An Argus Specialist Publication

Computing today

MARCH 1984
85p

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

NOW WE ARE FIVE!
Great fifth birthday issue

OUT NOW! 811 IDEAS FOR YOUR COMPUTER

NEW SAME-DAY DELIVERY
IDEAS BOOK

In the Inmac catalogue of ideas for your computer check to see what is in the latest issue for you:

Cables, cables and more cables! How to order custom-built cables. When to use screened cables. How to choose the right **EIA RS-232** extension cables for your VDU's and how to connect your **Apple, Commodore,** or other micros to **Qume, Diablo, NEC** printers. Compatible cables for connecting new peripherals to your **DEC, Data General, Hewlett Packard, IBM, Osborne** and **Tandy computers.**

Lifetime Floppies Read about Inmac Plus — the floppy with a lifetime guarantee! Choose from our range of **100% error-free** disk cartridges, **flag-free** disk packs, each with a 3-year replacement guarantee.

New Ideas! Dozens of exciting new products — many never seen in Europe before now — and lots of helpful hints on getting more from your computer department.

Service and Quality — a promise you can rely on! We promise a **next day** delivery service of our products, a **30-day risk-FREE trial**, and a **12-month** trouble-free quality replacement guarantee.

Send today for your **FREE** catalogue with no obligation to purchase — but be warned: you will want to buy once you have read the Inmac catalogue ideas book!

FREE! IF YOU USE A COMPUTER ☎ 09285-67551/01 740 9450

Send this coupon to
Dept. 628, Inmac (UK) Ltd., Davy Road,
Astmoor, Runcorn, Cheshire WA7 1QF.

Name_____

Company_____

Title_____

Address_____

_____

Postcode_____ Tel._____

Computer System_____

24 hour tel. no.
☎ 09285-67551

CONTENTS

VOL 6 NO 1 MARCH 1984

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

Acting Editor:
Peter Green

Editorial Assistant:
Jamie Clary

Senior Advertisement Manager:
Malcolm Wynd

Advertisement Manager:
Hazel Falconer

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, PO Box 35, Wolsey House, Wolsey Road, Hemel Hempstead, Herts HP2 4SS.

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.

CONSUMER NEWS

LPRINT UPDATE

The ZX Lprint Centronics printer interface for Spectrum computers from Euroelectronics (reviewed in our November 83 issue) is now being supplied compatible with Sinclair Interface 1 and Microdrives. Existing users of ZX Lprint Mk I and Mk II are being offered a conversion of their interfaces by Euroelectronics to ensure compatibility. Such conversion consists of insertion of a miniature switch in one of the connections and can be carried out by Euroelectronics for £3.25. Alternatively instructions and the required part can be supplied at £1.90. Both above prices include VAT and postage. Euroelectronics are at 26 Clarence Square, Cheltenham, Glos GS50 2JP (telephone 0242 582009).

ONE MILLION SPECTRUMS

On December 9th Sir Clive Sinclair established another landmark when the one millionth Spectrum left the production lines at Times/Dundee. With the success of the ZX81, Sir Clive has now produced two million-selling computers, which can't be bad going. Of course, during such a large production run you are bound to get some mutants, and the photo shows Sir Clive being presented with a rare albino Spectrum. We have no information on whether cross-breeding is to be attempted.

DRAWING WITH STYLUS

Having already produced a great deal of business and games software for the Commodore 64, Audiogenic have turned their attention to the peripherals market with the launch of Koala Painter — a graphics tablet for the Commodore 64. It allows 64 users to produce full colour drawings and illustrations directly on screen, limited only by the levels of their imagination. The system is extremely easy to use and allows users to take full advantages of the graphics and colour capabilities of the Commodore 64 without any need for programming knowledge.

The Audiogenic package consists of a graphics tablet with separate stylus, disc-based software, and an instruction manual. Connection to the 64 is simplicity itself, yet the package enables complete novices to produce high quality full colour illustrations far superior to the graphics included in some commercial games software.

The Koalapad, which interfaces directly into the 64, is relatively small (8" x 6" x 1") and lightweight (1 lb) so that it can be held naturally in the hand. the active pad surface is a 4" square that is slightly recessed with two push buttons located above it. The unit is contoured in such a way that it can be grasped easily in the left hand with the left thumb pushing one of the buttons, while the right forefinger or stylus does the drawing.

The Koala Painter menu, displayed on the screen, is divided into three sections: Commands, Brushes and Colour Palette. The user is able to use the menu to build up his picture, combining freehand drawings with the basic shapes included in the menu and with previously stored designs, before deciding on colouring. A brush option allows the user to vary the thickness of lines.

The Command section of the menu gives the user total control over the formation of his picture. Freehand drawings can be supplemented by integral facilities for Lines, Frames, Boxes, Rays, Circles and Discs. One command in particular, allows for mistakes to be rectified by 'undoing' the previous command — ideal for errors in colouring. Additional commands allow the selection of colour, copying from one area of the picture to another, copying elements from a previous picture to the current picture, creating mirror images, swapping between images and zooming to allow for detailed drawing or to correct small errors.

A choice of eight brush sizes allows the user to vary the style of the illustration by increasing or decreasing line widths, while the colour palette provides the choice of 16 colours and 16 patterns. With such facilities, extremely complex

colour illustrations, even suitable as a background for games programs, can be prepared.

Available from Audiogenic and their world-wide network of dealers and distributors, Koala Painter costs £89.95 including VAT. For further information, please contact Audiogenic Ltd, PO Box 88, Reading, Berks RG1 2SN.

GENIE G-MON CORRECTION

Owners of the 16K Colour Genie who are having trouble with the G-MON program published last month can get things going with a simple number change. Apparently the listing published tries to put the stack in an area of memory that, well, doesn't contain any memory in the 16K model! Look at line 59 and find the sequence of data that goes 49,0,192. Change this to 49,0,128 and you will find that the program will now work perfectly in the 16K model. The checksum mentioned in the text should now be 86261.

ELECTRON INTERFACES

MRL have added to their existing range of interfaces, a new series aimed at the Acorn Electron. The series has been dubbed the 'Electron Cloud' (groan) and includes a parallel printer interface, an analogue-to-digital converter with a joystick port, and a user port.

The units can be bought individually for £39.95 or as a set in one box for £79.95. The individual units can also be expanded internally by plug-in cards costing from £29.95 (cables, VAT and carriage are all included). Contact Micro Research at Industrial Unit 6, Knightsridge East, Livingston, West Lothian (telephone 0506 31605). MRL also supply interfaces for the Spectrum, Atari and Commodore machines.

THAT SIRTON SOMETHING

Sirton Computer Systems, manufacturers of the Midas range of computers, have launched a new Track-Ball Graphics Cursor unit. Designed for use with all hardware fitted with a standard eight-bit parallel input, the Sirton Track-Ball Cursor Unit can be easily and rapidly 'programmed in' using either BASIC or machine code. Software is supplied for use with the Pluto colour graphics board manufactured by IO Research.

Track-Ball cursors, already standard equipment in many military and scientific systems, are set to become the graphics manual interface of the future, with many advantages over the conventional 'joystick'. Chief among these is the fact that the track-ball relies upon optical rather than electronic precision in operation, providing the user with a 'vernier' effect that makes large- and small-scale graphics manipulation equally viable.

The Sirton Track-Ball Cursor Unit is ideal for such tasks as symbol collection, architectural planning and graph plotting. It is also highly efficient in three-dimensional graphics processes such as 'zoom' routines, when any part of a given image can be easily and rapidly centred between successive 'enlargements'. The unit is also fitted with eight function keys for command entries.

The Sirton Track-Ball Cursor Unit is priced at £325.00 and will appeal strongly to the scientific and technical installation with a need to simplify and standardise its graphic interfacing. For more information contact Sirton Computers Ltd, Unit 14, 29 Willow Lane, Mitcham, Surrey (telephone 01-640 6931).

THERE'S A SWITCH

With a new range of switching boxes introduced by Willis Computer Supplies, it is possible to switch instantly between computers, printers and VDU terminals, saving reconnection time and hundreds of pounds on extra peripherals that are usually left idle for most of the day. The Willis Switching Boxes all have female sockets and can be supplied to suit RS232 connections, Centronics connections (suitable for most micros) and IEEE connections. They are guaranteed for 12 months. For further information contact Willis Computer Supplies Ltd, PO Box 10, South Mill Road, Bishop's Stortford, Herts.

VERO INTERESTING

The 1984 edition of the Hobby Herald containing over 100 new products is now available. The greatly increased range now includes connectors for all applications including micros, telephone connectors, etching kits, new enclosures and many other items. Send 50p for your copy to BICC-Vero Electronics Ltd, Retail Department, Industrial Estate, Chandlers Ford, Hants SO5 3ZR (telephone 04215 62829: telex 47551). Please state if you would like the name and address of your local Vero stockist.

VISION ON

Seescan Devices of Cambridge are adding to their range of professional image stores by introducing a unit specifically designed to operate with the BBC microcomputer. The new unit combines a high quality vidicon camera and a software controlled Z80A-based intelligent framestore which is interfaced to the BBC via the 1 MHz bus giving second processor capability.

The unit allows television pictures to be 'captured' in real time, and downloaded into the main memory of the BBC, where supplied or user generated routines may be used to process the video data. After processing, the digital image may be copied back to the Seescan Devices unit where it is reformatted into a standard video signal. The digital camera provides a resolution of 128

by 128 image elements with 16 shades of grey. BBC colour graphics may be generated directly from the digitised image and may be downloaded to a dot matrix printer for hard copy if desired.

Principal applications for the digital camera include robot vision, games graphics, process control, scene analysis and pattern recognition. Together with the BBC this unit provides a powerful tool for many tasks requiring image analysis.

The digital camera costs £465 including VAT from Seescan Devices, 25 Gwydir Street, Cambridge CB1 2LG (Telephone: 0223 314553/61376 (24 hour answer service): telex: 81406).

INTERBEEB

Following the tremendous success of DCP's range of computer interfaces for the Sinclair ZX81 and ZX Spectrum computers, the company has now launched a set of products for the BBC Microcomputer system Model B. The main pack in the set is called Interbeeb and contains a complete electrical interfacing system in a neat compact case. Specifically the unit provides four relay outputs, four switch inputs, an eight-bit input port, eight-bit output port and an eight-channel analogue-to-digital converter. Additional packs or circuitry can be added to the rear of the unit on the DCP BUS connector — a standard electrical interface available on all DCP Interfaces so that extra packs are interchangeable between computer systems. The main pack comes complete with computer connecting cable and a specially designed low voltage mains power supply which is also supplied with the necessary cables.

Typical applications for the product include control experiments, heating systems, burglar alarms, model control, industrial monitoring and so on. It is likely that the pack will be used in schools and industry as well as home projects and the system is very easy to control using simple software which is fully explained in the manual provided.

The unit is available from a range of dealers and distributors at a recommended selling price of £59.95 including VAT.

Suitable for use with all CDP Interfaces with a DCP BUS facility (eg Interspec and Interbeeb) the new AD and DAC Packs provide fast Analogue-to-Digital conversion and Digital-to-Analogue Conversion in compact cases designed to match the rest of the DCP range. They feature precision voltage references and standard sockets for connections as well as parallel DCP BUS connectors for even further expansion. The packs sell for £19.95 each including VAT. Mail order sales are being handled by RH Electronics (Sales) ltd, Chesterton Mill, French's Road, Cambridge CB4 3NP (telephone 0224 311290).

BASICARE FOR THE 64

The Basicare Modular Expansion System is designed to increase the 'real world' computing capabilities of the popular Commodore 64. It has previously been available for the ZX81 and Spectrum. The system consists of a base plug-in expansion device, the C64, Persona, and a complete series of user-selectable modules for real world control, and memory expansion with one megabyte capacity.

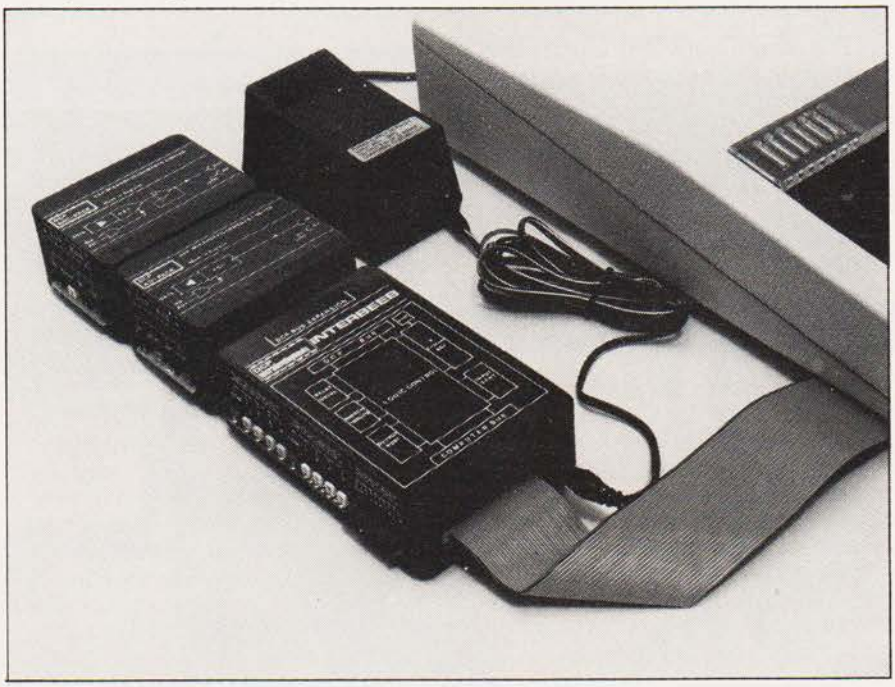
The C64 Persona plugs into the expansion port. It has four ROM cartridge slots, soft switched, to provide immediate access to user software. The C64 Persona also provides 8K of additional software in ROM with several important added features and functions. These include extensions to Commodore BASIC (APPEND, OLD and others); a complete machine code assembler; a simple, comprehensive controller for the sophisticated sound chip in the Commodore 64; and easier use of the hi-resolution graphics capabilities.

As a stand-alone expansion device, the C64 Persona will appeal to serious-minded Commodore owners. It can also be used as the heart of a comprehensive series of expansion modules built by Basicare that can provide virtually unlimited growth potential for the Commodore 64. These include memory expansion beyond the normal 38K RAM, up to a theoretical one megabyte limit. Memory mapping can be done via the Minimap module which can address blocks of memory in 8K sections. Memory expansion modules are available in increments of 16K (RAM 16) and 64K (RAM 64). The Pericon modules (PERIpheral CONtrollers) provide 'real world' interfacing. The newest module, Pericon d, incorporates relays that can be used to control motors. These capabilities were vividly demonstrated at the recent World of Commodore Show in Toronto where a Basicare system was used to control a robot made from standard Lego toy parts and motors. The same robot, controlled by a Basicare expansion system in conjunction with a Spectrum computer, recently appeared in Central Television's 'Magic Micro Mission'.

Recent additions to the range include the Link a (an analogue-digital converter) and a Real Time Clock module. Soon to be released is a Z80 card complete with on-board 64K RAM and double density disc controller card.

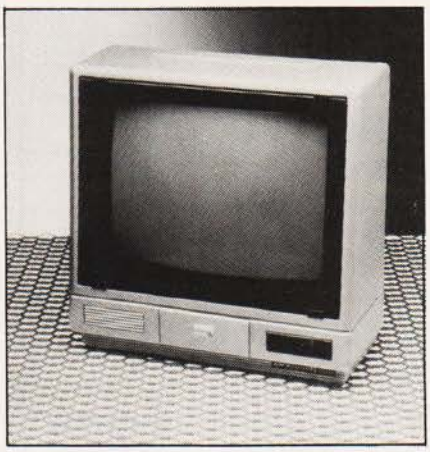
Basicare modules connect through a unique organic bus system provided by the C64 Persona. Modules can be added without limitation to provide the memory expansion and control functions required. Persona devices are available for other computers. This enables Basicare modules to 'migrate' to other computers, a feature unique to Basicare. Currently Personas are also available for the ZX81 and the Spectrum.

Basicare Microsystems can be contacted at 12 Rickett St, London, SW6 1RU (telephone 01-385 2135).

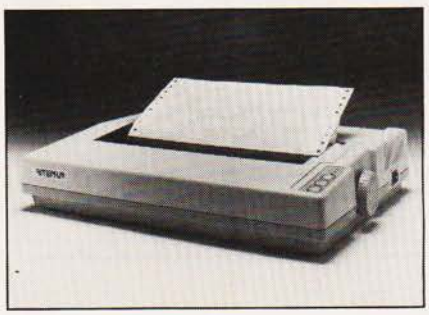


MPL MULTITUDES

Micro Peripherals Ltd have launched several new products: a colour monitor and three printers. The CM14, the UK's first all-in-one British-built colour monitor, has a 14" superbright screen, with RGB, RGBY, PAL composite and audio signal inputs. The audio output is 2 W. This range of input facilities makes the monitor ideal for use with a wide range of micros including the Apple, IBM, Sirius, BBC, Dragon, Oric-1, Lynx and Electron, as well as the home video recorder. The CM14 costs £199.



The MCP-40 is a four-colour printer/plotter with a parallel interface which can be used for charts, graphs and four-colour printing at up to 80 characters per line. It can be used with the BBC, Dragon, Oric-1, Apple and other computers, and should it look familiar, it's because it's the same printing mechanism as used in the Sharp, Tandy and Oric printer/plotters. The price is £113 excluding VAT.



The Riteman A1 matrix printer is a slimline 80-column printer which weighs only 5 kgs. It features a speed of 120 cps, hi-res graphics, italics, and standard control code capability, and costs £299 excluding VAT.

Finally the Star Gemini and Star Delta are two matrix printers aimed at the business and educational markets. The Gemini is 80 column or 136 column depending on model number, and costs £299 and £399 respectively, excluding VAT. Print speed is 120 cps. The Delta is priced at £399 excluding VAT and provides 160 cps printing with a 240 cps white space speed. An 8K buffer and parallel and serial interfaces are fitted as standard. Both types of printer have friction, tractor and roll feed, and ultra-high resolution graphics and downline loadable characters.

Micro Peripherals Ltd are at 69 The Street, Basing, Basingstoke, Hants RG24 0BY (telephone 0256 3232: telex 859669).

HANDY HP COMPUTERS

Clock and calendar functions, text-file editing and extended memory are among the built-in features of the HP-41CX, a new handheld computer announced by Hewlett-Packard. A built-in time module enables the HP-41CX owner to use the calculator as a time-based system controller, an alarm clock, an appointment reminder, a calendar, a timer or an advanced stopwatch. In addition to the standard time module functions, the HP-41CX has five new time commands designed to improve alarm capabilities and time operations.

Other features of the HP-41CX include an RPN (Reverse Polish Notation) operating system, in 24K byte of ROM, that allows users to see intermediate results and recover from errors easily. The HP-41CX also has Continuous Memory (CMOS) and an alphanumeric liquid-crystal display. The alphanumeric keyboard is redefinable, so users can assign their most frequently used programs or functions to any key for quick access and execution.

As with all the Series 40 machines, the HP-41CX system can be expanded to include a wide variety of peripherals via the Hewlett-Packard Interface Loop (HP-IL), a bit-serial interface designed for low-priced, battery-operable systems. HP-IL peripherals include printers and plotters and several low-cost instruments that can be controlled by the 40 Series.

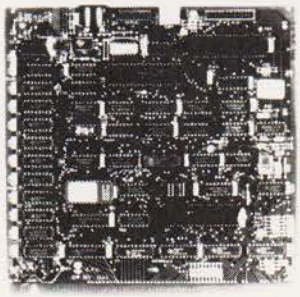
The HP-41CX can be further enhanced with plug-in software modules and software-solution books from Hewlett-Packard. More than 2,500 programs in a range of application areas are available. HP-41CX owners can write their own programs and store them on mini-cassettes or magnetic cards. Programs can also be recorded in bar code produced on an HP Graphics Plotter.

The HP-41CX is available from HP Appointed Series 40 Dealers, at a recommended retail price of £229.71. (Recommended prices of the HP-41C and HP-41CV remain at £144.55 and £203.86 respectively.) All prices quoted are exclusive of VAT.

DAISY, DAISY

C/WP Computers has added the new Silver Reed EXP 500 daisywheel to its range of printers. Printing bidirectionally at 14 cps, the EXP 500 supports 10, 12 and 15 pitch daisywheels and is supplied with a parallel Centronics interface as standard. An optional serial RS232 interface is available for a further £50 including VAT. Diablo 1610 emulation allows compatibility with WordStar and other word processing programs. The EXP 500 is attractively designed, quiet in operation and will accept a forms tractor unit. C/WP is offering the EXP 500 at £299 including VAT with a 12 month guarantee: they can be contacted at C/WP Computers, Willow House, Willow Place, London SW1P 1JH (telephone 01-828 9000).

