCT□
DISPLAY	TV□	MONITOR☑	SUPPLIED☑	
INTERFACE	PARA☑	SERIAL□	BUS☑	
GRAPHICS	BLOCK☑	USER□		
	LINE□	RES 80 by 50		
	COLOUR	TEXT 25 by 40		
SOUND	Single channel			

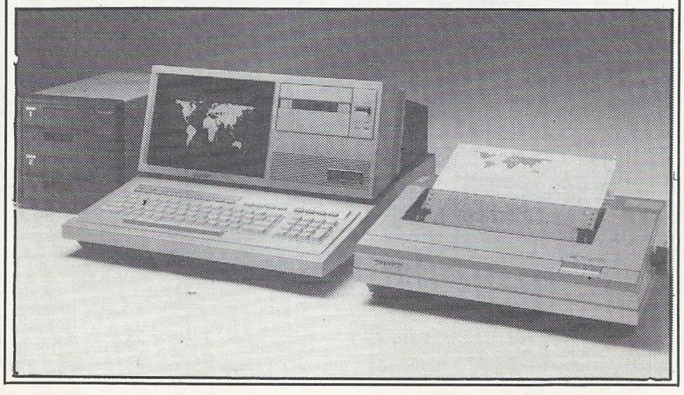
Notes: The Sharp MZ-80A is a Z80 based micro. An expansion unit, printer, floppy disc unit and other peripherals are available. Other languages can also be used such as Pascal merely by replacing the tape. With the floppy disc option the machine can respond to higher level software such as Disc BASIC and FDOS (including BASIC compiler). A small range of business and educational software is available. The supplier is **Sharp Electronics (UK) Ltd.** Thorp Road, Newton Heath, Manchester M10 9BE.



SHARP MZ-80B

MEMORY	64K RAM	2K ROM		
LANGUAGE	BASIC (on tape)			
CASSETTE	1800 baud built-in			
DISC	extra	DOS		
KEYBOARD	QWERTY☑	CURSOR☑	NUMERIC☑	FUNCT□
DISPLAY	TV□	MONITOR☑	SUPPLIED☑	
INTERFACE	PARA□	SERIAL□	BUS☑	
GRAPHICS	BLOCK☑	USER□		
	LINE☑	RES 320 by 200		
	COLOUR	TEXT 25 by 80		
SOUND	3 channels			

Notes: The Sharp MZ-80B is a Z80A based micro. Various other languages can be loaded as the machine is "soft", no language being fitted in ROM. Expansion unit, the MZ-80P5 printer and the MZ-80FB floppy disc drive are also available. The supplier is **Sharp Electronics (UK) Ltd.** Thorp Road, Newton Heath, Manchester.



BERKSHIRE

Kuma Computers Limited

Full support with software and hardware.
Sharp 700, Sharp MZ80A and MZ80B.
Unit 12, Horseshoe Pk., Horseshoe Rd.,
Pangbourne, Berkshire. RG8 7JW
Tel: (07357) 4335

COMPUTER 100
181 Oxford House, Reading,
Berks RG1 7UZ
TEL: 0734 591616
One of the UKs largest stockists

CAMBRIDGESHIRE

MS CONSULTANTS (CAMBRIDGE) LTD
Bridgefoot House,
159 High St., Huntingdon
PE18 6TF. Tel: 0480 51721
Specialists in Sharp
Hardware & Software

LANCASHIRE

STATION ELECTRICAL
Dept CT, Coastal Road,
West Bank,
Lancs LH26HN. Tel: 0524 824519
Large range of software, books and peripherals for SHARP MZ 700. 2nd hand computers bought, sold and part exchange. Also repairs.
SAE for lists

Sumita
12-14 Avenham St.
Preston (0772) 51686
Sharp MZ-3541 Based Software, Bakehouse Orders System, Barstock System.
SALES — SERVICE
SOFTWARE

LINCOLNSHIRE

SHARP CENTRE
16 Melville Street, Lincoln.
Tel: Lincoln 32379
All Sharp Microcomputers. Specialists in Industrial Process Control Systems. Customers include British Telecom, M.O.D. CRA & CTA Members

LONDON

SHARPSOFT LTD.
With the Sharp user in mind.
Sharp MZ 700 including cassette
£199.99
Sharpsoft Ltd, Crisallen House,
86-90 Paul Street, London EC2.
Tel: 01 - 729 5588.