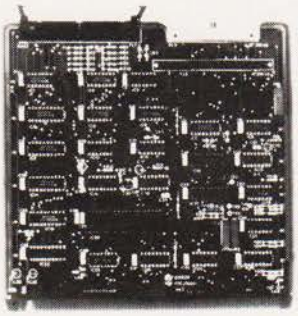
MicroValue 80-BUS MULTIBOARDS



GM813—CPU/64K RAM Board

* 4MHz Z80A C.P.U.
* 64K Dynamic RAM
* RS232 Interface
* Two 8-Bit I/O Ports
* Cassette Interface
* Extended & Page Addressing Modes
* CP/M Compatible Monitor

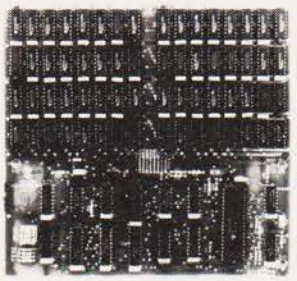
£225



GM829—Disk Controller Board

* Up To 4 Mixed 5.25 & 8" Drives
* SASI Hard Disk Interface
* Single & Double Density Operation
* Single & Double Sided Drive Support
* Supports 48 and 96 TPI Drives

£145

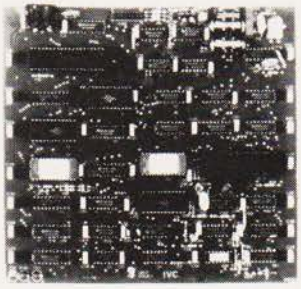


GM833—RAM-DISK Board

* Virtual Disk Operation
* 512K Dynamic RAM
* Port Mapped For Easy Interface Software
* Over 10 Times Faster Than a Floppy Disk

£450

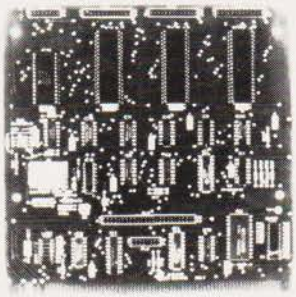
Please note: This board cannot be used as a conventional RAM board



GM812—Video Controller Board

* 80 Characters x 25 Line Display Format
* On-board Z80A Microprocessor
* Buffered Keyboard Input
* Programmable Character Generator
* 160x75 Pixel Graphics
* Light Pen Input

£125



GM816—Multi I/O Board

* 6 I/O Ports
* 4 Counter/Timer Channels
* On-Board Real Time Clock
* Battery Backup
* Further Expansion Capability

£125

PLUTO—Colour Graphics Board

* On-Board 16 Bit Microprocessor
* 640x576 Bit Mapped Display
* 192K Of Dual Ported RAM
* Comprehensive On-Board Software

£499

CLIMAX—Colour Graphics Board

* 256x256 Pixel Display
* 16 Colours
* Ultra-fast Vector & Character Generation
* Light Pen Input
* UHF or RGB Outputs

UHF Version **£199**

UHF & RGB Version **£220**

GM811—CPU Board

* 4MHz Z80A CPU
* 4 'Bytewide' Memory Sockets
* 2x8-Bit Input/Output Ports
* 8 Bit Input Port
* RS232 Serial Interface
* Cassette Recorder Interface

£125

GM803—EPROM/ROM Board

* Up to 40K of Firmware
* 2708 or 2716 EPROMS
* Page Mode Operations

£65

GM802—64K RAM Board

* 64K Dynamic RAM
* 4MHz Operation
* RAM Disable Function
* Page Mode Operation

£125

MP826—Static RAM Board

* 32K Static RAM
* Battery Backup
* Page Mode Operation

£225

EV814—IEEE488 (GPIB) Controller

* Cost Effective Controller
* Comprehensive Software Supplied
* Full Implementation
* Easy To Use

£140

GM827—87 Key Keyboard

* User Definable Function Keys
* Numeric Keypad
* Cursor Control Keys

£85

GM839—Prototyping Board

* Fibreglass P.C.B.
* 80-BUS Signal Identification
* High Density IC Capability

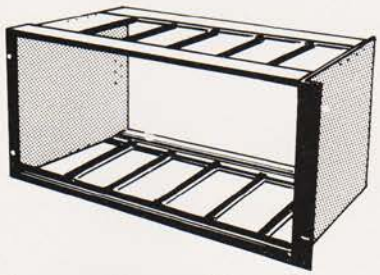
£12.50

MULTIBOARD COMPUTERS

Gemini Microcomputers

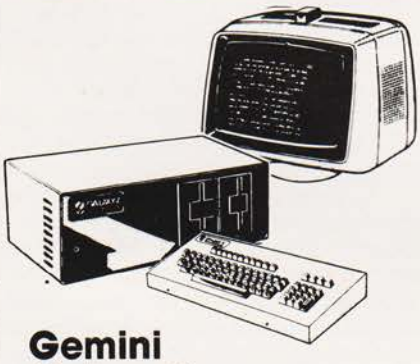
All the boards and components in the 80-BUS range are fully compatible and offer a very flexible and cost effective solution to your computer needs. For further information about the 80-BUS range contact your nearest MICROVALUE dealer.

MicroValue - MicroValue



Power Supplies, Mother Boards & Frames

* GM807 3A Power Supply	£40.00
* GM817 6A Switch Mode P.S.U.	£75.00
* GM843 10A Switch Mode P.S.U.	£95.00
* GM656 3 Slot Motherboard	£5.00
* GM654 5 Slot Motherboard	£6.00
* GM655 8 Slot Motherboard	£10.00
* MP840 14 Slot Motherboard	£47.00
* GM662 5 Board Frame	£50.00
* GM610 19" Frame	£37.50



Gemini Galaxy 2

"I would place the Galaxy at the top of my list" (Computing Today, April 1983)

* Twin Z80A Processors
* CP/M 2.2 Operating System
* 80x25 Video Display
* 64K Dynamic Ram
* Light Pen Interface
* Up to 1.6Mhz Disk Capacity
* Serial RS232 Interface
* Parallel Interface
* Numeric Keypad
* Definable Function Keys
* Cassette Interface
* 12" Monitor Included

from £1495

Quantum System 2000

Computerise Without Compromise

* 80-BUS Construction
* Serial & Parallel Interface
* Stylish Design
* Up To 2.4Mhz Disk Capacity
* Up To Three 5.25" Drives
* Fully Expandable
* Twin Z80A Processors
* CP/M Operating System
* 64K Dynamic Ram
* Definable Function Keys

Two-Drive Quantum £1910

Gemini Multinet

The Gemini Multinet enables as many people as possible to have access to their own microcomputer with mass storage and printer facilities for the lowest possible cost. This is achieved by providing a central 'fileserver' fitted with a Winchester hard disk unit and printer interfaces, in conjunction with a method of interconnecting up to thirty-one workstations to the fileserver. The fileserver and each station are fitted with the Gemini GM836 network interface board. A Micropolis 800K floppy disk drive is incorporated in the fileserver providing backup for the hard disk.

GM910 Galaxy 4 Multinet 5.4 M/byte fileserver	£2600
GM912 Galaxy 4 Multinet 10.8 M/byte fileserver	£2850
GM909 Galaxy 4 Multinet workstation	£650

Both fileservers and workstations are supplied complete with VDU's; the operating software is suppliedwith the fileserver.

Phoenix P12 Monitor

A high quality 12" data display monitor, ideal for Gemini systems. The P12 is available in both green and amber phosphor versions and has a resolution of 20Mhz.

£95

BUY FROM THE COMPUTER PROFESSIONALS

MICROVALUE DEALERS:

AMERSHAM, BUCKS
Amersham Computer Centre,
18 Woodside Road,
Tel: (02403) 22307

BRISTOL
Target Electronics Ltd., 16 Cherry Lane.
Tel: (0272) 421196

EGHAM, SURREY
Electrovalue Ltd.,
28 St. Judes Road, Englefield Green.
Tel: (07843) 3603

LEEDS
Leeds Computer Centre,
55 Wade Lane, Merrion Centre.
Tel: (0532) 458877

LONDON W2
Henry's Radio, 404 Edgware Road.
Tel: 01-402 6822

LONDON SW11
OFF Records,
Computer House, 58 Battersea Rise,
Clapham Junction.
Tel: 01-223 7730

MANCHESTER M19
EV Computing, 700 Burnage Lane.
Tel: 061-431 4866

NOTTINGHAM
Computerama, (Skytronics Ltd.)
357 Derby Road.
Tel: (0602) 781742

Telephone orders welcome

All prices are exclusive of VAT

MicroValue

REAL value – from the Professionals

SOFT WARES

SOFTWARE PRICE DECREASE

Ever since US game software has been imported into the UK, one factor has prevented their wide acceptance in the British market. Simply, that factor is price. While the high graphic and imaginative quality of these games has never been questioned, a number of factors have created an unreasonably high retail price level.

CentreSoft have changed this situation by cutting away the unnecessary frills and gimmicks — and concentrating on providing the best quality games at realistic prices. In conjunction with major US publishing houses — including Datasoft, one of the world's largest games companies — CentreSoft have re-packaged and re-priced software for the British market.

Of the first six titles which were released on 1st January through Boots and other CentreSoft dealers, the most exciting is undoubtedly Zaxxon. Probably the hottest and largest selling game in the world today, Zaxxon will now be available for only £14.95. That's about half of previous retail price levels — giving thousands more people the chance to enjoy these top US games.

ON COURSE

The Centre for Extension Studies at Loughborough University is offering in conjunction with Dean Associates an introductory course, "Using Computers in Occupational Training" on 6-7 March 1984.

Current developments in technology create an urgent need to re-train established staff in an area of rapid change. Computer based training (CBT) provides an invaluable aid to trainers: it provides trainees with individual instruction, is infinitely patient and offers motivation for rehearsal of what might otherwise be tedious repetition. CBT also enables the Training Manager to analyse progress and monitor closely each trainee.

This new course provides the essential framework to allow the trainer to exploit these new techniques with confidence. Apply for further details to Lynne Atkinson, Centre for Extension studies, University of Technology, Loughborough, Leics, LE11 3TU (telephone 0509 263171 ext. 213).

PHIPPS FOR EPSON

Phipps Associates, a software publisher of long standing, have announced a new area of activity: they have been appointed dealers for the Epson QX10 and HX20 computers and have also released a nominal ledger package and cash register software for the HX20. Also announced is the conversion of the Microshop retail stock control package for the QX10. A move to larger premises during the summer has provided the opportunity for this expansion into business systems. The new office has been equipped with a local area network for communications. This is made possible by the Clearway network controllers, for which Phipps have been appointed distributors. This links together the QX10, HX20, Spectrum and many other types of micros.

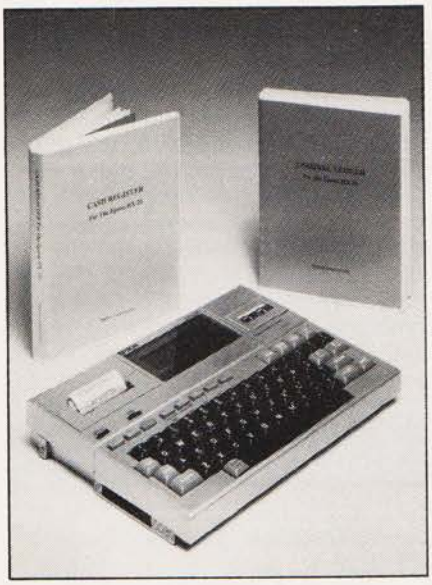
The HX20 nominal ledger package will be useful for travelling auditors, small businesses and anybody who has to struggle with a multi-column expense sheet every month. It features analysis of accounting data over 100 or more headings which have names and numbers chosen by the user. Posting facilities are provided for debits, credits and adjustments. The Audit Trail is maintained on the printer built into the HX20. The ledger is automatically self-balancing, and contra-entries, for example to cash at bank or to debtors, are provided automatically. Trial balance listing shows current month and month to date. Retail price of cassette and instruction book is £26.

The cash register software turns the HX20 into a POS terminal. Used in conjunction with the Epson bar code reader package, it can be used to capture EAN/UPC coded label data direct from packages. It features entry of stock movement data together with sales, credits, discounts, repeat registration as well as change calculation. Perhaps its major use will be for van salesmen, who need a portable invoicing system for collecting money from their customers. An inexpensive cash drawer can be linked to the HX20 and driven through the remote motor on/off socket of the HC20. Retail price of cassette and instructions is again £26.

The cash register software has been designed to run in conjunction with the Microshop package for retailers' stock control which has been converted to run on the Epson QX10. This features sales and contribution analysis by either of two reporting codes (for a clothes shop, these might be colour and style of garment), stock lists, stock valuations, and re-order control using re-order levels and re-order quantities. Included in the package is a general purpose sort program, index file management and screen control utility programs. These last three are accessible to the user allowing him to write his own programs, for use in conjunction with Microshop. The whole package retails for £320.

A major problem in software publishing is the extremely high degree of accuracy required in the product: most book publishes recoil in horror when told that a misplaced comma can ruin a book. The Phipps approach to this problem is to make sure wherever possible that there is no human hand interposed between the program running on the machine and the printed text. They have developed interfaces for a variety of micros to enter the program text direct into the printed copy by transmitting it from the micro on which the tested version was developed. The hardware design for the RS232 interface for the Spectrum was circulated free of charge to those who asked for it — a surprising number did so. The Clearway networking system installed in Phipps' new offices allows communication and transfer of files between a number of different computers. Programs developed on the QX10 can be transferred to the Sinclair Spectrum and vice versa.

The latest development in this line is the completion of a BASIC compiler for the Sinclair ZX Spectrum which allows a disc based machine,



with Wordstar as text editor, to be used for program development. The text of the program is then compiled and down-line loaded into the micro for testing. This software is not being made available to the market place as yet, but is being circulated to authors contracted to Phipps for use in games development.

For more information please contact Phipps Associates, 172 Kingston Road, Ewell, Surrey KT19 0SD (telephone 01-393 0283).

PRO PASCAL

Prospero Software have released a new Pascal compiler for 16-bit micros running the MS-DOS and PC-DOS operating systems. This version of Pro Pascal is a companion product to the CP/M-86 version announced in September. Both compilers were shown in public for the first time at Compec 83.

The 16-bit compilers are fully compatible with the original eight-bit Pro Pascal for CP/M. For a great many programs, no changes at all (apart from re-compilation) will be found necessary when upgrading to a 16-bit environment. According to Mike Oakes, a Director of Prospero: "The Z80 Pro Pascal compiler has been completely rewritten for the 8086. We find the object programs are in general more compact than on the Z80. The compilers themselves can process very large programs in as little as 66K of user RAM, making them particularly attractive to people with 128K machines."

The compiler package includes a link-editor, library manager, Pascal cross-reference generator, and comes with versions of the run-time library for use with or without an 8087 arithmetic co-processor. The cost of a single-user licence is £320. For further information contact Prospero Software Ltd, 37 Gwendolen Avenue, London SW15 6EP (telephone 01-785 6848).

A GEM FOR THE BEEB

Character Define and Envelope Editor are two new programs from Gem Software released as their Util-1 cassette for the BBC Micro using OS 1.0 and above. Util-1 retails for £9.95 (including VAT) from selected BBC-B dealers or direct from Gem Software.

Character Define provides a comprehensive and simple way to quickly create all manner of user-defined characters. Character Define can be used to create single or multiple character shapes and symbols for use in your programs. For the adventurous programmer, Character Define unlocks the door to a host of exotic screen characters — space-ships, alien creatures, alternative character sets and special-purpose symbols such as those frequently used in electronics, music and the like.

Envelope Editor helps the BBC-B user to unravel complexities of the Sound and Envelope commands, so that you can produce exactly the sounds required. Envelope Editor's graphic displays help the user in understanding the effects created by the modification of the BBC-B's Pitch and Amplitude envelopes.

Util-1 is supplied complete with an easy-to-follow comprehensive 20-page programmmers reference booklet and two function key definition strips. Both programs are written in BASIC, with Character Define employing 24K of memory, and Envelope Editor using 26K. More information can be obtained from Gem Software (Bishop's Stortford) Ltd, Unit D, The Maltings, Station Road, Sawbridgeworth, Hertfordshire.

GAME FOR A PRIZE

The success of any game — whether traditional, electronic or computerised — depends upon its inherent creativity. Originality and inspiration have been at the heart of all successful games. With the advent of inexpensive programmable home computers, a completely new era of games opportunities has been created. Estimates suggest that by the end of 1985 there will be some 50 million owners of home video games delivery systems throughout the world, underlining the scope within the whole leisure electronic industry for new development in video/computer games.

Against this background comes 'The International Video Game of the Year'. Sponsored by Video Games International Limited (VGI), the competition carries a first prize of $100,000, advance against world wide royalties, plus five further advances of $15,000 each for other category awards. The joint promoters of the competition are The International Register of Independent Computer Programmers Limited (IRP) and Mark McCormack's organisation the International Management Group (IMG). VGI and IRP are both subsidiaries of Ashley Industrial Trust plc, a British public company quoted on the London Stock Exchange.

Through the administration and promotion of international competitions, IRP intend to create a register of independent computer programmers and their programs. This valuable library of all types of programs will be made available to distributors of microcomputer software throughout the world. 'The International Video Game of the Year' is the first in this series of competitions.

Throughout the world there is now a new breed of games inventors/programmers, devoting countless hours of creativity at home, college, or university to the development of inventions and original games. The competition, which is being advertised in over 20 countries, is expected to attract entries from many thousands of enthusiasts. The aim, however, is to find games that represent a breakthrough in basic creative and commercial flair. Where necessary IRP will re-program these to a highly professional standard and from all this just six winning games will emerge. These six games collectively will be termed 'The International Video Games of the Year' and could well produce not only the next Space Invaders or Pac Man, but also the microcomputer's answer to such classics as chess and backgammon.

All the most popular traditional games such as Scrabble and Monopoly were developed outside of the industry itself, and now that the tools of computer games invention have become available to all, it would seem logical that this new industry will follow the same path; and if the computer equivalents to Scrabble and Monopoly are to emerge from independent inventors, what more likely a vehicle than through the medium of 'The International Video Game of the Year?

The $100,000 first prize is, for instance, twice the prize money Miss World receives, and this together with the total prize money of $175,000 guarantees a world wide interest in the competition from press and public alike. The overall winners will be announced in a one-hour TV Awards Special, to be televised throughout the world by IMG, which has staged numerous international

television successes including 'International Superstars'.

The competition has been divided into six categories:-

- Sports
- Simulators
- Arcade
- Strategy
- Adventure/Fantasy
- 'Special'

The 'Special' award will cover the best program which cannot be classified as a 'game', but which provides educational or entertainment value. Individual awards of $15,000 each will be made for the winner in each of the five specific categories, and all shortlisted games will receive 'Merit' awards. 'Merit' games will be permitted to carry the title 'An International Video Game of the Year — 'Merit Award' on the retail packs.

CP/M FOR ADAM

Coleco's Adam, the ColecoVision Family Computer System, is the first home computer to offer Digital Research's new Personal CP/M. This has been made possible by an agreement between the two companies. It is anticipated that the Coleco Adam will be available in Europe during the second quarter of 1984.

Under the agreement, Coleco will manufacture Digital Research's Personal CP/M for Adam on both digital data packs and floppy discs. Coleco and Digital Research have also agreed to jointly develop many new software products for Adam.

Personal CP/M includes the ability to trap errors — a feature that reduces the amount of debugging needed to complete a software program, and an easy-to-use code for screen cursor control and graphics. It opens up the large base of CP/M software for Adam, and utilizes the Smart Keys, messages and other visual aids that let the user easily control the functions of the operating system.

WORDS ON APPLE

A new WP program is available from Pete & Pam called PFS:Write, which, they claim, gives you the power of a word processor with the simplicity of a typewriter. It's easy to learn, with functions in plain English and 'Help' instructions which appear at the touch of a key. The instruction manual is clearly written and easy to follow.

With PFS-Write, what you see on the screen is what you get when you print. You can see just how your document will look — where the pages will break, how text will appear underlined or boldfaced — right on the screen. So there are no surprises afterwards. PFS-Write lets you write, correct and revise your business letters, memos and proposals almost effortlessly. As you type, your words appear on the screen, which is designed to resemble a blank piece of paper.

PFS:Write gives your documents a professional touch with features like boldfacing and underlining, centering and justification, page headings and footings, and automatic page numbering. Its flexibility lets you adjust the page length and margin, print single or double spaced, even address envelopes. And PFS:Write works with all popular printers, so you're not locked into a few printer choices. PFS: Write is available for the Apple IIE and the IBM PC and comes at a retail price of £98 plus VAT. Pete & Pam Computers are at New Hall Hey Road, Rossendale, Lancashire BB4 6JG (telephone 0706 212321).

WRITING YOUR OWN

Dynatech Microsoftware has launched the first program generator on the market to cost less than £40. Called the Home Filewriter, the program generator is specifically aimed at the Commodore 64 and Atari user. Any Commodore 64 or Atari owner who also posseses a disc drive can now write highly sophisticated programs without needing to know any tricky computer jargon. All programming instructions are simply entered in plain English on the keyboard.

In practice, the user types a layout on the screen, and then instructs the software program to translate this information into the appropriate computer code. In this way screens can be created, edited and used for entry and modification of information quickly and simply.

The Home Filewriter is aimed firmly at the domestic market, and is ideal for creating data-base application programs. Examples include maintaining an inventory of household belongings, cataloging coin and stamp collections, home accounts, club membership listings, address and mailing files, social club records, recipe files, tax records and so on. The program is self-prompting, rejects illegal input and informs the user when an error has been made.

Already Dynatech has made great in-roads in the United States selling Home Filewriter. More than 3,000 copies a month are being sold. At the moment Dynatech is discussing Home Filewriter deals with several top chain stores in the UK. It comes complete with a manual and data entry program for £39.95. Dynatech Microsoftware are at Rue du Commerce, Bonet, St. Peter Port, Guernsey, Channel Islands.

GRAFFCOM GO CONCURRENT

The boss of Britain's longest-established micro software manufacturer has come out in favour of the industry's newest development — Concurrent CP/M. He has announced that the entire range of his company's software products, ISBS-F and the 2020 series will be available for Concurrent CP/M at the beginning of 1984.

Robert Owen, Managing Director of Graffcom Systems, said: "Concurrent CP/M is a great new product that has the capability of pushing back the frontiers of microcomputing. Multi-tasking allows a greater throughput of work which all adds up to increased productivity. This means that ultimately, the users get more out of their machines".

The 2020 series encompasses Graffcom's range of office products and includes WP2020, a word processing package, CM2020, a configurable manager and FP2020, a financial planner. WP2020 is already a successful product, as powerful as most dedicated word processing systems. It already has some 5,000 users in the UK. It forms the basis for O-MAN and is being exported to a number of Canadian companies including hardware manufacturer, Xerox Canada.

More details from Graffcom Systems, 7 Rickett Street, London SW6 (telephone 01-385 9422).

THE APL OF ITS IBM

I.P. Sharp Associates has announced the first of a series of products for IBM personal computers — Sharp APL/PC. This is a full implementation of APL and is completely compatible with I.P. Sharp's mainframe software products. All features of Sharp APL on the mainframe are present in the PC version, including full arithmetic precision and no restriction on using large variables — features noticeably absent in other implementations. Sharp APL/PC will be available for purchase in the first quarter of 1984.

Because Sharp APL/PC is

exactly the same language as Sharp APL for mainframes, it provides a well-proven system: it has been available on I.P. Sharp's public timesharing service since the early 1970's and has earned a considerable reputation for quality and reliability.

The Sharp APL/PC system runs as a user program under the IBM personal computer disc operating system, version 1.1. or 2.0. The Sharp APL/PC system includes the interpreter, the file system, the session manager, several auxiliary processors, and utility software. The auxiliary processors provide access to the operating system, DOS files, asynchronous communications, the parallel printer, and a full screen manager.

The cost of Sharp APL/PC is £250 with substantial volume discounts. Each copy includes a diskette containing the Sharp APL/PC pocket reference, the Sharp APL/PC handbook, and APL stickers for the PC's keyboard. APL character ROMs enabling an IBM personal computer to display APL characters are also available from I.P. Sharp. Product enquiries to Elaine O'Donovan, I.P. Sharp Associates, 132 Buckingham Palace Road, London SW1 (telephone 01-730 4567).

VISI-CORP LAUNCH VISI-ON

Visi-Corp, one of America's leading Software houses, and producers of the ubiquitous Visi-Calc spreadsheet package (amongst other things), have unveiled their latest and possibly most exotic offering to date. Visi-On, a powerful new "applications environment" comprises an 'applications-manager' and four optional applications packages.

Visi-On-Calc is a spreadsheet package not unlike VisiCalc, (although Visi-Corp reps continually stress the distinction!), while Visi-On-Graph rapidly converts raw data into pictorial form.
Visi-On-Word is, as the name implies a word-processing package, and Visi-On-Query completes the list of soft wares, being a database management system. Visi-On-Mouse is the name Visi-Corp have given their mouse — although I smell a rat! This is, to the best of my knowledge, the first time that Visi-Corp have produced such hardware, and I am left wondering whether other mice are compatible with the system.

The Applications Manager is an essential pre-requisite to any of the other packages, interacting with the host machine's operating system and acting as a buffer between the OS and the applications packages in use. According to Visi-Corp, this means that only the applications manager is machine-dependant: hence the speed with which versions have been produced for several machines.

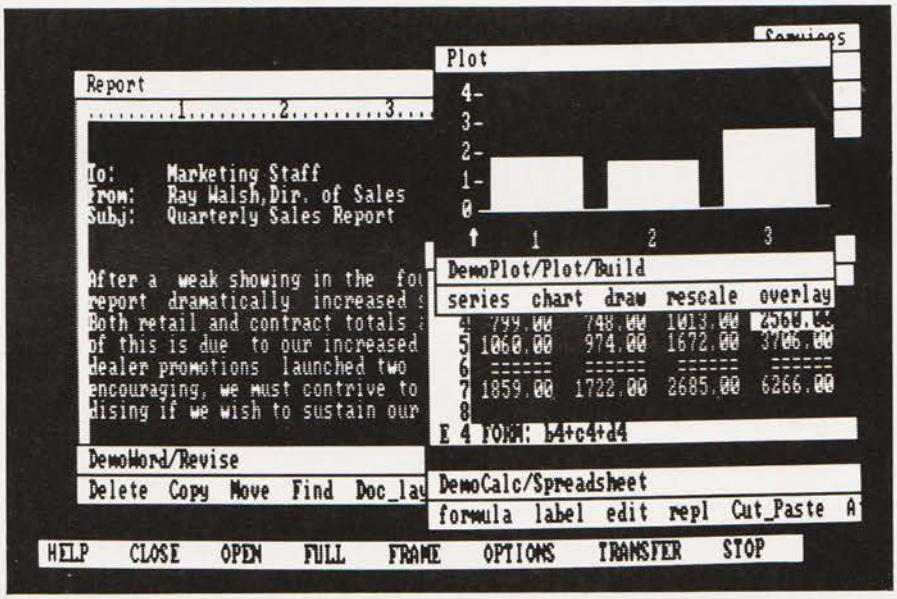
The system, demonstrated on the IBM Personal Computer (complete with Winchester and mouse) proved to be all that Visi-Corp had promised. Gone were the endless incomprehensible menu options, discreetly replace by a single 'window' at the bottom of the display containing explicit descriptions of the fundamental commands available to the user. Another window, situated at the top righthand corner of the screen shows a stack representing the files or 'jobs' currently under preparation, either or all of these files can be removed from the stack, and displayed upon the main working are of the screen.

With Visi-On, the system represent each new file, or a file lifted from the stack, as a sheet or 'window' on the screen. This is designed to symbolise the all-too-familiar desktop, where sheets can be stacked on upon another.

The Visi-On system is indeed very powerful, and yet very simple to use. Visi-Corp's philosophy of never leaving the user in the dark is followed throughout.

At times Visi-On seemed to be a little slower than one would have expected, especially when addressing the hard disc, and also at times as the windows were being updated. But in all fairness, this delay would not be evident to the everyday user.

Visi-Corp have guaranteed themselves a generous share of future markets with this one, by taking the uprecedented step of publishing and circulating the details and specifications for the applications manager to several major American software houses. Visi-Corp have assured us that, in principle at least, absolutely **anyone** can develop tailored applications packages around the manager. However, don't expect to tailor packages on your IBM PC — as a spokesman for Rapid Terminals, who are marketing Visi-On in this country, admitted at the launch — "You'll probably need at least a VAX to do it"!



Option selection has been designed to be as simple as possible. A 'screen pointer', controlled by the mouse, is displayed in inverse-video. In fact, the system goes further still, and displays a concise description of the effect of the command if it were selected. The user can then engage the command by depressing a 'select' key situated on the mouse. Any other option required can be selected in much the same way.

Each window can be enlarged and shuffled around for convenience, with the window containing the file currently under preparation double-bordered for clarity.

Versions of Visi-On are currently available for the IBM PC and XT, with versions for machines from Texas Instruments, Honeywell, Wang Compaq and Xerox to follow shortly. Prices (excluding VAT) are as follows:

Visi-On Applications Manager	£375
Visi-On-Calc	£295
Visi-On-Graph	£195
Visi-On-Word	£285
Visi-On-Query (Available early '84)	£285
Visi-On-Mouse	£185

For further information, contact Rapid Terminals, Rapid House, Denmark Street, High Wycombe, Bucks HP11 2ER (telephone 0494 26271).

LORDS OF TIME

Joins our range of acclaimed pure-text puzzle adventures, at £9.90, for:

BBC 32K ***COMMODORE 64*** ***SPECTRUM*** 48K ***LYNX*** 48K ***NASCOM*** 32K ***ORIC*** 48K ***ATARI*** 32K

ADVENTURE REVIEWS

"Adventures which have a fast response time, are spectacular in the amount of detail and number of locations, and are available to cassette owners . . I am extremely impressed . . The Level 9 Adventures are superbly designed and programmed, the contents first rate. The implementation of Colossal Cave (Adventure) is nothing short of brilliant; rush out and buy it. While you're at it, buy their others too. Simply smashing!"

– SOFT, Sept 83

"I found Dungeon exceedingly well planned and written, with a fast response. There are well over 200 locations and the descriptions are both lengthy and interesting. The objects number about 100. It could therefore take some months to explore the whole network, giving many hours of enjoyment in the process."

– C&VG, Sept 83

"The descriptions are so good that few players could fail to be ensnared by the realism of the mythical worlds where they are the hero or heroine . . .great fun to play."

–Which Micro?, Aug 83

"My appetite has been whetted and I intend to get my own copy (of Snowball) to play."

– What Micro?, Dec 83

ADVENTURE REVIEWS

"This has to be the bargain of the year. If adventures are your game then this (Colossal Adventure) is your adventure."

– HCW, 5 Sept 83

"Colossal Adventure is simply superb. Anyone who wishes to use adventures in an educational setting really must use and see this program as it emulates Crowther and Wood's masterpiece so well. For those who wish to move onto another adventure of similar high quality, Dungeon Adventure is to be recommended. With more than 200 locations, 700 messages and 100 objects it will tease and delight!"

– Educational Computing, Nov 83

Colossal Adventure is included in Practical Computing's Top 10 games choice: "Poetic, moving and tough as hell."

– PC, Dec 83

"To sum up, Adventure Quest is a wonderful program, fast, exciting and challenging. If you like adventures then this one is for you"

– NILUG # 1.3

"Colossal Adventure . . For once here's a program that lives up to its name . . a masterful feat. Thoroughly recommended"

– Computer Choice, Dec 83

"wholly admirable"

– Your Computer, Sept 83

MIDDLE EARTH ADVENTURES

1: COLOSSAL ADVENTURE
A complete, full size version of the classic mainframe game "Adventure" with 70 bonus locations added.

2: ADVENTURE QUEST
Centuries have passed since the time of Colossal Adventure and evil armies have invaded The Land. The way is long and dangerous; but with cunning you can overcome all obstacles on the way to the Black Tower, source of their demonic power, and destroy it.

3: DUNGEON ADVENTURE
The trilogy is completed by this superb adventure, set in the Dungeons beneath the shattered Black Tower. A sense of humour is essential!

THE FIRST SILICON DREAM ADVENTURE

1: SNOWBALL
The first of Pete Austin's second trilogy. The giant colony starship, Snowball 9, has been sabotaged and is heading for the sun in this massive game with 7000 locations.

THE LORDS OF TIME SAGA

7: LORDS OF TIME
Our congratulations to Sue Gazzard for her super design for this new time travel adventure through the ages of world history. Chill to the Ice-age, go romin' with Caeser's legions, shed light on the Dark Ages etc. etc. We'll be selling this game mail-order from January 1st.

Price: £9.90 each (inclusive)

Level 9 adventures are available from good computer shops, or mail-order from us at no extra charge. Please send order, or SAE for catalogue, to:

LEVEL 9 COMPUTING

Dept. CT, 229 Hughenden Road, High Wycombe, Bucks HP13 5PG

Please describe your Computer

CALPAC LEARNING SERIES

Colour graphics and sound are used extensively to make learning more enjoyable.

5. MULTIPLICATION TABLES. This program with correction sequences enables you to test individual multiplication tables, or all tables up to a maximum that you choose. Carriages are added to a train as the score increases – see the train glide through the countryside when the test is finished.

6. THE ROMANS. Answer questions on the Romans to win a horse, chariot, viaduct and temple. Answer enough correctly and the horse will gallop away with its chariot.

7. SPELLING TESTER. The words in the test are initially displayed on the screen. Then short sentences are used as prompts for the words, which must by typed correctly to add stages to a space-ship.
Replace the words and sentences in the program with weekly spelling lists, save them on cassette, and you will rapidly create a whole library of programs.

8. HOMOPHONES. Sale or sail; bare or bear? Choose the correct word to complete the sentence. Extra motivation is provided by a coloured pot-plant, which grows with each correct response. The plant flowers in an unusually tuneful fashion at the end of the program.

Published by Calpac Computer Software, 108 Hermitage Woods Crescent, St Johns, Woking, Surrey GU21 1UF

CALPAC

CALPAC LEARNING SERIES

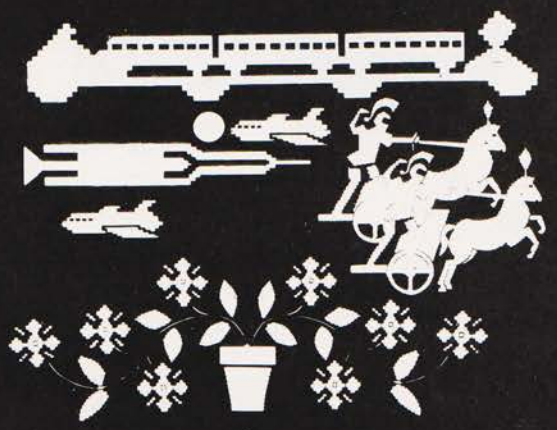
ZX SPECTRUM

VOL. 2

CALPAC

CALPAC LEARNING SERIES

PROGRAMS 5-8



5. MULTIPLICATION TABLES
6. THE ROMANS
7. SPELLING TESTER
8. HOMOPHONES

SPECTRUM 16K 48K
FROM 7 YEARS

CALPAC LEARNING SERIES

Colour graphics and sound are used extensively to make learning more enjoyable.

9. NOUNS, VERBS, ADJECTIVES AND ADVERBS. Sentences are displayed and you select the named parts of speech. Instant feedback and helpful comments ensure positive learning. Correct answers are rewarded by the creation of a coastal scene, which is animated at the end of the program.

10. VERB PRACTICE. You have to complete the sentences using the correct tenses of the verbs. The program concentrates on those irregular verbs that often cause difficulty. An underwater landscape is created as questions are correctly answered.

11. THE STRUCTURE OF THE FLOWER. This program explains how the parts of the flower are involved in the formation of seeds. This is a three part program which makes full use of high resolution colour graphics.

12. LONG DIVISION. This detailed program takes the learner through long division sums in easy stages. Correction sequences are automatically provided when they are needed. Sums with remainders can be chosen if required.

Published by Calpac Computer Software, 108 Hermitage Woods Crescent, St Johns, Woking, Surrey GU21 1UF

CALPAC

CALPAC LEARNING SERIES

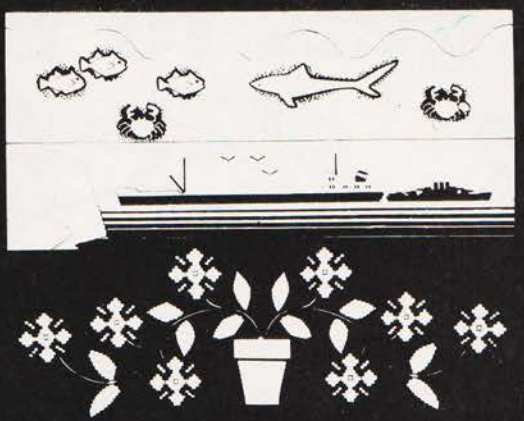
ZX SPECTRUM

VOL. 3 FROM 9 YEARS

CALPAC

CALPAC LEARNING SERIES

PROGRAMS 9-12



9. NOUNS, VERBS, ADJECTIVES AND ADVERBS
10. VERB PRACTICE
11. THE STRUCTURE OF THE FLOWER
12. LONG DIVISION

SPECTRUM 16K 48K
FROM 9 YEARS

Our software is suitable for use in the home, school or college.

For further details or retail stockists please telephone:- 048 67 2584

We have a demonstration cassette available for retailers or schools.

We would like to hear from good programmers, graduates or teachers wishing to participate in the expansion of our Learning Series on the Spectrum, BBC Model B and other machines.

Our software is available by direct mail from:-

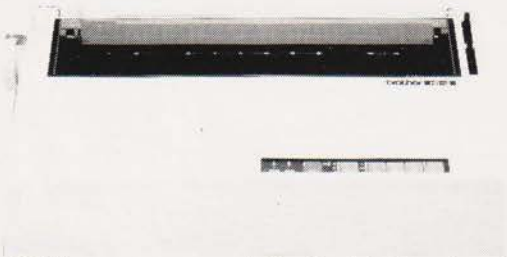
CALPAC COMPUTER SOFTWARE
108 Hermitage Woods Crescent
St Johns, WOKING, Surrey GU21 1UF

PRICE LIST

CALPAC LEARNING SERIES VOL 1	16K or 48K Spectrum	£9.50
CALPAC LEARNING SERIES VOL 2	16K or 48K Spectrum	£9.50
CALPAC LEARNING SERIES VOL 3	16K or 48K Spectrum	£9.50
CALPAC CHEMISTRY SERIES VOL 1	48K Spectrum	£7.50
CALPAC CHEMISTRY SERIES VOL 1	16K ZX81	£6.95
CALPAC PATTERNS	16K ZX81	£5.95

Additional features of the CALPAC LEARNING SERIES include:-
*Spelling checkers
*"Help" call up routine
*Easy insertion of subject material of your own choice into the programs

SPECIAL OFFER
Brother HR-15 Printer
~~£455.00~~ £399.00



17 Chrs per second. Perfect daisywheel print. Ideal for home and business use.

Printer for you . . .
. . . Typewriter for wife

SPECIAL PRICE £399 + VAT

will run Sinclair, BBC, Dragon, Sharp, Sirius, Vic 64, Apple, virtually any computer.

BROTHER CE-50 DAISYWHEEL TYPEWRITER
Perfect print quality. Lift off correction. Coloured carbon ribbons - 10 million chrs - extra daisywheel - 18 type styles.

PRINTER
13 Chrs per sec – ASC II wheel available – medium duty use – parallel or serial. Full **K.S.R.** – only £30.00 carriage paid

CROWN BUSINESS CENTRE
56 - 58 SOUTH STREET, EASTBOURNE,
EAST SUSSEX. (0323) 639983/20496

At last, a magazine that gets to the core of it.

As the proud owner of an Apple, you know that really helpful magazines don't grow on trees.

But now there's Orchard Computing. Orchard is a brand new quarterly put together specifically to suit the Apple user.

It brings you the very best in useful routines, business and games software.

And not a word of it that isn't directly relevant to you.

So if you like an Apple a day, take Orchard quarterly.

It's just what the doctor ordered.

Orchard
COMPUTING
Britain's best buy for Apple users

Simon N. Goodwin

EASYCODE PART 1

So that we can teach you the principles of machine code programming without worrying about what microprocessor you've actually got, this series uses a BASIC simulation. Clever, huh?

This series of four articles is aimed at anyone who would like to learn to program in machine code, the fastest and most intricate programming language on any home computer. Machine code is generally tens of times faster than BASIC; it is the language used for most sophisticated games and business programs.

This series will explain the principles which underly machine code programming rather than the nitty-gritty details of how to program the Z6509A 11-bit NMOS CPU with on-chip TTL I/O! The examples can be run on **any** popular computer which supports BASIC and a TV display.

MEAN MACHINES

When you set out to learn machine code, two major problems are likely to stand in your way. The first pitfall for the budding wizard is the unfriendliness of most 'monitor' programs. Monitors are programs which let you write machine code, in much the same way as a word processor lets you write letters.