MICROS FOR MANAGERS
149 Gloucester Rd.,
London SW7 4TH
New program for PC1251: cost estimatic and stock additive. PC5000 stock £1639. Ask us for portable and pocket computer hardware and software.
Tel: 01-370 5125
for appointment

MIDLANDS

JAXREST LTD
Linton House,
Catherine Street, Aston,
Birmingham.
Tel: 021 328 0543
Sharp Service and Support

NORFOLK

GRASSROOT COMPUTERS
25 Wellington Rd., Dereham.
TEL: 0362 4640
Specialist in small business computing. Send for details of stockbox — the complete and thorough stock system with 2,500 items per disc. Other modules to follow.

SCOTLAND

micro change ltd
170 Eastwood Mains Road, Clarkston,
Glasgow G76. Tel: 041- 554 7623
SHARP ALTOS SUPERBRAIN
Call us for your software and hardware requirements — We will tailor our data base to meet these.

LOWDATA SYSTEMS
A division of Lowdon Bros. & Co. (Engineers Ltd). PO Box 53, Blackness Road, Dundee, DD1 9JG.
Tel: (0382) 22314
Full range of Sharp Equipment stocked

WALES

STEVE'S COMPUTER COMPANY LTD
CASTLE ARCADE, CARDIFF.
Tel: 0222 — 371578
OPEN: Mon — Sat. 9 - 5.30
For Components to Computers

WEST SUSSEX

MICROCENTRE LTD.
SYSTEMS, SOFTWARE
SERVICE, SUPPORT
28-30 Station Road,
Bognor Regis, Sussex.
TEL: 0243 827779

YORKSHIRE

PROGRAM 1
For Business Software
35 Albion Street, Hull.
TEL: 0482 20022

AN ARGUS SPECIALIST PUBLICATION

electronics today INTERNATIONAL

EX42 KEYBOARD INTERFACE

As we said when we published the design for the EX42 printer interface, it seems a pity to have such a nice keyboard and not to make use of it. Well, now you can do, with this interface. Our next trick is to turn a ZX81 into a word processor... but don't hold your breath waiting for it!

It's no good, we'll have to go back – the Captain's forgotten his copy of ETI.

SIREN UNIT

After doing all the feature articles in this issue (well, nearly all), Phil Walker was itching to get some dirt under his fingernails, and here's the result! The ETI banshee's wailing will scare the burglars away — it's designed to accompany the ETI Warlock, published last month, and there will be more details of how to use the two together or independently.

ACTIVE LOUDSPEAKER

Active loudspeakers have a lot to offer, especially if you can escape the 'esoteric' price tag by building them yourself. So it's hardly surprising that we have already published one or two designs for active speakers — and it will hardly be surprising if we carry on publishing designs, there's no such thing as the definitively 'right' loudspeaker.

DIGITAL CASSETTE DECK

Do you tire of the pain of using an audio cassette player to store your computer programs? Do you long for a cheap(ish), fast reliable method of storage yet begrudge the cost of a floppy? Well, the next issue of ETI will offer you the solution in the form of a digital cassette deck. The advantages of this design over a conventional audio deck are two-fold. Firstly, the cassette deck is solenoid controlled, and these will be operated directly from the computer, by means of a special-purpose interface. Secondly, by not having the conventional audio amplifier in the way, it can be designed to have a very much higher baud rate than the average tape system. However, this is not at the expense of compatibility with proprietary software.

ALL THIS AND MORE IN THE SEPTEMBER ISSUE OF ETI, ON SALE AUGUST 3RD. PLACE YOUR ORDER NOW, EVEN IF YOU'RE GOING ON HOLIDAY.

Huh! I can't afford a holiday, so the least you lot can do is buy the mag while I'm stuck here slaving away...

COMPUTING TODAY

Lineage: 40p per word.

Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
IAN ATKINSON
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

SOFTWARE APPLICATIONS

Extender: A Machine Code Utility Program For The LII TRS80 & Video Centre comprising of:
BASYST: Converts Basic programs into System format with six letter file names. CLOSE: Closes program off prior to MERGEing. DATA COMPILER: Compiles data statements between two memory addresses. FIND & RESCUE: Reinstates Basic program, even when NEWed. HEX TO DEC: Converts 1 or 2 byte hexadecimal number to decimal equivalent. LOWCASE: Changes high case letters to lowcase. MERGE: Merges two programs together. RENUMBER: Renumbers whole or part of Basic program. ONESTROKE: Allows multiple entry of defined keywords. PUT: Puts a 1K block of memory to screen. SQUASH: Removes all unwanted spaces in program listing. RUN: Runs Basic program with single keystroke. SINGLE & PAGE: Allows single line or page scrolling. SYSTEM TAPE COPIER: Allows the copying of all system tapes. VALIDATE: Validates all line branches in listing.
£9.95 from:
OLYMPIC SOFTWARE,
15 Arnhem Close, Aldershot,
Hants, GU11 1RJ.