Most monitor programs expect the user to type cryptic one-letter commands or devious mnemonics. On a Spectrum or TRS-80 the unpronounceable word "LDIR" means 'copy memory'. "3D0G" means 'go to BASIC' to the Apple monitor. Similar examples abound.

Things become even worse when you try to test your program. In BASIC, if you mis-type a line number you are told something like "Line not found". Most monitors can't detect such a mistake — the machine code will zoom off to a non-existent line, usually with non-satisfactory results! From that point onwards the computer is out of your control.

Mistakes like this often cobble the entire contents of the computer's memory, forcing you to reset the machine and load your program all over again. The sheer speed of machine code makes it hard to diagnose the exact location and time of an error.

NEW IDEAS

The second problem to be faced by a would-be machine code programmer is the intricacy of the language. Machine code is different for every type of processor, but all of the cheap processors incorporate these principles:

- The idea that data and program are equivalent
- The idea of storing information on a 'stack'
- 'flags' and 'registers'
- 'addressing modes'
- Number representation (binary, decimal, hex)
- 'bit' manipulation

None of the principles are very complex, but most of them must be understood thoroughly before any useful programming can be done. To those six principles a seventh should perhaps be added:

- Jargon!

since, like other areas of programming, machine code has spawned a new vocabulary for humans as well as for computers. Jargon is, within limits, an efficient way of communicating ideas, so this series will not shy away from it. However, unlike the people you meet at computer shows, we will do our best to explain what we mean by each word before it is used!

THE UNIVERSAL CODER

This series features a standard BASIC program, developed and refined over three years, which demonstrates the first four points. Once these are understood it is easy to see the relevance of the others. The demonstration should give you the confidence to move from this 'model' machine code to the real thing. At the end of the series we will duscuss the differences between the demonstration and real machine code.

Program 1 is the 'toolkit' which you will use to teach yourself machine code. You probably remember how BASIC didn't really make sense until you got hold of a computer and actually used the language. 'Easycode' (Program 1) lets you learn machine code the same way, by combining the essence of machine code with the messages and safety-checks of BASIC.

Easycode is presented in two parts. The first part, listed this month, is a complete toolkit which allows you to program in simple 'machine code'. The second part will add extra facilities, allowing you to experiment with 'assemblers', 'disassemblers' and 'stack operations'. These terms will be explained, by example, in Parts 3 and 4 of the series.

Program 1 provides all the facilities you need to enter, modify and test machine code programs. You can also store your work on tape for retrieval later. The next listing will consist of lines to be added to the first program — it won't be a program in itself.

EASYCODE

There are two places where information can be stored in a computer — the memory and the processor. Easycode lets you watch information (programs and data) being copied and manipulated inside the computer. The program is a kind of 'computer simulator'. A typical display is shown in Fig 1.

Easycode simulates a computer with 100 'memory locations', numbered from 0 to 99. A 'memory location' is a storage space within a computer — a kind of electronic pigeon-hole in which a single value can be stored. Sometimes a memory location is referred to as an 'address' — the name of a place within a computer.

```
 0: 1 10  0  0  0  0  0  0  0  0
10: 0  0  0  0  0  0  0  0  0  0
20: 0  0  0  0  0  0  0  0  0  0
30: 0  0  0  0  0  0  0  0  0  0
40: 0  0  0  0  0  0  0  0  0  0
50: 0  0  0  0  0  0  0  0  0  0
60: 0  0  0  0  0  0  0  0  0  0
70: 0  0  0  0  0  0  0  0  0  0
80: 0  0  0  0  0  0  0  0  0  0
90: 0  0  0  0  0  0  0 42  0

(A=10).(X=0 ).(P=2 ).(Z=N).(C=N)

Command?

Halt at 2
```

Fig. 1 The Easycode display.

Easycode shows the contents of a memory location as a whole number which may range between 0 and 99. That number might represent a letter of the alphabet, or a colour, or anything you like. In Fig. 1 you can see that location 1 contains the value 10. Location 98 holds the value 42. (Every computer should contain 42 somewhere!). Most of the memory contains the value 0.

The top 10 rows of the display show the contents of the computer's memory, from location 0 in the top left corner to location 99 at the end of the tenth line. The leftmost column is an index. All of the values in the other columns can be altered — they always show the value in the appropriate memory location. The effect is rather like having an integrated circuit with a glass top — you can read the computer's memory.

The twelfth line of the display shows the contents of the processor. Broadly speaking a processor does four things:

- It fetches values from memory
- It alters values once they are fetched
- It stores values in memory
- It changes the sequence of operations performed depending upon the values it contains

The processor must have some memory of its own, so that it can remember values once it has fetched them. Memory locations inside a microprocessor are called 'registers'. There are usually between two and 20 useful registers in a processor. Easycode has three registers, named A, X and P (registers often have one or two-letter names). The value stored in each register is shown, next to its name, on the twelfth line of the display.

The 'A' register, or 'Accumulator', is used to hold the temporary results of calculations. The 'X' or 'Index' register contains either results or the 'index number' (address) of a memory location. Index registers are used by a computer to 'mark its place' in data.

The 'P' register is the 'Program Counter'. Every microprocessor has a program counter. It contains the address of the 'instruction' which is being executed.

INSTRUCTIONS EXPLAINED

A computer decides what operation to carry out by examining values in memory. These values are called 'instructions' — different values cause different operations to be performed. One value might mean stop, another might mean 'JUMP' (change the value of the program counter) and so on. A machine code 'program' is just a sequence of instructions.

Instructions are numbers fetched when the program counter indicates them. Values fetched for any other reason are 'data'. There's no reason why some locations shouldn't be both data and instructions at different times. The computer can produce data and then JUMP to it and treat it as instructions. This is potentially a very useful trick, which we will explore later in this series.

The program counter determines which location is examined for the next instruction. Normally the computer steps from low-numbered locations to higher ones, just as BASIC goes from one line-number to the next. Some values may cause the computer to skip locations or go back to a lower-numbered location — these machine code instructions correspond to the 'GOTO' command of BASIC.

The last important components of a processor are the 'flags'. These operate rather like railway points, telling the computer whether it should go ahead (to the following instruction) or turn elsewhere. Easycode has two flags, labelled 'Z' for 'Zero' and 'C' for 'Carry'. Each flag may have two values — 'set' or 'reset'. Some computer makers take a high moral tone and label these values TRUE and FALSE respectively. Easycode shows a 'Y' next to a flag's name when it is set, and an 'N' when it is reset. The significance of these flags will become clear later.

CLEAR	Blanks Easycode's memory and registers
HELP	Shows a summary of the commands
LOAD	Read memory contents from tape (or disk)
QUIT	Stops the Easycode BASIC program
RUN	Starts a machine code program
SAVE	Store memory contents on tape (or disk)
STORE	Changes the value at any memory location

Table 1. Easycode commands.

PLEA AND JUDGEMENT

The last two lines of the Easycode display are used for commands and messages. Reports from the computer appear on the bottom line. Commands are typed in capital letters on the line above. Table 1 is a complete list of the commands recognised by Program 1.

Since instructions and data are stored identically, the STORE command can be used to enter any kind of information. If you type STORE while the prompt 'Command?' is displayed the computer responds 'Address?'. 56A Stalingrad Mansions won't do — you must type the number of the memory location you wish to alter. If the value you type is not in the range 0 to 99 you are asked for another command.

Assuming that you typed a valid address, Easycode asks you for the value to be stored at that address. Enter another value between 0 and 99. Easycode takes it and stores it in memory. If you watch the display you will see the value appear.

Next you are asked for a value to be stored in the subsequent location. The sequence continues until the end of memory is reached or you type an invalid number (such as 100). You are then asked for a different address. Either select a new address and enter values as before or type 100 to halt the store operation and return to the 'Command?' prompt.

A SIMPLE PROGRAM

The simplest BASIC program is:

1 STOP

To write this in Easycode all we need to know is the instruction value which will cause a program to halt. Table 2 contains a full list of Easycode instructions. At the head of the list is 0 HALT — when the computer encounters a '0' instruction it will stop and display the value of the program counter.

In case of accidents, all of the computer's memory is filled with 0's when Easycode is first run. Wherever we start our program it will encounter a HALT and stop immediately. To confirm this, type RUN and then enter any address. Easycode loads the number you type into the 'P' register and then executes the instruction there. Notice that each memory location is flashed as Easycode reads an instruction from it.

We'll try something slightly more complicated next. The next program contains two instructions (wow). The first instruction loads a value into the 'A' register and the second one is the HALT which we have come to know and love.

If you consult Table 2 you will see that instruction value 1 means 'LOAD A;n'. This instruction takes up two memory locations (HALT only took one). The first location contains the instruction (1). The following location contains the data to be loaded, so

that the sequence of values 1 2 will cause the value 2 to be loaded into the A register. Use the STORE command to put the values 1 2 0 into memory from location 0 onwards.

When you RUN from location 0 you will see the 1 flash, then the value 2 will appear in the A register. The computer skips over location 1 (because it is data — part of the LOAD instruction) and flashes the contents of location 2 — the HALT.

ASSEMBLER MNEMONICS

It may seem rather pointless to use these odd names: 'HALT', 'LOAD A;2' and so on when we can't type them into the computer — we have to use the numeric values from Table 2 instead. These names are called 'mnemonics', (pronounced nem-on-iks) which is Greek for 'reminders', and they're designed as an aide memoire for programmers. The idea is that a sequence such as:

0: LOAD A;1
2: ADD;1
4: JUMP;2

makes a little more sense than the string of digits 1 1 5 1 10 2! Most machine codes use mnemonics, although they vary in detail from one processor to another. Easycode has 20 instructions (more will be added later) and consequently 20 mnemonics, listed in Table 2. Each mnemonic has a name (eg HALT, ADD, LOAD) and most of them have 'arguments' too — these describe what information is used and where it is stored. The instruction LOAD A;1 corresponds to A=1 in basic. ADD A;1 corresponds to A=A+1. JUMP;2 is similar in effect to GOTO 2.

The format of Easycode mnemonics is very similar to that of real machine code, although a semi-colon is used as a separator rather than a comma since BASIC INPUT statements tend to do strange things with commas!

Enter the sequence 1 1 5 1 10 2 into memory from location 0 onwards. When you RUN the program (starting at 0 once again) you will see Easycode counting in the A register. Watch the display as Easycode counts. Locations 2 and 4 flash alternately as the instructions within are executed. The value in the program counter P changes back and forth, and the accumulator A counts up steadily. You can pause the program at any point by pressing the SPACE key. Type an end of line to stop the program or any other key to re-start it. You can use the end of line key to halt the program immediately if you wish.

Code	Mnemonic	Purpose
0	HALT	Stop machine code program
1	LOAD A;n	Put next memory contents (n) in A.
2	LOAD A;@n	Put contents of address n in A.
3	STORE A;@n	Put contents of A at address n.
4	LOAD A;X	Copy contents of X into A as well.
5	ADD A;n	Add next memory contents to A.
6	SUB A;n	Subtract next memory contents from A.
7	SUB A;@X	Subtract the contents of the address numbered in X from the contents of A.
8	JUMPNC;n	Go to address n if carry is not set.
9	JUMPNZ;n	Go to address n if zero is not set.
10	JUMP;n	Go to address n.
11	LOAD X;n	Put next memory contents (n) in X.
12	LOAD X;@n	Put contents of address n in X.
13	STORE X;@n	Put contents of X at address n.
14	LOAD X;A	Copy contents of A into X as well.
15	ADD X;n	Add next memory contents to X.
16	SUB X;n	Subtract next memory contents from X.
17	LOAD A;@X	Put the contents of the address numbered in X in the A register.
18	STORE A;@X	Put number in A at the address in X.
19	ADD A;@X	Add the number at address X to A.

'n' represents any value between 0 and 99.

Table 2. Easycode instructions (8K version).

USING THE FLAGS

If you let the count continue all the way up to 99 you will see something interesting happen. When A contains 99 and 1 is added there isn't room for the value 100. Easycode, like all machine codes, simply throws away the extra digit — the one — and counts from 0 again. The computer has, in effect, said '99 plus 1 is 0, carry 1'. When the value in A 'overflows' the carry flag becomes set — the display shows 'C=Y'.

The carry flag is set whenever an operation results in a carry or a borrow. When you try to SUBtract 1 from 0 you will get 99 borrow 1 — the register will hold 99 and, once again, the carry flag will be set.

The zero flag works in a similar way, but it becomes set whenever an operation ends up with a zero value. The zero flag also becomes set if you LOAD or STORE a zero. You can use this rule to test for any value — just load the number to be tested into the A register and subtract the value you want to test for. If the zero flag is set after that, you know that the number and the expected value were the same.

This flag-waving is all very well, but it seems rather pointless unless we can tell the computer to make decisions depending on the value of the flags. There are two Easycode instructions which test the flags, doing different things depending upon what they find. The instruction JUMPNZ;n tells the computer to JUMP to the instruction at location 'n' if the zero flag is NOT set. If the flag **is** set, the computer simply skips over the JUMPNZ and performs the subsequent instruction. The JUMPNC;n instruction is identical except it tests the carry flag, producing the effect of the BASIC line:

IF C <> Y THEN GOTO n

It is easy to see how we can use this instruction. Change the contents of locations 3 and 4 to 10 and 8 respectively. Now our program is:

0: LOAD A;1
2: ADD A;10
4: JUMPNC;2

The program now counts quickly until it tries to add 10 to 91 — the result is 1, carry 1, and the program 'falls through' to location 6.

One important thing to note about

machine code is that the computer can't tell instructions and data apart. This can have unfortunate consequences if you jump to the wrong address. Consider what would happen if we started the above program at address 1 instead of 0 . . .

The computer finds a 1 at address 1. It treats that as LOAD A;next, and puts the value 5 (the ADD instruction!) into A. Next it finds the 10 at address 3. 10 means JUMP, so it jumps to the address in location 4 — an 8. Notice that we've ended up with a completely different program, simply by starting one location later.

Sometimes mistakes like this will cause the computer to try to execute a non-existent instruction — a value greater than 19, for instance. A real computer might do unpredictable things in such a circumstance, but Easycode can detect the error. If you make that kind of mistake Easycode stops and prints the message 'Unknown Instruction'.

THE PROGRAM

Program 1 is a complete listing of the Easycode program for the TRS-80 Model 1 or Video Genie. The only requirements are a display at least 32 columns wide and 16 lines long, 8K of user memory, string handling, and a BASIC which allows characters to be read from the keyboard as a program runs. The expanded version of the program requires a 40 by 16 display (or larger) and 16K or memory.

The listing is extensively commented, so that it should be possible to work out the effect of instructions from the listing even if you can't make it out by experimentation. Some parts of the program have been deliberately kept simple rather than efficient, on the grounds that it is better to have a slow correct program than a speedy one which doesn't always work!

Once you have converted the program it should be easy to identify the parts which can be accelerated. Keep to the same line numbers as much as possible, since this will make it easier to add the extra instructions introduced in Part 2. On a Spectrum or ZX81 you should divide all of the line-numbers by five.

Next month we'll explain how to convert the program for almost every machine under the sun, and we'll demonstrate multiple precision arithmetic, input-output and even moving graphics. Don't miss it!

Program 1. The 8K version of Easycode.

```
100 REM ** EASYCODE Small Version.
110 REM ** (C) 1983 Simon Goodwin

1000 CLEAR 100 ' Set variables to zero
1010 DIM R(1),M(99)
1020 MAX=19 ' Highest instruction code
1050 GOSUB 9000 ' Set up display
1060 GOTO 10000 ' Get command
3490 REM ** Poll keys; Space=wait, <CR>=abort
3500 T$=INKEY$ ' keyboard scan
3510 IF T$=CHR$(13) THEN ABRT=1
3520 IF T$<>CHR$(32) THEN RETURN
3530 ROW=15
3540 COLUMN=1
3550 GOSUB 6000 ' Position cursor on message line
3560 PRINT"Waiting at";P;
3570 PRINT"Press a key";
3580 T$=INKEY$
3590 IF LEN(T$)=0 THEN 3580 ' No key yet, loop
3600 GOSUB 4000 ' Scrub the message
3610 GOTO 3510
3990 REM ** Clear line (leave cursor at start)
4000 GOSUB 6000 ' Position cursor
4010 FOR J=1 TO 32
4020 PRINT " "; ' One space
4030 NEXT J
4040 GOSUB 6000 ' Reset cursor
4050 RETURN
4490 REM ** Read number 0-99 to N (100=error)
4500 INPUT T$
4510 IF T$<"0" OR T$>=":" THEN 4570 ' Not digit
4520 N=VAL(T$)
4530 IF N<0 THEN 4570
4540 IF N>99 THEN 4570
4550 IF N<>INT(N) THEN 4570 ' F.P. Basic only
4560 RETURN ' No error
4570 ROW=15
4580 COLUMN=1
4590 GOSUB 6000 ' Cursor to message line
4600 PRINT"# Number beyond range 0-99";
4610 N=100
4620 RETURN ' Error found
4990 REM ** Update display of registers & flags
5000 CARRY=0
5010 ZERO=0
5030 IF R(K)>=0 THEN 5060
5040 R(K)=R(K)+100
5050 GOTO 5080 ' Set carry
5060 IF R(K)<100 THEN 5090
5070 R(K)=R(K)-100
5080 CARRY=1
5090 IF R(K)=0 THEN ZERO=1
5100 ROW=12
5110 COLUMN=25
5120 GOSUB 6000 ' Set up for zero flag
5130 PRINT"N";
5140 GOSUB 6000 ' Position cursor
5150 IF ZERO=1 THEN PRINT"Y";
5160 COLUMN=31
5170 GOSUB 6000 ' Set up for carry flag
5180 PRINT"N";
5190 GOSUB 6000 ' Position cursor
5200 IF CARRY=1 THEN PRINT"Y";
5210 COLUMN=4
5220 N=R(0)
5230 GOSUB 8000 ' Update accumulator display
5240 COLUMN=11
5250 N=R(1)
5260 GOSUB 8000 ' Update X register display
5270 GOTO 11500 ' Get next instruction
5490 REM ** Mark and update the current locn.
5500 ROW=INT(P/10)+1 ' F.P Basic only
5510 COLUMN=(P-10#ROW)#3+34
5520 GOSUB 6000 ' Put the cursor there
5530 PRINT"  "; ' <2 SPC>
5540 ROW=12
5550 COLUMN=18
5560 N=P
5565 K=I ' Save instruction code
5570 GOSUB 8000 ' Update Program counter
5580 N=M(P)
5590 I=P
5600 GOSUB 16500 ' Redraw the current locn.
5610 GOSUB 3500 ' Poll the keyboard
5615 I=K ' Restore instruction code
5620 RETURN
5990 REM ** Position cursor at column & row
6000 PRINT @ COLUMN+ROW#64-65,"";
6010 RETURN
7990 REM ** Print N at current coordinates
8000 GOSUB 6000
8010 T$=STR$(N)+" " ' Force 2 character width
8020 PRINT MID$(T$,2,2); ' in range 0-99
8030 RETURN
8990 REM ** Draw the display
9000 I=0
9005 CLS ' Clear screen
9010 FOR ROW=1 TO 10
9020 COLUMN=1
9030 N=ROW#10-10
9040 GOSUB 8000 ' Print the 'index'
9050 PRINT":";
9060 FOR J=1 TO 10
9070 COLUMN=J#3+1
9080 N=M(I)
9090 GOSUB 8000 ' Print one memory element
9100 I=I+1
9110 NEXT J,ROW
9120 ROW=11
9130 COLUMN=1
9140 GOSUB 6000 ' Position cursor
9150 FOR I=1 TO 32
9160 PRINT"-";
9170 NEXT I
9180 ROW=12
9190 GOSUB 6000 ' Position cursor
9200 PRINT"(A=00).(X=00).(P=00).(Z=N).(C=N)";
9210 ROW=13
9220 GOSUB 6000 ' Position cursor
9230 FOR I=1 TO 32
9240 PRINT"-";
9250 NEXT I
9310 ROW=15
9320 GOSUB 6000 ' Cursor on message line
9330 PRINT"EASYCODE (C) 1983 Simon Goodwin.";
9340 RETURN
9990 REM ** Get the user's next command
10000 ROW=14
10010 COLUMN=1
10020 GOSUB 4000 ' Clear prompt line
10030 PRINT"Command";
10040 INPUT T$ ' Force caps if need be
10050 ROW=15
10060 GOSUB 4000 ' Clear message line
10070 IF T$="RUN" THEN 11000
10080 IF T$="SAVE" THEN 12000
10090 IF T$="LOAD" THEN 13000
10100 IF T$="HELP" THEN 14000
10110 IF T$="QUIT" THEN 15000
10120 IF T$="CLEAR" THEN RUN ' Start again
10130 IF T$="STORE" THEN 16000
10140 REM ** Line reserved for disassembler
```

```
10150 REM ** Line reserved for assembler
10160 PRINT "* ";T$;" is not a valid command";
10170 GOTO 10000
10490 REM ** Program has been stopped
10500 ROW=15
10510 COLUMN=1
10520 GOSUB 6000 ' Prepare for message
10530 PRINT"* Program stopped";
10540 GOTO 10000 ' Get a command
10990 REM ** 'RUN' command pre-processor
11000 COLUMN=1
11010 ROW=14
11020 GOSUB 4000 ' Clear the prompt line
11030 PRINT "Start address";
11040 GOSUB 4500 ' Get the start of the program
11050 IF N>99 THEN 10000 ' Error
11060 P=N ' Set the program counter
11070 ABRT=0 ' Clear the abort flag
11490 REM ** 'RUN' main loop for each instruction
11500 I=M(P) ' Get next instruction
11504 GOSUB 5500 ' Update display, check keys
11506 IF ABRT=1 THEN 10500  ' Quit if requested
11510 COLUMN=1
11520 ROW=15
11530 GOSUB 6000 ' Put cursor on message line
11540 IF I<1 THEN 11560 ' Halt code
11550 IF I<=MAX THEN 11600 ' Other instruction
11560 IF I=0 THEN PRINT"HALT";
11570 IF I<>0 THEN PRINT"* Unknown instruction";
11580 PRINT" at";P;
11590 GOTO 10000 ' Get next command
11600 IF P<>99 THEN 11630 ' Not end of memory
11610 PRINT"* No end on program";
11620 GOTO 10000 ' Get next command
11630 P=P+1
11640 J=M(P) ' Get operand
11650 P=P+1 ' Point to next instruction
11660 K=0 ' Assume a register A instruction
11670 IF I>10 THEN IF I<17 THEN K=1 ' Wrong ! Register X
11680 ON I GOTO 20000,20100,20200,20300,20400,20500,
                20600,20700,20800,20900,20000,20100,
                20200,20300,20400,20500,21000,21100,
                21200 ' Execute instructions
11990 REM ** 'SAVE' current program
12000 GOSUB 12500 ' Get the name
12010 COLUMN=1
12020 ROW=15
12030 GOSUB 4000 ' Clear message line
12040 PRINT"Saving data";
12050 PRINT£-1,"Easy",T$ ' Program & file name
12060 FOR I=0 TO 9
12070 PRINT £-1,"Easy",M(I),M(I+10),M(I+20),
              M(I+30),M(I+40),M(I+50),M(I+60),
              M(I+70),M(I+80),M(I+90)
12080 NEXT I
12090 GOSUB 4000 ' Clear message, all done
12100 GOTO 10000 ' Back to Command
12490 REM ** Read filename into T$
12500 ROW=14
12510 COLUMN=1
12520 GOSUB 4000
12530 PRINT"Prepare tape & enter name";
12535 T$=""
12540 INPUT T$
12550 RETURN
12990 REM ** 'LOAD' memory from tape
13000 GOSUB 12500
13010 COLUMN=1
13020 ROW=15
13030 GOSUB 4000 ' Clear message line
13040 PRINT "Searching tape";
13050 INPUT £-1,T1$,T2$ ' Read names
13060 IF T1$="Easy" THEN 13100
13070 GOSUB 4000 ' Prepare for message
13080 PRINT "* This is not an EASYCODE file";
13090 GOTO 10000 ' Try another command
13100 IF T$="" THEN 13120 ' Get next file
13110 IF T$<>T2$ THEN 13030 ' Keep looking
13120 GOSUB 4000 ' New message coming
13130 PRINT"Loading data record";
13135 COLUMN=20 ' Cursor after text
13140 FOR I=0 TO 9
13145 GOSUB 6000 ' Re-position cursor
13150 PRINT I+1;
13160 INPUT £-1,T$,M(I),M(I+10),M(I+20),
            M(I+30),M(I+40),M(I+50),M(I+60),
            M(I+70),M(I+80),M(I+90)
13170 IF T$<>"Easy" THEN I=20
13180 NEXT I
13185 COLUMN=1
13190 GOSUB 4000 ' Clear message
13200 IF I<19 THEN 13230 ' No error
13210 PRINT"* Loading error";
13220 GOTO 10000 ' Get another command
13230 GOSUB 9000 ' Re-draw memory-map
13240 GOTO 10000 ' Get command
13990 REM ** 'HELP' command received
14000 CLS ' Clear the screen
14010 PRINT"Valid EASYCODE commands are:
14020 PRINT
14030 PRINT"RUN to execute a MON+ program"
14040 PRINT"SAVE to store one on tape"
14050 PRINT"LOAD to read one from tape"
14060 PRINT"HELP to view this message"
14070 PRINT"QUIT to return to Basic"
14080 PRINT"CLEAR to reset MON+ memory"
14090 PRINT"STORE to enter data or program"
14120 PRINT"Please press <CR> when ready";
14130 INPUT T$
14140 GOSUB 9000 ' Redraw display
14150 GOTO 10000 ' Get next command
14990 REM ** 'QUIT' routine (nice and simple!)
15000 CLS ' Clear screen
15010 END ' That's all folks
15990 REM ** 'STORE' data or program
16000 COLUMN=1
16005 ROW=15
16010 GOSUB 4000 ' Clear messages (for later)
16015 ROW=14
16020 GOSUB 4000 ' Clear prompt line
16030 PRINT "Enter address (100 to stop)";
16040 GOSUB 4500 ' Get number
16050 IF N>99 THEN 10000 ' Error
16060 K=N
16070 ROW=15
16075 COLUMN=1
16080 GOSUB 6000 ' Set up next prompt
16090 PRINT "Enter data (100 to stop)";
16100 ROW=14
16110 GOSUB 4000 ' Set up varying prompt
16120 PRINT"Address";K;"=";
16130 GOSUB 4500 ' Get number
16140 IF N>99 THEN 16000 ' Error
16145 I=K
16150 GOSUB 16500 ' Store in memory & display
16160 K=K+1 ' Select next location
16170 IF K<100 THEN 16070 ' Get more
16180 ROW=15
16185 COLUMN=1
16190 GOSUB 4000 ' Clear old message
16200 PRINT "* End of memory reached";
16210 GOTO 10000 ' Get new command
16490 REM ** Put value N in M() and on screen
16500 M(I)=N
16510 ROW=INT(I/10)+1 ' F.P Basic only
16520 COLUMN=(I-10*ROW)*3+34
16530 GOSUB 8000 ' Print the number
16540 RETURN
19990 REM ** LOAD Register;number
20000 R(K)=J
20010 GOTO 5000 ' Set flags & update display
20090 REM ** LOAD Register;memory
20100 R(K)=M(J)
20110 GOTO 5000
20190 REM ** STORE Register;memory
20200 I=J
20210 N=R(K)
20220 GOSUB 16500 ' Display alteration
20230 GOTO 11500 ' No flags - just get next
20290 REM ** LOAD Register;Register'
20300 R(K)=R(1-K)
20310 P=P-1 ' Only a 1 char. instruction
20320 GOTO 5000
20390 REM ** ADD Register;number
20400 R(K)=R(K)+J
20410 GOTO 5000
20490 REM ** SUB Register;number
20500 R(K)=R(K)-J
20510 GOTO 5000
20590 REM ** SUB A;@X
20600 R(0)=R(0)-M(R(1))
20610 P=P-1 ' Only a 1 char. instruction
20620 GOTO 5000
20690 REM ** JUMPNC;address
20700 IF CARRY=0 THEN P=J
20710 GOTO 11500
20790 REM ** JUMPNZ;address
20800 IF ZERO=0 THEN P=J
20810 GOTO 11500
20890 REM ** JUMP;address
20900 P=J
20910 GOTO 11500
20990 REM ** LOAD A;@X
21000 R(0)=M(R(1))
21010 P=P-1 ' 1 char. instruction
21020 GOTO 5000
21090 REM ** STORE A;@X
21100 N=R(0)
21110 I=R(1)
21120 P=P-1 ' 1 char. instruction
21130 GOSUB 16500 ' Store & display
21140 GOTO 11500
21190 REM ** ADD A;@X
21200 R(0)=R(0)+M(R(1))
21210 P=P-1
21220 GOTO 5000
```

BBC NEWS FROM DPL

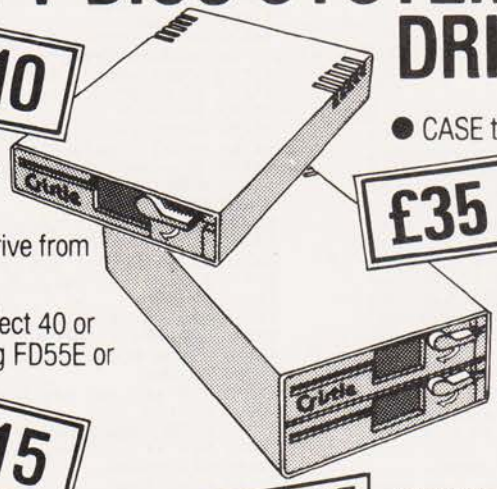
BBC COMPATIBLE FLOPPY DISC SYSTEMS AND DRIVES

£10

- Case to hold one drive
- Ribbon cable to connect one drive to BBC Micro
- Power cable to power one drive from BBC Micro
- Switch on rear of case to select 40 or 80 TPI operation when using FD55E or FD55F.

£15

HANDBOOK AND FORMATTING DISC



£35

- CASE to hold two drives (complete with blanking panel for use should only one drive be installed).
- Ribbon cable to connect **two** drives to B.B.C. Micro.
- Integral power supply for **two** drives.
- Switch on rear of case to select 40 or 80 TPI operation when using FD55E or FD55F.

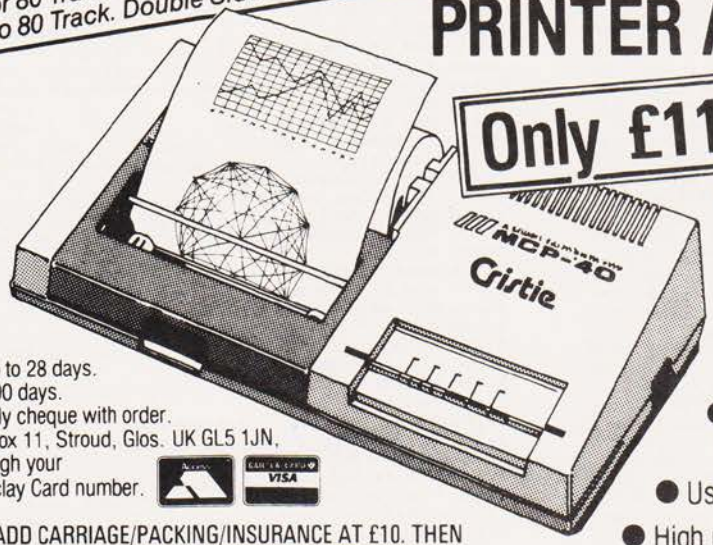
TEAC FD55 SERIES 5.25" 1/2 HEIGHT SLIM-LINE DISC DRIVES.

	Capacity 40TPI	80TPI	
FD55A 40 Track. Single Sided.	100K	N/A	£130
FD55B 40 Track. Double Sided.	200K	N/A	£180
FD55E 40 or 80 Track. Single Sided.	100K	200K	£155
FD55F 40 to 80 Track. Double Sided.	200K	400K	£218

THIS CASE CONTAINS ALL HARDWARE NECESSARY FOR TWO DRIVES. JUST SELECT ONE OR TWO DRIVES FROM THOSE SHOWN BELOW. IF YOU SELECT JUST ONE NOW, THE SECOND CAN JUST PLUG IN LATER.
CASES ONLY SUPPLIED WITH DRIVES.

PRINTER AND PLOTTER —MCP 40

Only £110



- 4 colours
- Selectable 40 or 80 characters of print per line, ideal for plotting pie charts, bar graphs, biorhythms, etc
- Easy to use—simple software commands
- Standard Centronics interface—cables available for most micros
- Uses 4½" plain paper
- High resolution—over 100 steps/inch

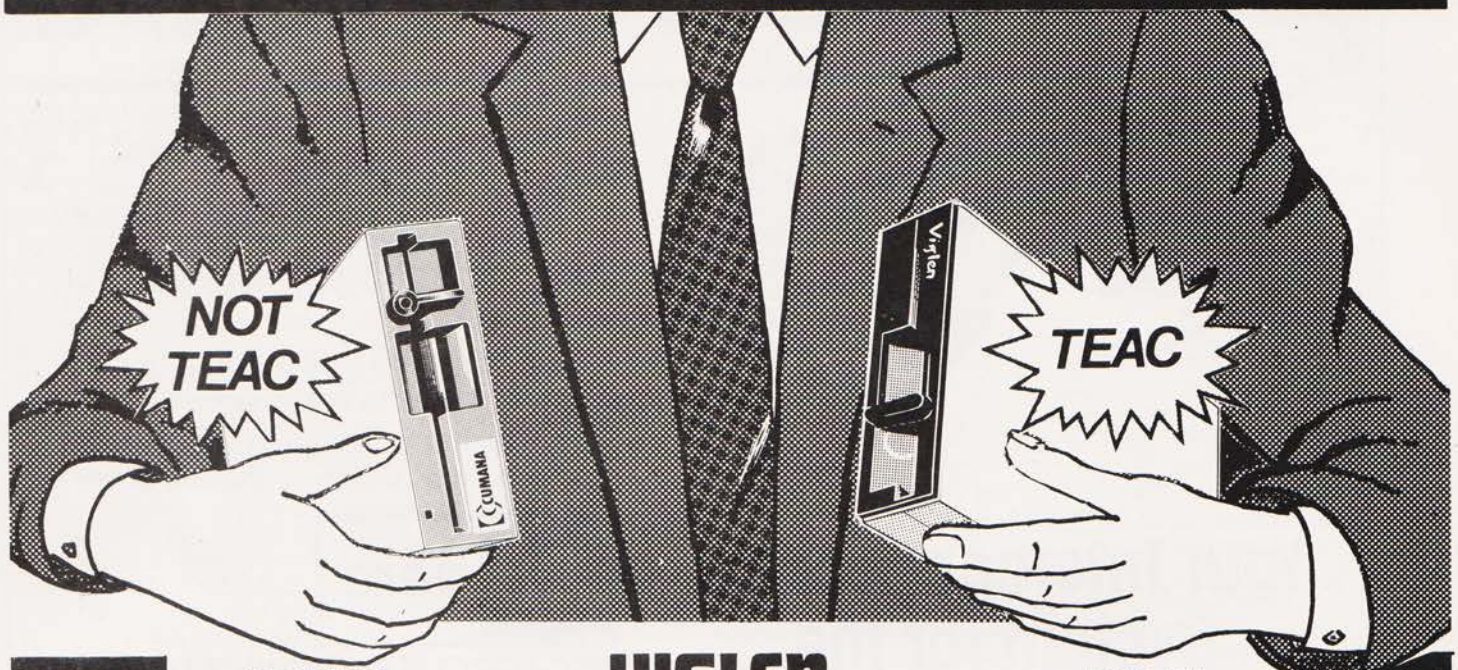
DELIVERY: Up to 28 days.
WARRANTY: 90 days.
TERMS: Strictly cheque with order.
Send to P.O. Box 11, Stroud, Glos. UK GL5 1JN,
or phone through your
Access or Barclay Card number.

TO ORDER: ADD CARRIAGE/PACKING/INSURANCE AT £10. THEN V.A.T. AT 15% TO TOTAL. (£10 COVERS NEXT DAY SECURICOR DELIVERY.) THE ABOVE PRICES ARE VALID UNTIL END OF MARCH 1984.

DIGITAL PERIPHERALS LTD.

Rodney House, Church Street, Stroud, Glos. U.K. GL5 1JN Tel: (04536) 71387. Telex 43551. A Cristie Company

BY FAIR COMPARISON



CUMANA
Dual Disk Drive
CD50F without DFS
£221.67
(As quoted in BERKSHIRE COUNTY GOVERNMENT SUPPLIES CATLOGUE)

VIGLEN DISK DRIVES

MEAN HIGH QUALITY UNBEATABLE MONEY VALUE!

VIGLEN
Dual Disk Drive
£170.00

JUST COMPARE . . . Whether you are an Educational Advisor, School, College, University, A Government Department, Company or Home User seeking high quality at value for money, VIGLEN DISK DRIVES will give you the quality and reliability that would confidently safeguard your decision to buy a VIGLEN PRODUCT!

"Page reproduction from catalogue"

MITSUBISHI DRIVES	Excl.VAT	Incl.VAT
400K SINGLE DRIVE	£213.04	£245
800K DUAL	£379	£435.85
800K & POWER SUPPLY	£399	£458.85

SPECIAL OFFER!
FULL HEIGHT DRIVE
£145 inc VAT
READY TO PLUG IN

449016 | Alternative Dual Disc Drive (Cumana CD50F) without DFS | Each | 548 85

Compare our prices!

TEAC 55

SINGLE DRIVES CASED	Excl.VAT	Incl.VAT
40 TRACK 100K	£144.35	£166
40 TRACK 200K	£200	£230
40/80 TRACK SWITCHABLE 200K	£173.05	£199
40/80 TRACK SWITCHABLE 400K	£223.48	£257
DUAL DRIVES CASED		
40 TRACK 200K	£278.26	£320
40/80 TRACK SWITCHABLE 400K	£349.57	£402
40/80 TRACK SWITCHABLE 800K	£452.18	£520
DFS KIT	£ 86.20	£ 99
P.S.U.	£ 28	£ 32

ENQUIRE ABOUT OUR 48-HOUR GUARANTEED DELIVERY SERVICE!

*These drives are fully compatable with other Computers and can be used should you change your computer.

HOW TO ORDER

BY POST
To purchase any of the items simply fill in your requirements listed above, in the coupon Enclose your cheque/P.O or use your Access or Barclaycard Please make cheque payable to; VIGLEN COMPUTER SUPPLIES and post to address below, allow 7 to 14 days for delivery & add £ 8–00 postage, package & insurance on all prices.

BY TELEPHONE -(01) 843 9903
Credit Card holders (Access/Barclay Card only) can purchase by telephone. Please give Card No, Name, Address and the item required.

Post to: VIGLEN COMPUTER SUPPLIES UNIT 7 TRUMPERS WAY · HANWELL · W7 2QA

Please send me the following items

I enclose Cheque/P.O. for £ _____

I prefer to pay Access/Barclaycard
(Delete whichever not applicable)

CARD No _____
SIGNATURE _____
NAME _____
ADDRESS _____

Tel (Day) _____ Tel (Eve) _____

CREDIT CARDS VALID TO BE SIGNED BY CARD HOLDER. ADDRESS AND NAME MUST BE THE SAME AS CARD HOLDER'S

CT3/84

Viglen
COMPUTER SUPPLIES
UNIT 7 TRUMPERS WAY
HANWELL W7 2QA
Tel: (01) 843 9903

EMS

Garry Marshall

BOOK PAGE

Four of this month's books belong to a series called 'The clear and simple home computer series'. The series is published by Windward which is an imprint of W.H. Smith. In view of Smith's stated policy of reducing the number of computer magazines and books stocked in their shops by retaining only the better ones, it is natural to expect that any books published under their auspices should be rather special. It could be embarassing if they refused to stock their own books!

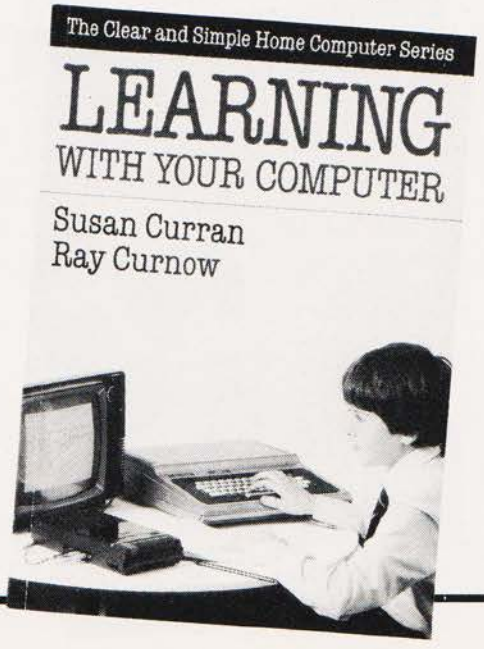
The series shows every sign of very careful planning. There is a series consultant, Richard Pawson, who is the editor of a monthly computer magazine, and is now editor of the **Home Computer Course** part work, so that his credentials speak for themselves. The acknowledgements at the back of each book give what is essentially a masthead, crediting the series editor, art editor, editor, designer and so on. In fact, the series seems to have been planned more along the lines of a part work than as a conventional series of books.