UTILITIES

QL UTILITIES

4 Programs on Microdrive for Sinclair QL to prevent Directory overflowing the screen provide single key loading or DELETION of files, repeat FORMATING of cartridges and back-up copying of whole or part of the cartridge. **£10** from:
W.D. SOFTWARE,
Hill Top, St. Mary, Jersey, C.I.
Tel: 0334 81392

HARDWARE

NEWBRAIN & SANYO

Professional Micro Computers for the price of hobby machines.

NEWBRAIN ON SPECIAL OFFER

With over £200.00 free software. (Accounts, databases, etc, etc) Limited Offer-ring now!

SANYO 550/555 COMPUTERS

Micropro Wordstar, Calcstar etc at no extra cost! **Printers:** Epson, KDC, Juki, Shinwa, Daisystep 2000 etc. **Sanyo Monitors & Recorders.**

Call STEVENAGE (0438) 812439 anytime for hardware/software lists **Mail Order** and **Access** facilities.

ANGELA ENTERPRISES
4 Ninnings Lane, Rabley Heath,
Welwyn, Herts AL6 9TD.

M & J SOFTWARE

fig-Forth Assembly Source Listings .... £7 each
Available for the following processors: 6502,Z80, 6809, 8080, 1802, 9900, 6800, 8086/88 and PDP11.
fig-Forth Installation Manual £5
A complete 'how to do it' guide to the implementation of FORTH from the above listings.

Cheques and POs to:
M & J SOFTWARE, 34 Grays Close, Scholar Green, Stoke-on-Trent ST7 3LU.
Tel: (0782) 517876

SOFTWARE

ASTROLOGY

Self-teaching and Accurate Calculation Programs for many micros including:
48K ORIC, I/Atmos, Sharp MZ80A/K/700, Commodore 64, 16K VIC20, BBC/Electron, Sinclair 16K, ZX81 & 48K Spectrum, Dragon 32/64, Tandy Colour, NewBrain, Colour Genie, Texas TI99a, Tandy TRS80/Genie & PET

Please send large SAE to:-
ASTROCALC
Dept (CT), 67 Peascroft Road, Hemel Hempstead, Herts HP3 8ER.
Tel: 0442 51809

COMPUTER SOFTWARE BUSINESS AND GAMES FOR MOST COMPUTERS
Commodore 64, VIC 20, BBC, Atari, Dragon, Spectrum MZ700, ZX81.

New releases for CBM 64, Spectrum, BBC, Atari, Dragon, Spectrum, MZ700, ZX81.

New releases for CBM 64, Spectrum, BBC. Every 2 weeks — just send SAE and we send you a new list every 2 weeks.

Altai C10 data cassettes 40p each. 5 C15 data cassettes £2.40 5¼" floppy discs D/sided — DL density. £2.30 each or 5 for £10.

Send to: **M. J. Seaward,**
St. Olaf's Road, Stratton,
Nr. Bude, Cornwall EX23 9AF
Tel: (0288) 4179

GAMES

COMPUTER SOFTWARE and accessories, send 2 x 16p stamps for free lists stating which computer to: Electronic Facilities, 3 High Street, Sandy, Beds SG19 1AG.

SOFTWARE EDUCATIONAL

ATARI 400/800 owners learn French or German and save money! £10 off Atari language course send only £14.95 plus £2.95 P&P. All credit cards accepted. Callers welcome open 7 days a week. Warehouse 10, Dept. PCT, 210 Tower Bridge Road, London SE1 or phone 01 407 8793.

AUCTIONS

MICRO COMPUTER AUCTIONS

Regular monthly auctions for all micro hard & software.

Send for entry form or next catalogue to:

MICRO COMPUTER AUCTIONS (CT),
59 Grays Inn Road, London WC1X 8TL

Tel: 01-242 0012 (24 hours)

ACCESSORIES

BLANK CASSETTES

Guaranteed top quality computer/audio cassettes at great budget prices.
Packed in boxes of 10 with labels, inlay cards and library case.
Prices include VAT, post and packing.