The titles themselves are quite carefully chosen. I suppose that any series intent on providing an introduction to home computing and home computers must have its **Introduction to Computing** and **First Steps In BASIC** or their equivalents. But since these books are inevitably similar to so many others written before them, there is an extra reason why they had better be good. A book called **Learning With Your Computer** seems to me an excellent idea. The topic is long overdue for treatment at an introductory level, and is something that many people would like to know about. **Games, Graphics and Sounds** is also a sure-fire success, given that most micros are used for games and that graphics and sound are the two most attractive features of micros, whether in games programs or programs of any other kind.

After all this preparatory work, let us see if the authors have delivered their contributions to the standard of the planning and the production. The two more original titles can naturally be considered together. **Learning With Your Computer** and **Games, Graphics and Sounds** are both by Susan Curran and Ray Curnow. The authors remark that most people buying a home computer plan to use it for educational purposes, but that is not what they are actually used for: in fact, most computers are used for playing computer games. I think that this can be accepted without question, and it follows that the authors have written two very different books both of which should appeal to the majority of homes where a computer is to be found!

Both books have to resolve the same problem concerning programs and programming, in that most beginners are users of programs but would probably like to be able to write them. These books cannot be just catalogues of educational or games software. Nor can they reasonably be expected to take the beginner to the stage where he can write substantial educational or games programs, for a good deal of sophistication is needed in either case. The problem is resolved nicely in both books by surveying the **types** of programs that are available, presenting enough BASIC for the beginner to make a start at programming, and then presenting some listings of medium scale programs with the exhortation to try them. Besides dealing with this problem in the same way, both books have much the same format.

Learning With Your Computer begins with a review of the history of using the computer as an educational aid. Computer-aided instruction (CAI) and computer-aided learning (CAL) are examined and explained. In CAI the initiative is with the computer, which typically takes the user through a drill or provides information on a particular topic and then provides multiple choice questions at strategic points. The user can proceed if he gives the correct answer, while if a wrong answer is given, the information is represented in another way intended to remedy the lack of understanding as revealed by the incorrect answer. CAL provides the user with a learning environment that he can explore and in which he can test ideas. Thus, with CAL learning is an active process and the initiative is with the user. Logo is the prime example of a system for supporting CAL. Its 'microworlds' are learning environments. The development of effective techniques for use in CAI and CAL has been the subject of investigations for a surprisingly long time.



The book then looks at children and computers, examining what is available to help children learn with computers, and how they can learn with computers. It also makes the strong point that children cannot be too young to start using computers. It then explains how resource programs such as word processing programs and database systems can be of considerable value as educational aids. The rest of the book consists of two chapters on buying hardware and software that are too brief to be very helpful and the final chapter which contains listings, explanations and remarks for six educational programs. The programs are written in BASIC and range from 50 to 100 lines in length. They are written for the Dragon and can, as the authors suggest, be readily adapted for other computers. (I think the graphics might be quite hard to adapt though.)

I thought that the book started very well, and that the final chapter was of value with its program listings and the very clear flowcharts for them, but the central chapters are very slight. However, the book should prove invaluable to any owner of a home computer wanting to use it for educational purposes, but not knowing how to go about it.

Almost all of the remarks on the last book apply equally to **Games, Graphics and Sounds.** It starts with a short history of computer games that is very interesting, and stresses the role of professional programmers in creating the first computer games as a relaxation and contrast to their other programming duties. Computer graphics and sound generation are both given a brief treatment, but one that is adequate at the introductory level. Then two slight chapters on computer hardware and writing games precede the large chapter of program listings and flowcharts. The programs include Hangman, Breakout, an artist's drawing program, and

a program that enables the computer to be used like an organ by playing tunes on its keys. The programs are clearly explained and fit the aims of the book very well. Despite sagging in the middle, this is a good introductory book on games, graphics and sound.

I should perhaps make the point that both books are written for the beginner and succeed at that level, but a reader who is reasonably au fait with computers and their uses might find, after their interesting introductory chapters, very little that is new.

First Steps In BASIC is also by Susan Curran and Ray Curnow. It has a breezy approach to its subject, is gently paced and carefully structured. It continually encourages the reader to try out his ideas on a computer and to learn by doing as well as by reading. It takes a fairly conventional order in its treatment of BASIC, but there is nothing wrong with that, for it ensures that nothing important is omitted. It is the informal and encouraging style that gives the book its attraction. The last two chapters are on writing longer programs and the steps to be taken to proceed further in BASIC than the book takes you. Thus, the reader is not only given an introduction to BASIC but also receives an encouraging push towards the next phase. I found the book very attractive and can warmly recommend it as an introduction to BASIC.

I have saved **Introduction to Computing** by Peter Lafferty until last because, in contrast to the previous three books, I found it very disappointing. I have explained why I think that another book with this, or a similar, title must be very good to justify its existence. This is not a very good book. In fact, it is not even adequate. It cannot hold a candle to **The BBC Computer Book**. There is nothing new in the book, the presentation is not sufficiently lively to compensate, there is a certain amount that is misleading and, inexcusably, even some that is wrong. Did you know that '. . . information is stored as a pattern of low and high voltages, known as an electrical pulse'? (Page 23.) Or that 'The accumulator can store a number and, if given a second number, will add the two and store the result. It is very useful to the ALU'? (Page 102.)

The book starts with an explanation of what a home computer is, covers what it is used for and how it works, and moves on to the writing of programs, choosing a computer and the ways of expanding it. The (admittedly difficult) task of writing a general book about personal computers of which the many types are all different, is never satisfactorily resolved. A general treatment of some aspect frequently ends with the remark that the manual will give the details relevant to a particular computer. Although this is true, who wants to read a book that keeps telling you to read the manual? (Anyway, if the manuals were any good we might not need the book.)

The final chapter entitled 'Into The Future' claims to give us a glimpse of the future but satisfies itself with an account of some of the more advanced current developments. This may be a blessing in disguise.

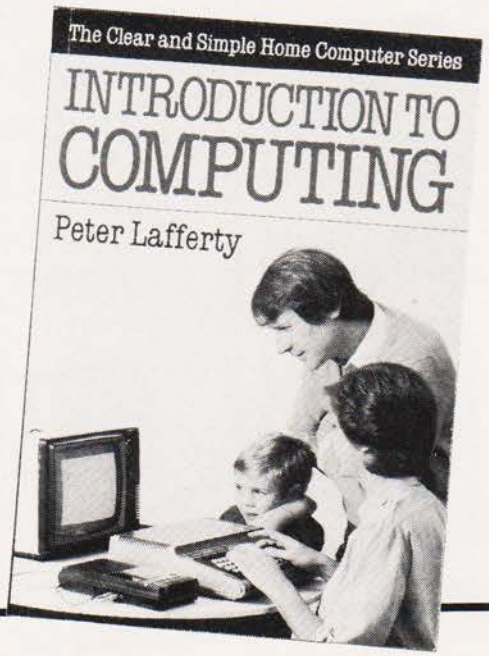
If I have laboured my point, it isn't with the aim of being unkind to the author. I am aware of the pressure that authors are under in writing any book, but particularly with short deadlines hanging over their heads.

I do think, though, that a lot of preparatory work can be wasted, and a series not be as good as it should, if manuscripts are not carefully vetted prior to publication. There is no need to offer the reading public an inferior book on the subject of this one.

As far as the series is concerned as a whole, its format is very attractive. There is a certain amount of overlap between different books, but I think that this is inevitable. In general, I found the illustrations a little disappointing (with the honourable exception of the flowcharts). Too often they repeated the text rather than complementing it. Also, the two-colour reproduction gives them a rather 'muddy' appearance. I realise that the use of multi-coloured diagrams would raise the price of the books, but to see what is possible it is only necessary to turn to the **Home Computer Course.** Their illustrations have an impact, an attractiveness and a role in expanding the text that these do not.

In complete contrast, I recommend that you read **Communications — Getting It All Together** by Sir George Jefferson. Don't buy it though, try to find it in a library. The author is chairman of British Telecom, and this pamphlet gives the text of a lecture to the British Computer Society. This may not sound very exciting, but it actually makes very good and informative reading. The author's theme is the convergence of telecommunications and computing to give Information Technology. His coverage is clear and authoritative, and his style is simple and direct, so much so that one can almost hear him speaking as one reads. He points out that many of the fundamental concepts of computer operation, such as registers, stored programs and machine code were

embodied in telephone exchanges as long as 75 years ago. His coverage of current and future developments explains how computers help to improve satellite communication services, and that the introduction of artificial intelligence can vastly improve the usefulness of telecommunications systems. If a computer is used to switch an international call from one country to another, there is no reason why it should not translate the conversation from one language to another at the same time (in principle, at least!). A short and stimulating read.



This month's books are:

Learning With Your Computer by Susan Curran and Ray Curnow, Windward, 144 pages, £4.95.

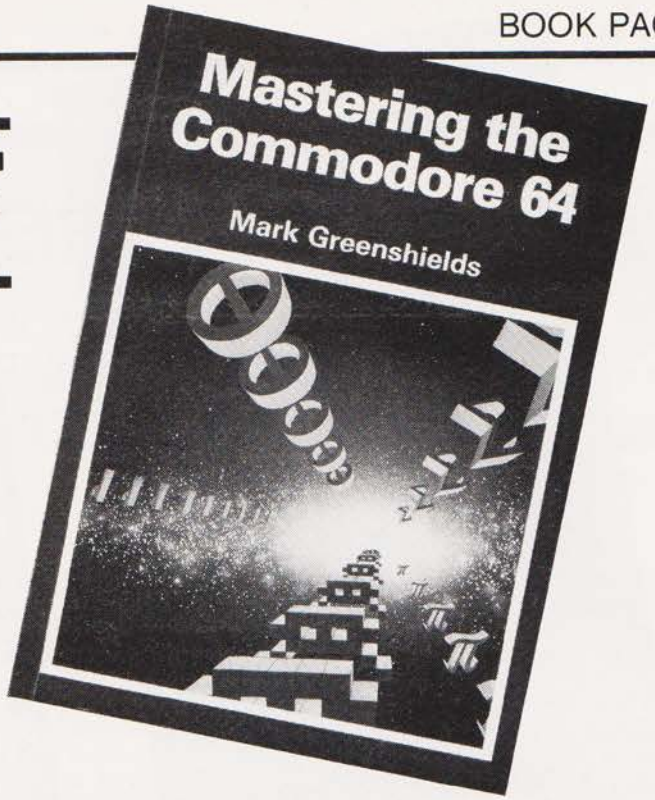
Games, Graphics and Sounds by Susan Curran and Ray Curnow, Windward, 144 pages, £4.95.

First Steps In BASIC by Susan Curran and Ray Curnow, Windward, 192 pages £4.95.

Introduction to Computing by Peter Lafferty, Windward, 188 pages £4.95.

Communications — Getting It All Together by Sir George Jefferson, Wiley Heyden, 14 pages, £5.

IN BRIEF



COMMODORE 64 MACHINE CODE MASTER

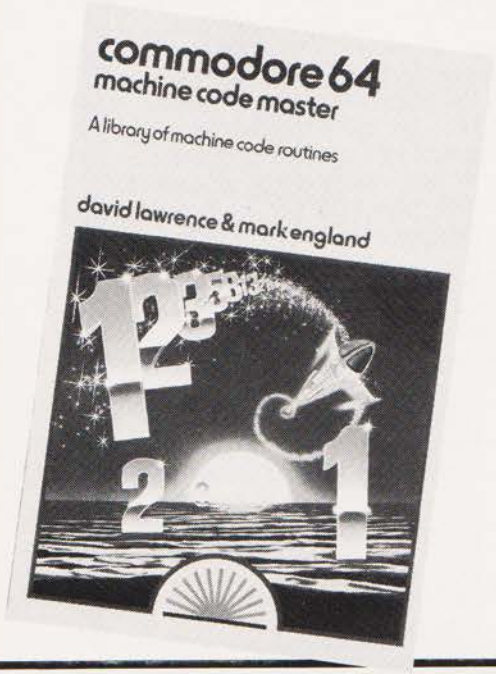
by David Lawrence and Mark England, Sunshine Books
Price: £6.95

Many books that refer to machine code in the title can be quite a let-down to the programmer who has passed beyond the novice stage. You get books full of addresses which you strive to understand, tacked onto crude loader program that leaves the aspiring learner completely bewildered. Not so with this book — it is easy to read and states at the outset that its aim is **not** to teach machine code! What is sets out to do is give a really worthwhile machine code programming tool. This contains a Monitor to allow examination and change of memory contents, a Disassembler which translates machine code programs into assembly language format, and a file editor and assembler which allow assembly language programs to be developed and translated into machine code. All this is written in fully documented BASIC, in a modular form that easily permits step-by-step checking of what you have entered (although some modules require the entire program to be keyed in before they will function as intended).

Although this book is not a machine code primer, the routines provided and the presentation are such that the average reader, with some slight persistence, will inevitably find his or her general knowledge and understanding of machine code is greatly enhanced. The assembler you eventually create is a full two-pass type and has complete labelling and error-checking facilities, and although it is a little slow it should not be easily faulted.

In addition to this 'Mastercode' tool, there are a number of machine code routines that extend the Commodore 64's BASIC. Some of these are UNDEAD (also known as OLD or UNNEW; the retrieval of a program that has been NEWed); PLOT (print at any point on the screen); DELETE (block line deletion); and RESTORE (to a specified line number). These and several other routines merely whet the appetite for the things that may be possible when you read this book, which is definitely to be recommended.

P.F.



MASTERING THE COMMODORE 64

by Mark Greenshields
Interface Publications

A slightly frustrating book that is rather like many highly publicised television programs — at the end of them you wish they had lasted another hour or two! (You must be picking up different channels to my TV set — Ed). Perhaps this is being a little unfair to a quite useful book, but it does try to cover rather a lot of ground, and sometimes you are left thinking that you would have liked an extra page or three of explanation. There are chapters on the 64's BASIC language, colour, animation, music and sound synthesis, programmable characters, sprites, high resolution graphics, comparisons of the BASIC to that of four other machines, peripherals, and speeding up and improving your BASIC programs — and all in 94 pages.

The second section of 60 pages is an introduction to machine code, including a simple no-frills hexadecimal loader program that enables you to enter the various examples given in this section. Colour, animation, sound and music, programmable characters and sprites are discussed, together with a brief introduction to the use of interrupts. Some programs to try, and a reasonably comprehensive set of appendices complete the 219 pages.

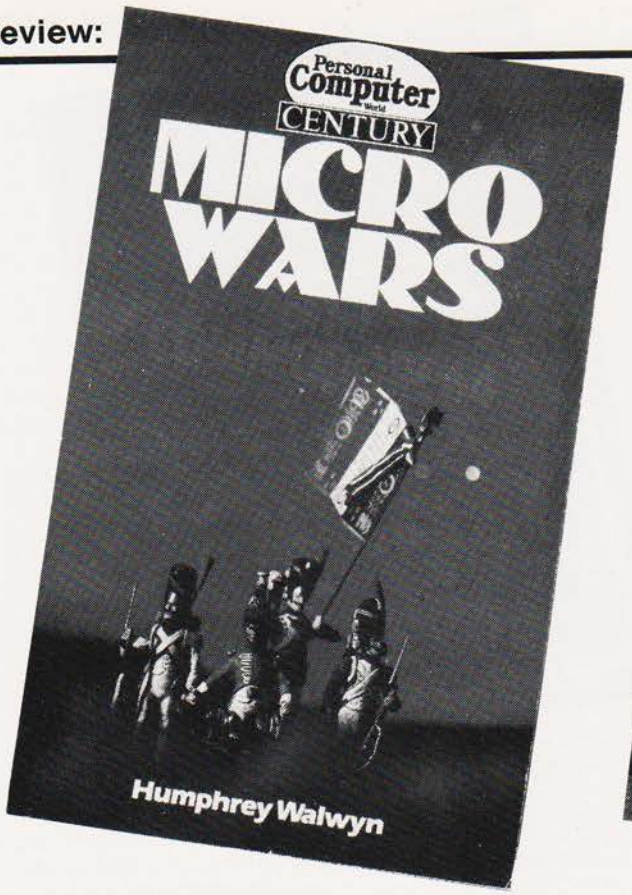
Although each topic (in both sections) is covered at fairly breakneck speed, this book could prove useful to new users of the Commodore 64 who have a little knowledge of computing, are trying to get that little bit more from their machine and who need a few hints to stimulate their thinking processes. Not a book that can be recommended highly, but well worth consideration.

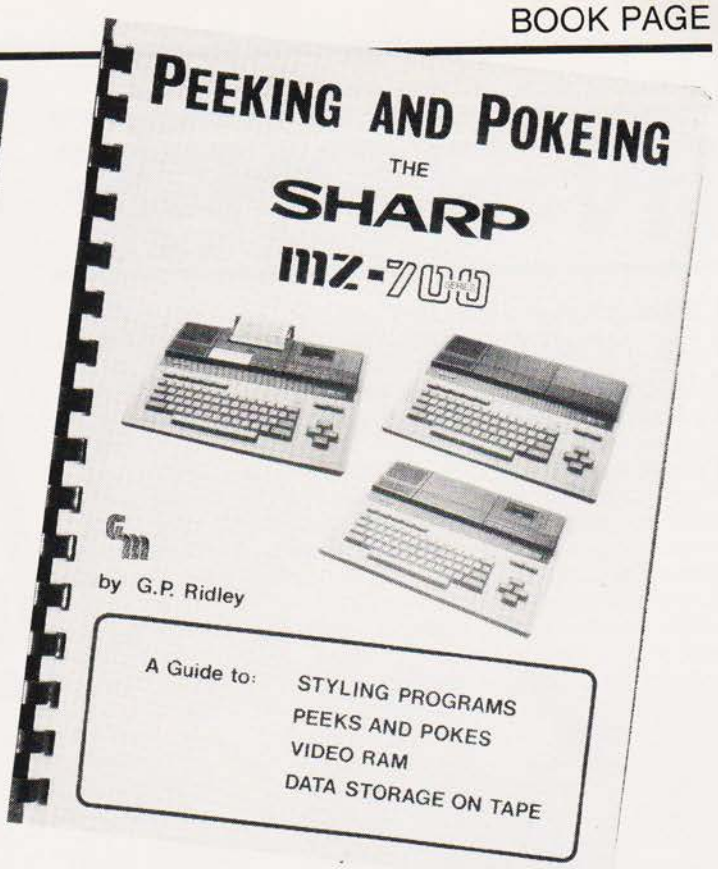
P.F.

MICRO WARS ON THE COMMODORE 64

by Humphrey Walwyn
Century Publishing, 76 Old Compton Street, London W1V 5PA
Price: £5.95

This is a book for wargamers with computers. It presents a series of wargames on various subjects, ranging from a battle in ancient times to a World War II air-versus-sea tactical battle. Each game is well-presented: there are historical notes, general notes on the game, a breakdown of





the listing and a long example of play as well as the listing itself. Although the programs have been written for the Commodore 64, they are easily adaptable to any micro and notes are provided to aid such conversions.

There are six wargames in the book, the first being a re-fight of the Battle of Waterloo for two players. The second is an ancient battle for two players, showing the confusion that armies of those days often got into. The third ia rather trivial game for one player against the computer, illustrating an airborne torpedo attack against a ship. In the fourth game, two players design and build a World War I plane, then try out their designs against each other. The fifth game is based on the same idea, but in this case involves World War I navies. Finally, the sixth game is a general wargame for two players, which can be adapted to many scenarios.

Considered as wargames, all of these games are excellent, but as computer games they are clumsy to play. A computer can provide an opponent for a lone wargamer: most of these games require two players. Also, in some of the games a player is requested not to look at the screen during the other person's turn, which means you get to play musical chairs as well as the wargame.

I can recommend this book as an introduction to computer wargaming, but if you wish to go further and write your own wargames you will need to supplement it with something about generating computer opponents.

M.E.

PEEKing And POKEing the Sharp MZ-700

by G.P. Ridley
D.C. Brennan Engineering, 14 North Western Avenue, Watford, Herts
Price: £6.95

The beauty of the Sharp range of computers is that the BASIC operating system is contained, not in ROM, but is loaded into the RAM after you've switched on. The main idea behind this is to allow the use of alternative languages without wasting redundant address space, but it also lets the keen programmer get in and mess around with BASIC. You can customise your operating system, add new commands or just make better use of the old ones. Of course you need to find your way around the code first and in my experience, Sharp (UK) are not equipped to deal with this sort of enquiry.

As owners of the previous Sharp models in the MZ range will know, there are a couple of useful books available that give you all the information you really need in this area. These are **PEEKing and POKEing the Sharp MZ-80K** and — surprise — ditto **the Sharp MZ-80A.** The launch of the MZ-700 last year has resulted in this, the third book in the series, and it is as useful as its predecessors.