□ (C5)	£4.35	□ (C10)	£4.40
□ (C12)	£4.45	□ (C15)	£4.50
□ (C30)	£4.70	□ (C60)	£5.30
□ (C90).	£7.00		

BASF FLOPPY DISCS

Prices of boxes of 10

□ 5¼ Single side/ **Single** density	£18
□ 5¼ Double side/Double density	£19
□ 5¼ Double side/Quad density	£23

DISC DRIVES

Include Manual, Leads, Utilities Disc
□ TEAC 55A 40 tracks – £139 each
□ TEAC 55F 40/80 switchable D.S. – £209 each **FREE DELIVERY UK ONLY**
Indicate quantity of each product required in boxes
Cheque/P.O. enclosed for £ ____________

NAME ....

ADDRESS ....

....

PROFESSIONAL MAGNETICS LTD
Cassette House, 329 Hunslet Road, Leeds LS10 3YY
FREEPOST Tel. (0532) 706066 **CT**

BOOKS AND PUBLICATIONS

PARAPHYSICS Journal (Russian Translations); Psychotronics, Kirlianography, Heliphonic Music, Telekinetics, Computer Software. SAE 4 x 9". Paralab, Downton, Wiltshire.

ALARMS

BURGLAR ALARM equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. C.W.A.S. Ltd., 100 Rooley Avenue, Bradford BD6 1DR. Telephone: 0274 731532.

FERRANTI
I.Te.C.

BBC MICRO CASSETTE LEADS

(a) 7 pin din to 7 pin din
(b) 7 pin din to 2 x 3.5 mm & 1 x 2.5 mm Jacks
(c) 7 pin din to 3 pin din & 1 x 2.5 mm Jack

ONLY £2.25 ea inc p&p/VAT

Ferranti Oldham ITeC
Department 101, Orme Mill,
Crimbles Street, Waterhead,
Oldham OL4 3JA

FOR SALE

APPLE IIE Datastar package with 280 card, CP/M and many utilities £170 ono. Walters Reading 56525.

CORTEX First £120. Telephone 0900-66028.

SMALL BUSINESS System for sale Tandy c/c 32K, dual disks and printer. Wordprocessor spreadsheet ++software. £800 ono.

SERVICES

DRAGON 32 variable dump and listing of your BASIC software from cassette or DRAGON DAS. For fast service send back-up media, SAE and £1.25 per program to: Mr. A. Seymour, 13 Meller Close, Bedlington, Surrey CR0 4UB. Phone: 980 6501.

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

BERKSHIRE

Micro General
PRINTER SPECIALISTS
Call for advice on printer selection.
Epson & Microline stockists. Interfaces & cables available for all popular micros.
6 THE BIRCHWOODS, TILEHURST, READING.
TEL: 0734 25226

CHESHIRE

northern computers
Churchfield Rd., Frodsham.
Tel: (0928) 35110

Open: 6 days 9-6. Retail and Wholesale. Apple II & III, Atom, BBC, VIC20/64, Newbrain, Dragon 32, Electron, Spectrum, All accessories. Easy parking off M56.

Computer Junk Shop
We Buy, Sell, Break Computers & Peripherals.
10 Waterloo Rd, Widnes, Halton. Tel: 051 420 4590.

CORNWALL/DEVON

A. B. & C. COMPUTERS (CT)
Duchy House, 6 Lower Aylmer Sq., St. Austell.
Tel: 0726 64463/67337
Wide range of popular Micros, Printers, books and accessories. We stock all U need — try us first for service and competitive prices

HERTFORDSHIRE

NEWBRAIN & SANYO
HARDWARE & SOFTWARE
Printers, Epson, KDC, Juki etc. Monitors, Tape Recorders, Books, Expansions, CP/M. **Sanyo** 550/555 **Computers.** Access/Mail Order. Ask for details.
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

LANCASHIRE

LANCASHIRE MICROS
51 QUEEN STREET, MORECAMBE.
Tel: 411435. Also open Sundays.
Sinclair, Commodore, Acorn/BBC + Dragon.
Authorised dealer for Bug-Byte, Imagine, Quicksilva.
Artic, Melbourne House, Silversoft etc.

NSC COMPUTER SHOPS
29 Hanging Ditch, Manchester.
Tel: 061 832 2269
Apricot, Sirius, Kaypro, authorised BBC dealers. Torch, full range of printers and monitors plus software and all other computer peripherals.

LINCOLNSHIRE

SHARP CENTRE
16 Melville Street,
Lincoln.
Tel: Lincoln 32379.
Open: 9am-5.30pm closed Wed.

LONDON

A. J. Duchesne
(Computer Consultants) Limited
Specialists in Small Business Computing
10-12 Creechurch Lane, London EC3A 5AY
Telephone: 01-621 0433

COMPUTING +
264 Earls Court Road, London SW5. Tel: 01 373 4508 or **01 373 5000 Ext 240.** Suppliers of mini and micro computers for home and office use, together with supporting software. Leading manufacturers' products supplied.
Come and see the Mackintosh.

SOUTH LONDON

CROYDON COMPUTER CENTRE
Authorised Acorn Service Centre
29a Brigstock Rd., Thornton Heath, Surrey. Tel: 01 - 689 1280
BBC, Acorn, Electron, Genie, Oric, Kaga Microvitek Zenith Monitors. OKI 80, 82A + 84 Printers. Paper, Ribbons, Software etc. BUY-HIRE.

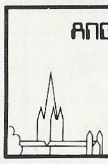
MIDDLESEX

L.B. ELECTRONICS
11 Hercies Rd, Hillingdon.
Tel: Uxbridge 55399 (24hr ans. service)
Open: 6 days, 9.30am-6pm, (lunch 1-2.15 except Sat) Surplus equipment, memory, EPROMs etc. Also established mail order service.

SCREENS MICROCOMPUTERS
6 Main Avenue, Moor Park, Northwood
Middlesex.
Tel: Northwood (09274) 20664
Official Dealers for: Acorn, Atari, Commodore, Dragon, Memotech, Oric, Sinclair, Sirius, Apricot.

NORFOLK

ANGLIA COMPUTER CENTRE



88 St Benedicts Street,
Norwich.
Tel: (0603) 29652/26002.
Open: 6 days 9am-5.30pm.

NORTHAMPTONSHIRE

NORTHAMPTON
HOME COMPUTER CENTRE
58A Wellingborough Road, Northampton.
Tel: (0604) 22539
Open: 6 days a week from 10 - 6
SPECTRUM ZX81 REPAIR SERVICE

NORTHERN IRELAND

NEWBURN Ballycorry Co. Antrim
6 Days until 8pm. Gemini, Galaxy, Nascom, Acorn Computers. Discs, monitors, printers, servicing, hire. Industrial Control. Accounts, Word Processors etc.
WHITEHEAD 78330

SCOTLAND

VICTOR MORRIS GLASGOW
TANDY TRS 80, VIC 20, VIDEO GENIE, APPLE PANASONIC, CUMANA, EPSOM ETC.
340 Argyle Street, Glasgow G2: 041 221 8958

SUSSEX

GAMER
24 Gloucester Road, Brighton.
Tel: 0273-698424.
Open: Mon-Fri 10am-5.30pm,
Sat 9am-5.30pm.

TYNE AND WEAR

HCCS ASSOCIATES
533 Durham Rd., Low Fell, Gateshead. Tel. Newcastle 821924.
Open: 6 days 9am-5.30pm (Sat 10am-5.30pm). Specialists in: Acorn, BBC, Video Genie, VIC 20.

WALES

STEVE'S COMPUTER COMPANY LTD
CASTLE ARCADE, CARDIFF
Tel: 0222 — 371578
OPEN: Mon — Sat. 9 — 5.30
FOR COMPONENTS TO COMPUTERS

SIR Computers Ltd.
91 Whitchurch Road, Cardiff.
Tel: 0222 - 21341
BBC ACORN COMPUTER

WARWICKSHIRE

OEM Rugby MicroCentre
OEM COMPUTER SYSTEMS
9-11 Regent Street, Rugby CV21 2PE,
Tel: (0788) 70522/3/4
THE FINEST COMPUTER SHOWROOMS IN THE MIDLANDS

BUSINESS & LEISURE MICROCOMPUTERS
16 The Square, Kenilworth.
Tel: Kenilworth 512127.
Open: Mon-Fri 9am-5pm. ½ day Thur (lunch 1-2). Retail and Wholesale.

YORKSHIRE

BRADFORD'S COMPUTER SHOP
at Thomas Wright (Bradford) Ltd., Thorite House, Laisterdyke.
Tel: Bradford 668890.
Open: Mon-Fri 8.45-5.30. (Sat 12am)

ADVERTISERS INDEX

COMPUTING TODAY

CLASSIFIED ADVERTISEMENT — ORDER FORM

If you have something to sell now's your chance! Don't turn the page — turn to us!

Rates of charge: 35p per word per issue (minimum of 15 words).Please state classification

and post to: **COMPUTING TODAY, CLASSIFIED DEPARTMENT**
1 GOLDEN SQUARE, LONDON W1.