The book begins with an explanation of the Sharp's features and memory organisation written for the beginner, followed by a brief run-through of some of the keywords, the tone generator and the unusually-organised video RAM. There is also a good explanation of the second character set which is only available on the British version of the computer and is not mentioned in the User's Manual. Then we get to the real meat, which is improving the BASIC and making a back-up copy of the revised language on tape.

Among the useful routines are a TRACE command that doesn't print line numbers all over the screen, variable speed RUNning of programs, bell on READY, toggle screen LISTing and stepping through the listing, data tape prompt messages and so on. The following chapter gives a selection of PEEKs and POKEs, plus some USR calls, which greatly increase the versatility of any BASIC programs you write.

Further tips include a rather more detailed explanation of the use of the built-in four-colour plotter, how to recover 'lost' programs by resetting the program pointers, program protection, and a demo program for data handling. For the machine code programmer there's a selection of BASIC interpreter points and useful addresses and RAM monitor subroutines.

The review copy of this book was an early version and contains a few silly spelling mistakes and errors (which are pretty obvious), though the publisher tells me these have been corrected in the reprint which is currently on sale. To both the BASIC programmer and the machine code fans this is an excellent handbook, and can be thoroughly recommended.

P.N.G.

Software News

INNOVATIVE
TRS 80·GENIE SOFTWARE
from the professionals

NEWS FLASH!

Molimerx and Logical Systems of Milwaukee, U.S.A., have joined forces to bring their customers a lower costing product and faster and more efficient service.

From January 1984 all LDOS 5.1.× Logical Systems products, together with some of the LDOS (TRSDOS) 6.× products, will be available from Molimerx at the pound equivalent of the U.S. Dollar retail price. In other words, for the first time the considerable range of products of Logical Systems Incorporated will be available to the end user in the United Kingdom at the price at which the American customer can buy it in the U.S.A. All support for these products is being shifted to England, so that as from 1st January, U.K. customers can have the benefit of this important line, exactly as if it had been written and produced over here.

Adjustments for the exchange rate will be made every six months or so. We are starting with the present exchange rate of 1.48. After VAT is added this scheme results in the price schedule (plus P&P) that follows:

Name	Brief Description	Previous Selling Price	Present Selling Price
DISK/DISK	Convert a disk file to another "disk drive"!	£n/a	**£ 76.94**
FED II 5.1	All purpose disk file editor	£ 27.60	**£ 27.60**
FILE MANAGER 5.1	Utility for mass manipulation of files	£ 33.40	**£ 30.30**
FILE MANAGER 6.0	Utility for mass manipulation of files	£ 33.40	**£ 30.30**
FILTER PACKAGE 1	Filters to enhance LDOS	£ 22.71	**£ 22.54**
FILTER PACKAGE 2	Filters to enhance LDOS	£ 22.71	**£ 22.54**
FIX DISK	A collection of patches for LDOS	£ 13.80	**£ 7.76**
HELP 5.1	LDOS and LBASIC help	£ 17.25	**£ 14.78**
HELP GENERATOR 5.1	Create your own HELP files	£ 33.35	**£ 33.35**
HELP TEXT SOURCE	Source files for creating main HELP files	£ 17.25	**£ 14.78**
INVENTORY	An aid to inventory tracking	£ 82.50	**£ 76.94**
I/O MONITOR	Disk I/O error intercept utility for LDOS	£ 22.43	**£ 14.78**
LDOS 5.1.×	New generation disk operating system	£105.80	**£100.28**
LDOS TECH. HELP	Technical help for LDOS	£ 20.70	**£ 20.70**
LED	Screen orientated text editor	£ 21.85	**£ 21.85**
MAIL/FILE II	A mailing list database manager	£ 82.50	**£ 76.94**
MEMDISK	Additional disk type storage	£ 28.69	**£ 22.54**
QUIZ MASTER	Questions and Answers — Master includes general	£ 33.40	**£ 30.30**
QM GEOGRAPHY	Questions and Answers — Geography Requires Quiz Master	£ 17.25	**£ 14.78**
QM MATH	Questions and Answers — Maths Requires Quiz Master	£ 17.25	**£ 14.78**
Smal-LDOS	Miniature of the original LDOS	£ 43.70	**£ 43.70**
T.B.A. 5.1	Basic text processing utility	£ 51.75	**£ 51.75**
T.B.A. 6.0	Basic text processing utility	£ 57.50	**£ 57.50**
ULTRA TREK	Space Wargame	£n/a	**£ 11.87**
UTILITY DISK I	LDOS enhancement package	£ 33.46	**£ 30.30**

TEL: [0424] 220391/223636

MOLIMERX™ LTD

TELEX 86736 SOTEX G

A J HARDING (MOLIMERX)

1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

GRAFPAD

...for as many uses as YOU can imagine!

BBC MODEL 2 • SPECTRUM COMMODORE 64

With Grafpad you can now add a new dimension to your computer enjoyment, but most important, it helps you create your own application programmes by the simple use of the Grafpad!

The Grafpad comes complete with a cassette comprising two programmes.

THE FIRST PROGRAMME

displays the co-ordinates of your screen area. The co-ordinates are based on the screen with a grid size of 1023 x 1279 pixel, also in the Grafpad giving you a grid size of 320 x 256 pixels!

THE SECOND PROGRAMME

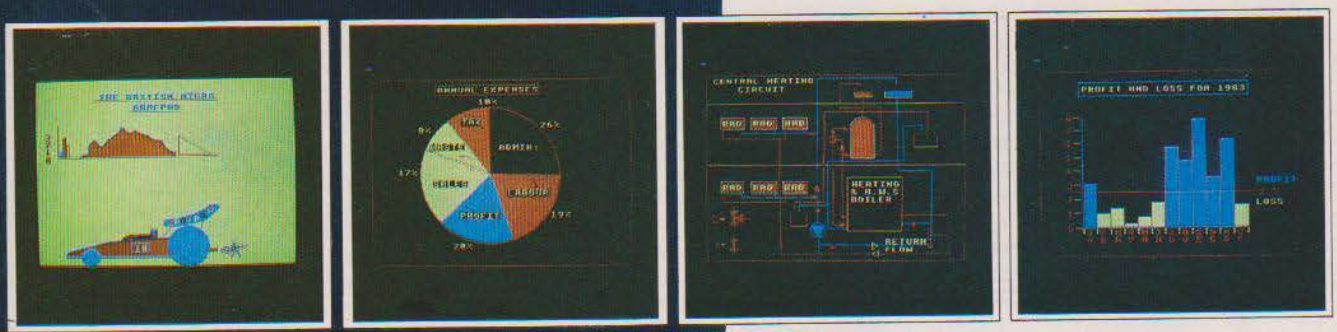
provides you with the utilities for circles, squares, triangles, free-hand, erasing line-drawing etc, and of course, full "Fill-in" facility in 16 different colours by the simple use of the pen!

Draw from a simple apple to a computer circuit - store in cassette or disk, perhaps transfer direct to a printer - in black and white or full glorious colour!

Purchase unique C.A.D. (Computer Aided Designs) programme and add further enjoyment and professionalism to your computer designs!

The Grafpad comes complete with Operational Manual, Programmes, The Grafpad and Pen and it simply plugs in your computer.

(Size: 25mm height x 55mm width x 260mm depth)
Weight: 1.2 kg (Gross)



(The above designs were drawn by a 12-year-old at our showrooms!)

HOW TO ORDER:

BY TELEPHONE:
If you are an American Express, Barclaycard, Diners Club or Access Card Holder simply telephone us giving your Card No., Name, Address and item(s) required and your orders will be dispatched within 48 hours!

BY POST:
Simply fill in the coupon, enclosing your cheque/P.O. made payable to: BRITISH MICRO, or use the special section for Credit Card Holders, and post to the address below.
Please allow 14 days for delivery.

Post to: BRITISH MICRO, UNIT Q2, PENFOLD WORKS
IMPERIAL WAY, WATFORD, HERTS. WD2 4YY

Please send me Grafpad for:

☐ BBC MODEL 2 ☐ SINCLAIR SPECTRUM ☐ COMMODORE 64
(Please tick)

Qty.	Item	Ex. VAT	Inc. VAT	Total
	Grafpad Complete	£125.00	£143.75	
	C.A.D. Programme	£18.00	£20.70	
		Postage, Packing & Insurance		£5.00
			TOTAL	£

I enclose my cheque/P.O. for £.....

I prefer to pay with my American Express, Barclaycard, Diners, Access Card
(Please cross out whichever is not applicable)

CARD NO. _____

SIGNATURE _____

NAME _____

ADDRESS _____

Address above must be the same as card holder.

- **DEALER INQUIRIES WELCOMED**
- **SPECIAL DISCOUNTS FOR EDUCATIONAL AUTHORITIES**

BRITISH MICRO
A HEGOTRON GROUP COMPANY
Unit Q2, Penfold Works,
Imperial Way, Watford, HERTS WD2 4YY
TEL: (0923) 48222. TELEX: 946024

P. J. Kenworthy

BBC POKER

This is an excellent simulation of five-card poker, except for one thing — you'll never be able to read your opponent's expression. Runs on a BBC Micro with 32K.

This program has been written for a 32K Model B BBC Microcomputer with a 1.2 Operating System, though it should work on a 32K Model A or on any previous Operating System since no machine-code is used. The working of the program revolves around three sets of strings, together with the computer's ability to dissect these up for its own purposes.

The fifty-two cards in the pack are stored in the string array B$( ) in the following format:

1 of Clubs
2 of Clubs
:
10 of Clubs
11 of Clubs

and so on.
These are then shuffled (by use of random numbers), and dealt out alternately into the player's and computer's hands. These are represented by the string arrays P$( ) and C$( ). From then on, the program revolves around the data stored in these two arrays.

The program makes use of user-defined characters in order to provide the on-screen display of the cards. However, the fact that the program is over 12K long, together with the need for a fairly high density text format, meant colouring of the cards became impossible.

RUN WITH THE PACK

When the program is first run, after offering the user instructions, it asks whether you wish to play Wild Poker or not. (It should be noted that the user is already expected to know how to play Poker. This program is not a teaching guide!). In wild poker, deuces can represent any value of card, but flushes, straights and runs are not allowed.

After the user has made the decision as to play Wild Poker or not, his/her cards are then displayed. The program then asks whether the user wishes to fold (ie whether the hand is too poor to play). If the answer is "Y", the player's score has £30 deducted from it (the stake at the beginning of each hand), and the computer's score is incremented by the above said amount.

If the user decides to carry on, a prompt is then displayed asking how many cards the player would like to discard. A maximum of three is allowed. These cards are then removed from the screen and their replacements displayed. Once again the user is asked whether he/she would like to fold. If the reply to this is no, the computer will go into the next part of the program.

The computer then looks at hits hand and decides on how many cards to discard. Once this has been carried out, it checks its cards again and decides whether to fold or not. Assuming that it has not, it is time for the gambling to begin.

The computer chooses (on the throw of a hypothetical coin) who will bet first. If it is the player a prompt will be displayed asking for the amount of the bet to be typed in. The maximum stake is £1000, the minimum is £1. Bets below or above these amounts are not allowed. Once the player has typed in his/her bet, the computer will reply as to whether it is going to fold, see the player's bet or raise it. If it has raised the bet, the player will be offered three options. These options are also offered if the computer bet first. These options are:

- To fold
- To see the computer's bet
- To raise the computer's bet.

If the first option is chosen, all the money in the pot at the time is added to the computer's score, new cards are dealt and another hand is begin.

The second option subtracts the computer's last bet from the player's score, then displays both hands on the screen. The computer then decides who has won, adjusts the scores by the relevant amounts, deals new cards and starts another hand.

If the player chooses to raise the computer's bet, the computer's last bet, plus this further bet is taken away from the player's score.

Each of these options is carried out identically when it is the computer's turn.

Should, during the course of a hand, the player's score drop below zero, a loan of £1000 is offered to him/her. If this offer is taken up, the amount is credited to his/her score. The only stipulation attached to this generous offer is that the loan must be paid back in full after 10 hands without the score again dropping below zero. Otherwise the game is over and the computer has won. The player's score is allowed to drop below zero once the loan has been accepted, but it is normally quite difficult to win enough hands in order to be not in debt when the loan has to be repaid.

Should the computer go into debt it will automatically take out a loan, though the same rules apply as to the player.

The game will continue until either the player or the computer is unable to pay back a loan.

In normal play (ie not Wild Poker) the following list gives the order of merit of hands:

1) Royal Flush (A,K,Q,J,10 of the same suit).
2) Straight Flush (run of five cards of the same suit).
3) Four cards of same value.
4) Full House (one pair and three cards of the same value).
5) Flush (five cards of same suit).
6) Straight (run of five cards of any suit).
7) Three cards of same value.
8) Two Pairs.
9) One Pair.
10) Ace high downwards.

As I have stated previously, flushes and straights are not allowed in Wild Poker. It should also be remembered that in Wild Poker, four of a kind is not the highest hand possible. Five of a kind can and do exist!

After the first hand of any game has been played, the winner of that hand will bet first in the next hand — and so on.

CONVERSIONS

As I stated earlier, this program was written on a BBC Model B. To enable it to work on a 16K Model A, the program would have to be re-written so that it ran in Mode 7 all the time. This would mean the loss of any of the graphics present in this version.

BBC Poker should work on any other micro which has a similar sized memory and on-screen text format (40 by 32 in this case). I shall now go through any of the points which would need to be adjusted to enable the program to work on another micro.

The PROCedures used throughout the program would need to be replaced by GOSUB statements. The two reasons I used this particular function in my program were that, first, it would run faster, and second, the listing would be more easily understood by the reader.

CHR$(141) on line 90 creates double-height characters, while CHR$(136) in line 350 makes the following line of printed text flash on and off. The VDU statements between lines 170 and 210 create the user-defined characters employed by the program. The characters lie between character codes 224 and 228, and represent a club, diamond, heart, spade, and square block respectively. The VDU statement at lines 420 and 620 creates a text window which starts at line 13 on the screen . This means that when CLS is used, only the screen from this line downward is cleared. VDU 26 (lines 580 and 600) return the screen to normal.

RND(20), which chooses a number at random, will pick an integer between one and 20 inclusive. INT instructions may be necessary on other micros.

An apostrophe following a PRINT statement will cause a line to be missed out. Thus PRINT'" should leave a gap of three lines.

The three SOUND commands at lines 1880, 1910 and 3920 produce tones of different pitches which last for approximately half-a-second. REPEAT-UNTIL are merely versions of a FOR-NEXT loop, the difference being that in the former, the loop is REPEATed UNTIL a certain defined condition is met.

A statement such as TAB(6,10) is used to tell the computer to PRINT on the screen six columns across and 10 lines down. The command *FX15,1 in line 4300 has the effect of flushing out the keyboard input buffer prior to a GET. This may or may not be necessary on other computers — on the PET, for example, you could use FOR I=1 TO 10: GET A$: NEXT I to achieve the same effect.

The musical accompaniment to the screen titles has little or no effect on the game, so the whole of PROCMUSIC could be ignored provided the following adjustments are made:

- Line 110 should be replaced with a command to GET a character from the keyboard, ie A$=GET$.
- Line 120 should be deleted completely.
- The string variables in lines 240 and 260 should be identical to the string in line 110.

Apart from the commands listed above, the program should work on any computer with a fairly standard form of BASIC. As the program listing, and the explanation of it, are so long, we are publishing the complete listing this month and a detailed description of its operation in the April issue.