					£5.25
					£7.00
					£8.75
					£10.50
					£12.25
					£14.00
					£15.75
					£17.50

Please place my advert in **COMPUTING TODAY** for ...... issues commencing as soon as possible.

I am enclosing my Cheque/Postal Order/International Money Order for: (delete as necessary) £..... (Made payable to A.S.P. Ltd)

OR Debit my Access/Barclaycard
(Delete as necessary)

□□□□□□□□□□□□□□□□□□

All classified advertisements must be paid for in advance.

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss/Ms) ....
(delete accordingly)

Address..............................

....

....

Signature.............................. **Date**..................

Daytime Tel. No. ....

Please include my business details in the next available issue of Computing Today:

Business Name: ....

Address: ....

....

....

Tel. No.: ....

Open Hrs: ....

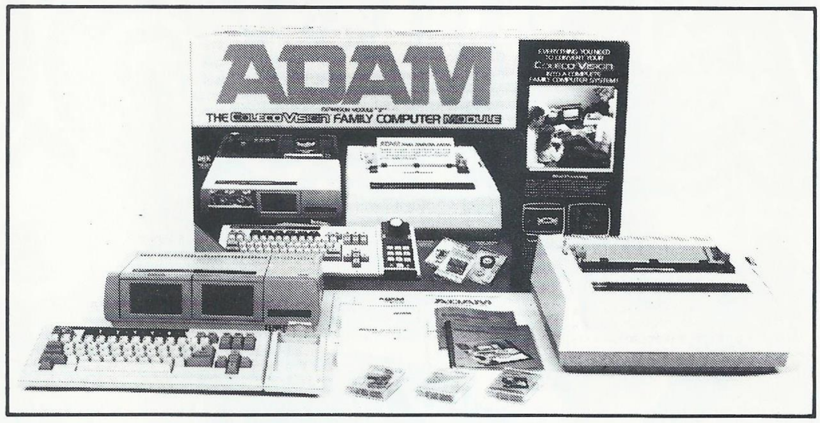
Contact (Office Use Only): ....

Post To: Computamart, Computing Today, 145 Charing Cross Rd., London WC2H 0EE.

ONLY £17.50!

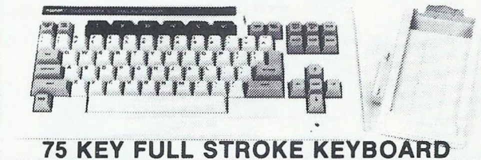
A COMPLETE PACKAGE - ALL THIS FOR £499!

- * 80K* RAM (Exp to 144K)
- * Full Stroke Keyboard
- * 256K Data Storage Unit
- * Daisywheel Printer
- * Built-in Word Processing
- * Buck Rogers Arcade Game
- * Colecovision Compatible

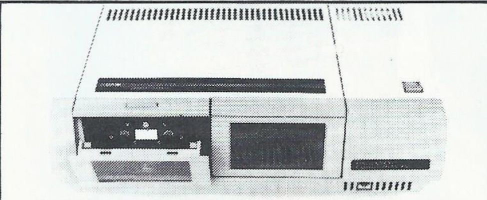

ADAM
THE COLECOVISION FAMILY COMPUTER MODULE
EVERYTHING YOU NEED TO CONVERT YOUR COLECOVISION INTO A COMPLETE FAMILY COMPUTER SYSTEM

ADAM™ - £499

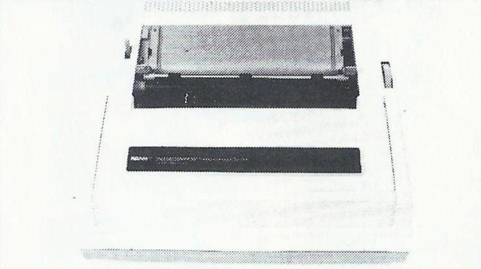
Inc VAT



75 KEY FULL STROKE KEYBOARD



MEMORY CONSOLE & DATA DRIVE



DAISYWHEEL PRINTER

COMPREHENSIVE INSTRUCTIONS

COLECOVISION GAMES CONSOLE

QUITE SIMPLY - VALUE FOR MONEY!

If you're looking for real value in a computer system, one which can handle anything from serious Word Processing to enhanced Colecovision style video games such as Buck Rogers, look no further. The Coleco Adam is here with a package which will make you wonder if you're dreaming when we tell you about it. A price breakthrough in computer systems, Adam is comprised of an 80K RAM memory console with a built-in 256K digital data drive; a professional quality, stepped and sculptured 75 key full-stroke keyboard; a letter quality daisywheel printer and a full word processing program built into the Console. Two additional pieces of software, Smart BASIC and also 'Buck Rogers - Planet of Zoom' (the ultimate in advanced video games), are included as well as a blank digital data pack. Adam can be used with any domestic colour Television set.*

MEMORY CONSOLE/DATA DRIVE: *The heart of the Adam system is the 40K ROM and 64K RAM memory console which combines with the 32K ROM and 16K RAM in Colecovision to give you a total of 72K ROM (including 24K cartridge ROM) and 80K RAM (expandable to 144K). Built into the memory console is a digital data drive which accepts Adam's digital data packs, a fast and reliable mass storage medium that is capable of storing 256K of information, that's about 250 pages of double spaced text! The console is also designed to accomodate a second optional digital data drive.

FULL STROKE KEYBOARD: The Adam keyboard has been designed as a professional quality keyboard that combines ease of use with an impressive array of features. It is stepped and sculptured for maximum efficiency and has 75 full stroke keys which include 6 colour coded Smart Keys which are redefined for each new application; 10 command keys which are dedicated to the word processing function, and 5 cursor control keys for easy positioning of the cursor at any point on the screen. You can attach a Colecovision controller to the keyboard to function as a numeric keypad for easy data entry. It can also be held like a calculator, a feature which makes working with numbers particularly easy. The joystick part of the hand controller can be used in the same way as the cursor control keys, to move the cursor around the screen.

LETTER QUALITY PRINTER: The SmartWriter letter quality daisywheel printer is a bi-directional 80 column printer which prints at a rate of 120 words per minute. It uses standard interchangeable daisywheels, so a variety of typestyles are available. The printer has a 9.5 inch wide carriage for either single sheets or continuous fan fold paper and uses standard carbon ribbons. It is comparable to many printers which cost as much as the total Adam package. The printer can be used either with the Adam's SmartWriter word processing program or as a stand alone electronic typewriter.

BUILT-IN WORD PROCESSOR: Adam comes with SmartWriter word processing built-in. This program is so easy to use that you only have to turn the power on and the word processor is on line and ready to go. Detailed instruction books are not necessary as the Computer guides you step by step, working from a series of Menu commands. It enables you to type in text, then completely edit or revise it with the touch of a few keys. Changes are readily made and a series of queries from the computer confirm your intentions, so that you can continuously double check your work as you type.

COMPATIBILITY WITH COLECOVISION: By using high speed interactive microprocessors in each of the modules, the Coleco Adam is designed to take additional advantage of both the 32K ROM and 16K RAM memory capability in the Colecovision. If you do not already own a Colecovision Console (£99 inc VAT), then you will need to purchase this when you initially purchase your Adam Computer package (£499 inc VAT), making a total purchase price of (£598 inc VAT).

WHAT IS COLECOVISION: Colecovision is one of the worlds most powerful video game systems, capable of displaying arcade quality colour graphics of incredible quality on a standard Colour TV set. The console (see picture bottom left) accepts 24K ROM cartridges such as Turbo and Zaxxon and is supplied with the popular Donkey Kong cartridge and a pair of joystick controllers. Colecovision has a range of licenced arcade hits available such as: Gorf, Carnival, Cosmic Avenger, Mouse Trap, Ladybug, Venture, Smurf, Pepper II, Space Panic, Looping, Space Fury, Mr Do, Time Pilot, Wizard of Wor and many others. So there you have it, Adam plus Colecovision the unbeatable combination. Send the coupon below for your FREE copy of our 12 page Colour brochure giving details on the complete Adam system.

SILICA SHOP LTD., 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX Tel: 01-309 1111 or 01-301 1111

ORDER NOW - OR SEND FOR A FREE COLOUR BROCHURE

To: SILICA SHOP LTD, Dept CT 0984, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX Telephone: 01-309 1111 or 01-301 1111

LITERATURE REQUEST:

□ Please send me your FREE 12 page colour brochure on Colecovision/Adam

□ I own a .... Videogame □ I own a .... Computer

Mr/Mrs/Ms: .... Initials: .... Surname: ....

Address: ....

....

....

.... Postcode: ....

ORDER REQUEST:

Please send me: □ Adam (add-on package only) .... £499 inc VAT

□ Adam & Colecovision (£499+£99) ..... £598 inc VAT

□ I enclose Cheque/P.O. payable to Silica Shop Limited

□ CREDIT CARD - Please debit my Access/Barclaycard/Visa/Am Ex/Diners Club

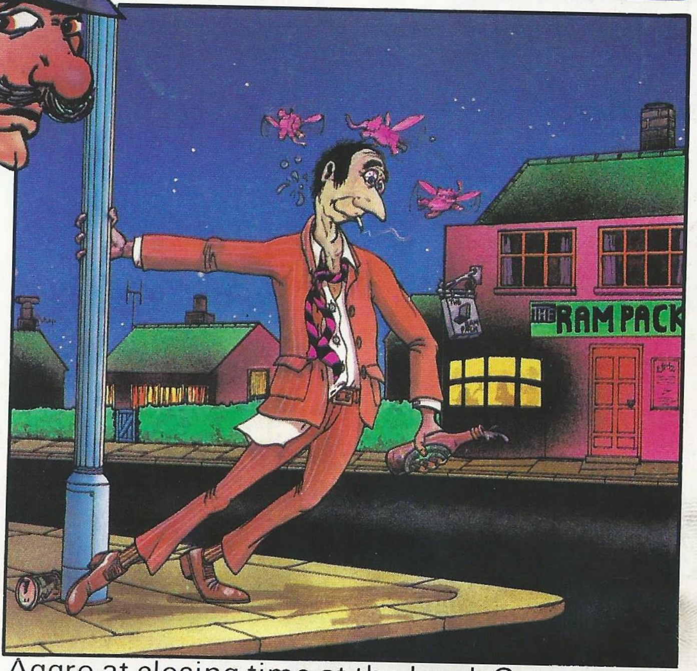
Card Number: ....

Rev 2

ANIROG

NEW TURBO 64 GAMES

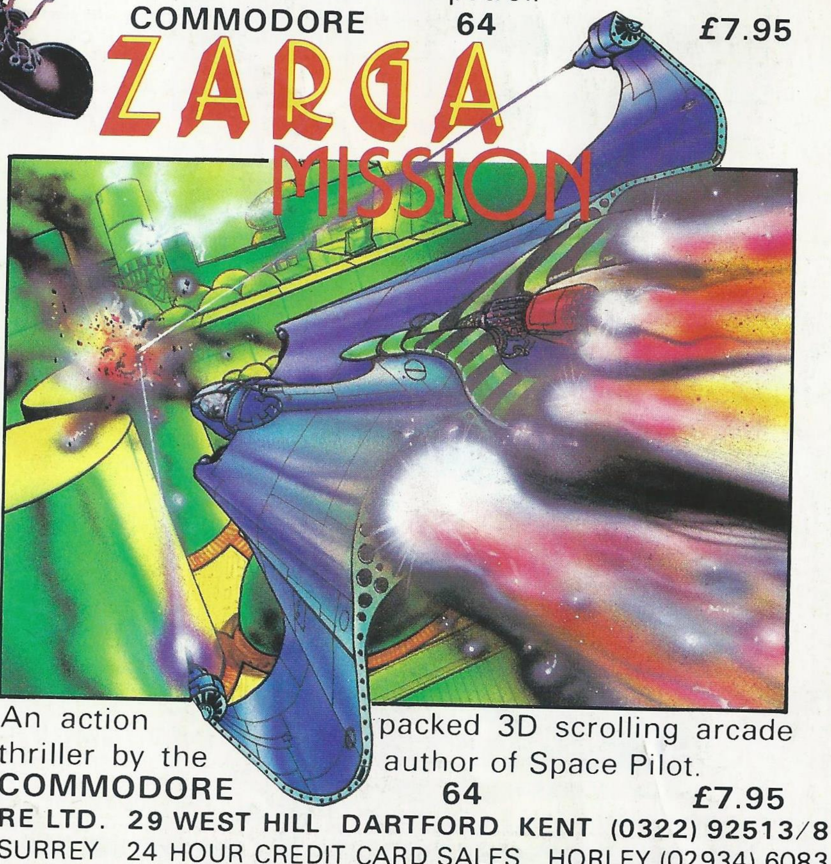
P.C. FUZZ



Aggro at closing time at the local. Getaway car screeches to a halt outside a bank. Ingenious methods employed by the Mafia to literally spirit away the loot from the High Street. Never fear - P.C. Fuzz is on patrol.

COMMODORE 64 £7.95

ZARGA MISSION



An action packed 3D scrolling arcade thriller by the author of Space Pilot.

COMMODORE 64 £7.95

Also available on Disk at £9.95

TRADE ENQUIRIES: **ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8**
MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083
PAYMENT BY CHEQUE P.O. ACCESS/VISA 50p POSTAGE & PACKAGING