Listing 1. The BBC Poker program.

```
   10 ON ERROR GOTO 5250
   20 CLEAR
   30 *FX11,0
   40 V$=" "
   50 d=.03042:e=.3562:f=3.042
   60 d1=2.34E-3:e1=.0137:f1=.117
   70 LOAN=0:LOAN1=0:Z2$=""
   80 MODE 7
   90 FOR N = 1 TO 2: PRINT TAB(9);CHR$(129);CHR$(
141);"BBC POKER":NEXT
  100 PRINT''"Do you require instructions (y/n)?"
;
  110 REPEAT PROCMUSIC: UNTIL ( A$="Y" OR A$="N" )
  120 A1$=A$
  130 *FX15,1
  140 VDU23,1,0;0;0;0;:REM**CURSOR OFF**
  150 DIM B$(52),P$(5),C$(5),AA$(13),PP(5),H$(6),S
$(5),SCORE(3),PS(2)
  160 DIM ST(2),SS(2),PT(2),CC(2)
  170 VDU 23,224,8,28,28,107,127,107,8,28
  180 VDU 23,225,8,28,62,127,62,28,8,0
  190 VDU 23,226,54,127,127,127,62,28,8,0
  200 VDU 23,227,8,28,62,127,127,127,28,62
  210 VDU 23,228,255,255,255,255,255,255,255,255
  220 J1=1000:K1=1000
  230 PROCINITIALIZE
  240 IF A1$="Y" OR A1$="N" THEN 260
  250 GOTO 110
  260 IF A$="Y" THEN PROCINSTRUCTIONS
  270 CLS
  280 PRINT''CHR$(133);"  Do you wish to play Wild
 Poker"
  290 PRINT CHR$(133);"  ( i.e. 2'S count as any c
ard )"
  300 PRINT''TAB(14)"(Y/N)?";
  310 Z1$=GET$
  320 IF Z1$="Y" OR Z1$="N" THEN 340
  330 GOTO 310
  340 PRINT''CHR$(134);" Your cards will now be d
isplayed"
  350 PRINT TAB(4,24);CHR$(136);CHR$(130);"PRESS A
NY KEY TO CONTINUE";:A$=GET$
  360 VDU7
  370 MODE4
  380 VDU23,1,0;0;0;0;:REM**CURSOR OFF**
  390 PROCB
  400 IF LOAN>10 THEN PROCDECIDE
  410 PROCDISPLAY
  420 VDU 28,0,31,39,13
  430 PRINT "Do you wish to fold (Y/N)?"
  440 A$=GET$
  450 VDU7
  460 IF A$="Y" OR A$="N" THEN 480
  470 GOTO 440
  480 IF A$="Y" THEN PROCFOLD:V$="C":CLS:PROCSCORE
:PROCINITIALIZE:GOTO 370
  490 CLS:PRINT"How many cards to be replaced (MAX
 3) ";
  500 R$=GET$:R%=VALR$
  510 VDU7
  520 IF R%<0 OR R%>3 THEN GOTO 500
  530 IF R%=0 THEN CLS:GOTO 630
  540 PRINT''"Type in no.of card (1-5 from left)"
  550 FOR X%=1 TO R%
  560 A$=GET$:A%=VALA$
  570 IF A%>5 OR A%<1 THEN 560
  580 VDU 26:PROCREMOVE:PROCCHOOSENEW
  590 NEXTX%
  600 VDU 26:CLS
  610 PROCB:PROCDISPLAY
  620 VDU 28,0,31,39,13
  630 PRINT"These are your final cards"
  640 PRINT''"You may fold if you wish (Y/N)"
  650 A$=GET$
  660 IF A$="Y" OR A$="N" THEN 680
  670 GOTO 650
  680 IF A$="N" THEN 700
  690 GOTO 480
  700 CLS
  710 PROCCHANGECOM
  720 PRINT"I have changed ";EE;" cards"
  730 PROCEVALUATE
  740 IF B>0 THEN LL=0:GOTO 780
  750 IF RND(4)<4 AND Z1$="Y" THEN LL=1:GOTO 780
  760 IF RND(4)<3 THEN LL=1:GOTO 780
  770 CLS:V$="P":PRINT''"I have folded":PROCCOMFOLD
:PROCSCORE:PROCINITIALIZE:GOTO 370
  780 PRINT''''"Press any key to continue":A$=GET$
  790 CLS
  800 IF V$="C" THEN 920
  810 IF V$="P" THEN 840
  820 DD=RND(2)
  830 IF DD=2 THEN 920
  840 PRINT"You will bet first"
  850 PROCBETSCORE
  860 INPUT''"How much are you going to bet",V
  870 IF V<=0 THEN PRINT''"The bet must be more tha
```

```
n that!":INPUT V:GOTO 870
  880 IF V>999 THEN PRINT "Too much!":GOTO 860
  890 IF J1-V<0 AND LOAN=0 THEN PROCLOAN
  900 J1=J1-V:L1=L1+V
  910 GOTO 960
  920 CLS
  930 PROCWEIGHTING
  940 BET=INT((B*2)*RND(20))
  950 PRINT "I will bet you `";BET:L1=L1+BET:K1=K1
-BET:LIMIT=LIMIT+1:GOTO 1970
  960 LIMIT=LIMIT+1
  970 PROCWEIGHTING
  980 CLS
  990 IF V>B*50 AND B<2 AND L1<1000 AND RND(3)<2 T
HEN 770
 1000 IF B<2 AND V>B*40 AND K1<300 THEN 770
 1010 IF K1>300 AND V>K1 THEN 770
 1020 IF LOAN1>0 AND B<2 AND K1<1000 THEN 770
 1030 IF LIMIT=4 OR V>B*40 AND RND(3)<3 THEN L1=L1
+V:K1=K1-V:A$="I will see you":GOTO 1050
 1040 GOTO1940
 1050 CLS:PRINT A$
 1060 PRINT:PROCBETSCORE
 1070 PRINT''"Press any key to continue":A$=GET$
 1080 VDU 26:CLS
 1090 PROCB
 1100 PROCDISPLAY
 1110 PRINT'"My cards are:-":PRINT':PROCA
 1120 VDU 28,0,31,39,12
 1130 PROCDISPLAY
 1140 PRINT
 1150 PROCEVALUATE:SCORE(1)=B:PS(1)=KK
 1160 PROCB
 1170 PROCEVALUATE:SCORE(2)=B:PS(2)=KK
 1180 IF SCORE(1)=0 AND SCORE(2)=0 AND Z1$="N" THE
N 1330
 1190 IF SCORE(1)=SCORE(2) THEN 1460
 1200 IF SCORE(1)>SCORE(2) AND SCORE(2)<>0 THEN 18
80
 1210 IF SCORE(1)<SCORE(2) THEN 1910
 1220 IF Z1$="N" THEN 1330
 1230 IF SCORE(2)>SCORE(1) THEN 1910
 1240 IF SCORE(1)>0 THEN 1880
 1250 YY=5
 1260 PROCB:PROCASCENDING:W1$=S$(YY):W1=VAL(LEFT$(
W1$,2))
 1270 PROCA:PROCASCENDING:W2$=S$(YY):W2=VAL(LEFT$(
W2$,2))
 1280 IF W1=W2 THEN YY=YY-1:GOTO 1260
 1290 IF W2=1 THEN 1880
 1300 IF W1=1 THEN 1910
 1310 IF W1<W2 THEN 1910
 1320 GOTO 1880
 1330 PROCA
 1340 SCORE(3)=0
 1350 PROCSUIT
 1360 PROCSTRAIGHT
 1370 PROCROYAL
 1380 SCORE(1)=SCORE(1)+SCORE(3)
 1390 SCORE(3)=0
 1400 PROCB
 1410 PROCSUIT
 1420 PROCSTRAIGHT
 1430 PROCROYAL
 1440 SCORE(2)=SCORE(2)+SCORE(3)
 1450 GOTO 1230
 1460 IF PS(1)>2 OR PS(2)>2 THEN 1650
 1470 FF$=H$(1)
 1480 IF VAL(LEFT$(FF$,2))=1 THEN 1910
 1490 PROCA
 1500 PROCEVALUATE
 1510 IF VAL(LEFT$(H$(1),2))=1 THEN 1880
 1520 IF VAL(LEFT$(H$(1),2))<VAL(LEFT$(FF$,2)) THE
N 1910
 1530 IF VAL(LEFT$(H$(1),2))=VAL(LEFT$(FF$,2)) THE
N 1550
 1540 IF VAL(LEFT$(H$(1),2))>VAL(LEFT$(FF$,2)) THE
N 1880
 1550 YY=5
 1560 PROCASCENDING:W1$=S$(YY):W1=VAL(LEFT$(W1$,2)
)
 1570 PROCB:PROCASCENDING:W2$=S$(YY):W2=VAL(LEFT$(
W2$,2))
 1580 PROCA
 1590 IF W1=W2 THEN YY=YY-1:GOTO 1560
 1600 IF W1=1 THEN 1880
 1610 IF W2=1 THEN 1910
 1620 IF W1=2 OR W2=2 THEN W1=W2:GOTO 1590
 1630 IF VAL(LEFT$(W1$,2))<VAL(LEFT$(W2$,2)) THEN
1910
 1640 GOTO 1880
 1650 IF PS(1)>PS(2) AND SCORE(1)=SCORE(2) THEN 19
10
 1660 IF PS(2)>PS(1) AND SCORE(1)=SCORE(2) THEN 18
80
 1670 FOR N=1 TO 2:ST(N)=0:SS(N)=0:PT(N)=0:CC(N)=0
:NEXT
 1680 PROCB
 1690 TT=1
 1700 PROCEVALUATE
 1710 FOR N=1 TO 5
 1720 IF PP(N)>0 THEN ST(TT)=PP(N):PT(TT)=VAL(LEFT
$(S$(N),2)):TT=TT+1
 1730 NEXT
 1740 PROCA
 1750 TT=1
 1760 PROCEVALUATE
 1770 FOR N=1 TO 5
 1780 IF PP(N)>0 THEN ST(TT)=PP(N):CC(TT)=VAL(LEFT
$(S$(N),2)):TT=TT+1
 1790 NEXT
 1800 IF ST(1)>SS(1) AND SS(2)=0 THEN 1910
 1810 IF ST(2)=0 THEN 1880
 1820 IF ST(1)=SS(1) AND ST(2)=SS(2) THEN 1830
 1830 IF PT(1)>CC(1) AND PT(2)=0 THEN 1910
 1840 IF PP(1)>CC(1) AND PP(2)>CC(2) THEN 1910
 1850 IF PT(2)=0 THEN 1880
 1860 IF CC(1)>PT(1) AND CC(2)>PT(2) THEN 1880
 1870 IF PT(1)>CC(2) AND PT(2)>CC(1) THEN 1910
 1880 SOUND 1,-6,30,10:V$="C":PRINT'"I win":PRINT'
'"PRESS ANY KEY TO CONTINUE":A$=GET$
 1890 VDU7
 1900 K1=K1+L1:VDU 26:CLS:PROCSCORE:PROCINITIALIZE
:GOTO 370
 1910 SOUND 1,-6,85,10:V$="P":PRINT'"You win":PRIN
T''"PRESS ANY KEY TO CONTINUE":A$=GET$
 1920 VDU7
 1930 J1=J1+L1:VDU 26:CLS:PROCSCORE:PROCINITIALIZE
:GOTO370
 1940 BET=INT((B*2)*RND(20))
 1950 K1=K1-(BET+V):L1=L1+BET+V
 1960 PRINT "I have raised you by `";BET
 1970 PROCBETSCORE
 1980 PRINT'"These are your options:-"
 1990 PRINT'"1) Fold":PRINT"2) See Me":PRINT"3) Ra
ise Me"
 2000 PRINT'"Type in your option"
 2010 A$=GET$:R=VALA$
 2020 IF R<1 OR R>3 THEN 2010
 2030 ON R GOTO 2040,2050,2090
 2040 K1=K1+L1:V$="C":CLS:PROCSCORE:PROCINITIALIZE
:GOTO 370
 2050 A$="You have seen me"
 2060 L1=L1+BET:J1=J1-BET
 2070 IF J1<0 AND LOAN=0 THEN PROCLOAN
 2080 GOTO 1050
 2090 CLS:PROCBETSCORE
 2100 PRINT'"By how much do you wish to raise me"
 2110 INPUT V
 2120 IF V<=0 OR V>999 THEN 2110
 2130 L1=L1+V+BET:J1=J1-(V+BET)
 2140 IF J1<0 AND LOAN=0 THEN PROCLOAN
 2150 GOTO 960
 2160 DEFPROCBETSCORE
 2170 PRINT'"Your Score =`";J1
 2180 PRINT"My Score   =`";K1
 2190 PRINT"Pot        =`";L1
 2200 ENDPROC
 2210 DEFPROCSUIT
 2220 SUIT=0
 2230 FOR N=1 TO 4
 2240 IF RIGHT$(S$(1),2)=RIGHT$(S$(N+1),2) THEN SU
IT=SUIT+1
 2250 NEXT
 2260 IF SUIT>1 THEN SCORE(3)=SCORE(3)+(SUIT*.1)
 2270 IF SUIT=4 THEN SCORE(3)=SCORE(3)+2.5
 2280 ENDPROC
 2290 DEFPROCSTRAIGHT
 2300 PROCASCENDING
 2310 Z=0
 2320 FOR N=1 TO 4
 2330 IF VAL(LEFT$(S$(1),2))=(VAL(LEFT$(S$(N+1),2)
)-N) THEN Z=Z+1
 2340 NEXT
 2350PROCNOISE
 2360 IF Z=3 AND VAL(LEFT$(S$(1),2))=2 AND VAL(LEF
T$(S$(5),2))=1 THEN Z=4
 2370 IF Z>1 THEN SCORE(3)=SCORE(3)+(Z*.1)
 2380 IF Z=4 THEN SCORE(3)=SCORE(3)+2.25
 2390 ENDPROC
```

```
 2400 DEFPROCROYAL
 2410 Z=0
 2420 PROCASCENDING
 2430 FOR N=1 TO 5
 2440 Z=Z+VAL(LEFT$(S$(N),2))
 2450 NEXT
 2460 IF Z=47 AND VAL(LEFT$(S$(4),2))=13 THEN SCOR
E(3)=SCORE(3)+2.25
 2470 ENDPROC
 2480 DEFPROCLOAN
 2490 CLS
 2500 PRINT"Your score has dropped below zero"
 2510 PRINT'"You may have a loan of `1000 provided
   that you are able to pay it back in ten games w
ithout still being in debt"
 2520 PRINT'"Do you want the loan (Y/N)"
 2350 PROCNOISE
 2540 IF Z2$="Y" OR Z2$="N" THEN 2550
 2550 IF Z2$="N" THEN END
 2560 LOAN=1:J1=J1+1000
 2570 CLS
 2580 ENDPROC
 2590 DEFPROCDECIDE
 2600 PRINT "Ten games have now elapsed and your s
core minus the loan is ";J1-1000;
 2610 IF J1-1000<0 THEN PRINT"so unfortunately you
r game is over":END
 2620 PRINT ".Congratulations are in order"
 2630 Z2$="":LOAN=0:J1=J1-1000
 2640 PRINT'''"PRESS ANY KEY TO CONTINUE":A$=GET$
 2650 ENDPROC
 2660 DEFPROCCOMLOAN
 2670 PRINT "I have taken a loan of `1000 to clear
 my debt.If in 10 games I cannot pay this  back wi
thout going into debt again then you have won"
 2680 K1=K1+1000
 2690 PRINT'':ENDPROC
 2700 DEFPROCCOMDECIDE
 2710 IF K1-1000<0 THEN PRINT'"I am still in debt
after 10 goes.Therefore you have won":END
 2720 K1=K1-1000
 2730 LOAN1=0
 2740 ENDPROC
 2750 DEFPROCA
 2760 FOR N=1 TO 5:S$(N)=C$(N):NEXT
 2770 ENDPROC
 2780 DEFPROCB
 2790 FOR N=1 TO 5:S$(N)=P$(N):NEXT
 2800 ENDPROC
 2810 DEFPROCCHANGECOM
 2820 PROCA
 2830 PROCEVALUATE
 2840 IF B=0 THEN PROCASCENDING:GOTO 3090
 2850 IF B=4 AND Z1$="Y" THEN PROCASCENDING:GOTO 3
090
 2860 W=1
 2870 REPEAT
 2880 IF PP(W)=5 THEN W=6:GOTO 3080
 2890 IF PP(W)>0 THEN H=W:W=6:GOTO 2910
 2900 W=W+1
 2910 UNTIL W=6
 2920 F=PP(H)
 2930 IF H>=5 THEN 2970
 2940 FOR G=H+1 TO 5
 2950 IF VAL(LEFT$(C$(H),2))=VAL(LEFT$(C$(G),2)) T
HEN F=F+1
 2960 NEXT
 2970 HH=5-F
 2980 IF HH>3 THEN HH=3
 2990 E=H:EE=HH
 3000 REPEAT
 3010 E=E+1
 3020 IF E>5 THEN E=1
 3030 IF VAL(LEFT$(C$(E),2))=2 AND Z1$="Y" THEN 30
10
 3040 IF VAL(LEFT$(C$(H),2))=VAL(LEFT$(C$(E),2)) T
HEN 3010
 3050 PROCCHANGE
 3060 HH=HH-1
 3070 UNTIL HH=0
 3080 ENDPROC
 3090 T=RND(3)
 3100 IF T=1 THEN 3090
 3110 PROCA
 3120 FOR N=1 TO T
 3130 NN=NN+1
 3140 C$(N)=B$(NN)
 3150 NEXT
 3160 EE=T:GOTO 3080
 3170 DEFPROCASCENDING
 3180 FOR J=1 TO 5
 3190 FOR N=1 TO 4
 3200 M=VAL(LEFT$(S$(N),2)):MM=VAL(LEFT$(S$(N+1),2
))
 3210 IF M=1 THEN GOSUB 3260:GOTO 3240
 3220 IF M=2 AND Z1$="Y" AND MM<>1 THEN GOSUB 3260
:GOTO 3240
 3230 IF M>MM THEN GOSUB 3260
 3240 NEXT N,J
 3250 ENDPROC
 3260 IF MM=1 THEN RETURN
 3270 IF MM=2 AND Z1$="Y" THEN RETURN
 3280 L$=S$(N+1):S$(N+1)=S$(N):S$(N)=L$
 3290 RETURN
 3300 DEFPROCDEUCE
 3310 O=0
 3320 CC=1
 3330 FOR Z=1 TO 5
 3340 IF Z=N THEN 3370
 3350 IF VAL(LEFT$(S$(Z),2))=2 THEN CC=CC+1:GOTO 3
370
 3360 IF PP(Z)>0 THEN PP(Z)=PP(Z)+1
 3370 NEXT
 3380 FOR K=1 TO 5:O=O+PP(K):NEXT
 3390 IF O>0 THEN 3440
 3400 PROCASCENDING
 3410 KK=5
 3420 IF VAL(LEFT$(S$(KK),2))=2 THEN KK=KK-1:GOTO
3420
 3430 PP(KK)=PP(KK)+CC
 3440 ENDPROC
 3450 DEFPROCCHANGE
 3460 NN=NN+1
 3470 C$(E)=B$(NN)
 3480 ENDPROC
 3490 DEFPROCDISPLAY
 3500 FOR A=1 TO 9
 3510 FOR B=1 TO 35 STEP 7
 3520 PRINT TAB(B,A);STRING$(5,CHR$228)
 3530 NEXT B,A
 3540 RESTORE 3820
 3550 FOR A=1 TO 13
 3560 READ AA$(A)
 3570 NEXT
 3580 G=1
 3590 COLOUR 0:COLOUR 129
 3600 FOR A=0 TO 35 STEP 7
 3610 B=VAL(LEFT$(S$(G),2))
 3620 D$=RIGHT$(S$(G),2)
 3630 IF D$="BS"THEN E$=CHR$224
 3640 IF D$="DS" THEN E$=CHR$225
 3650 IF D$="TS" THEN E$=CHR$226
 3660 IF D$="ES" THEN E$=CHR$227
 3670 IF B>10 THEN 3760
 3680 FOR S=1 TO LEN AA$(B) STEP 2
 3690 Z=VAL(MID$(AA$(B),S,1)):ZZ=VAL(MID$(AA$(B),S
+1,1))
 3700 PRINT TAB(A+ZZ,Z);E$
 3710 NEXT S
 3720 IF B=10 THEN PRINT TAB(A+1,1);"10";TAB(A+4,9
);"10":GOTO 3780
 3730 IF B=1 THEN PRINT TAB(A+1,1);"A";TAB(A+5,9);
"A":GOTO 3780
 3740 PRINT TAB(A+1,1);B;TAB(A+5,9);B
 3750 GOTO 3780
 3760 PRINT TAB(A+1,5);AA$(B)
 3770 PRINT TAB(A+1,1);E$;TAB(A+5,9);E$
 3780 G=G+1:IF G=6 THEN A=35:GOTO 3790
 3790 NEXT A
 3800 VDU 20
 3810 ENDPROC
 3820 DATA 53,2383,235383,22248284,2224538284,2224
52548284
 3830 DATA 22245254738284,2224335254738284,2224424
45362648284,22243342446264738284
 3840 DATA JACK,QUEEN,KING
 3850 DEFPROCCOMFOLD
 3860 J1=J1+L1
 3870 ENDPROC
 3880 DEFPROCFOLD
 3890 K1=K1+L1
 3900 ENDPROC
 3910 DEFPROCREMOVE
 3920 SOUND 1,-6,53,5
 3930 X=1:Z=9
 3940 REPEAT
 3950 PRINT TAB((A%*7)-6,X);"      "
 3960 PRINT TAB((A%*7)-6,Z);"      "
 3970 X=X+1:Z=Z-1
 3980 UNTIL X=6
```

```
 3990 ENDPROC
 4000 DEFPROCCHOOSENEW
 4010 NN=NN+1
 4020 P$(A%)=B$(NN)
 4030 ENDPROC
 4040 DEFPROCSCORE
 4050 IF K1<0 AND LOAN1=0 THEN PROCCOMLOAN:LOAN1=1

 4060 IF Z1$="Y" THEN PRINT TAB(7);"WILD POKER"
 4070 PRINT'"YOUR SCORE =";J1
 4080 PRINT'"COMPUTER SCORE =";K1
 4090 IF LOAN>0 THEN PRINT'"You have ";10-LOAN;"
games left to clear your loan"
 4100 PRINT'''"      PRESS ANY KEY TO CONTINUE":A$=
GET$
 4110 PROCNOISE:CLS
 4120 IF Z2$="Y" THEN LOAN=LOAN+1
 4130 IF LOAN1>0 THEN LOAN1=LOAN1+1
 4140 IF LOAN>10 THEN PRINT'':PROCDECIDE
 4150 IF LOAN1>10 THEN PROCCOMDECIDE
 4160 ENDPROC
 4170 DEFPROCINSTRUCTIONS
 4180 CLS
 4190 FOR N=1 TO 2:PRINT TAB(9);CHR$(129);CHR$(141
);"INSTRUCTIONS":NEXT
 4200 PRINT''CHR$131;"USERS SHOULD KNOW HOW TO PLA
Y POKER."
 4210 PRINT CHR$131;"   THIS IS NOT A TEACHING GUI
DE!"
 4220 PRINT''"The program will play either wild or
    ordinary poker.In wild poker,2's count  as any
 card, but runs and flushes are   not allowed."
 4230 PRINT'''"Both the computer and the player st
art  off with `1000 ,but should either score drop
below zero, then loans of a further`1000 are allow
ed provided they can be  paid back in ten games ti
me."
 4240 PRINT''TAB(4);CHR$(136);"PRESS ANY KEY TO CO
NTINUE":A$=GET$:CLS
 4250 VDU7
 4260 PRINT"This is the winning order of hands :-"
 4270 PRINT''TAB(5)"1) Royal flush (A,K,Q,J,10 of
same"'TAB(8)"suit)."
 4280 PRINTTAB(5)"2) Straight flush - run of 5 in
"'TAB(8)"same suit."
 4290 PRINTTAB(5)"3) Four cards of same value."
 4300 PRINTTAB(5)"4) Full House - one pair and thr
ee"'TAB(8)"of the same suit."
 4310 PRINTTAB(5)"5) Flush - Five cards of same"'T
AB(8)"suit."
 4320 PRINTTAB(5)"6) Straight - run of 5 cards of"
'TAB(8)"any suit."
 4330 PRINTTAB(5)"7) Three cards of same value."
 4340 PRINTTAB(5)"8) Two pairs."
 4350 PRINTTAB(5)"9) One pair."
 4360 PRINTTAB(4)"10) Ace high downwards."
 4370 PRINT'TAB(12)CHR$130;"Good luck!"
 4380 PRINT''TAB(5);CHR$(136);"PRESS ANY KEY TO C
ONTINUE"
 4390 A$=GET$:CLS
 4400 VDU7
 4410 ENDPROC
 4420 DEFPROCINITIALIZE
 4430 *FX 15,1
 4440 L1=0:LIMIT=0
 4450 G=1
 4460 RESTORE 4530
 4470 FOR X=1 TO 4
 4480 READ A$
 4490 FOR N=1 TO 13
 4500 B$(G)=STR$(N)+" OF "+A$
 4510 G=G+1
 4520 NEXT N,X
 4530 DATA HEARTS,DIAMONDS,SPADES,CLUBS
 4540 FOR N=1 TO 30
 4550 I1=RND(52):I2=RND(52):I3=RND(52)
 4560 H$=B$(I3)
 4570 B$(I3)=B$(I1)
 4580 B$(I1)=B$(I2)
 4590 B$(I2)=H$
 4600 NEXT
 4610 NN=1
 4620 J=1
 4630 REPEAT
 4640 P$(J)=B$(NN):NN=NN+1
 4650 C$(J)=B$(NN):NN=NN+1
 4660 J=J+1
 4670 UNTIL NN=11
 4680 L1=L1+60:J1=J1-30:K1=K1-30
 4690 ENDPROC
 4700 DEFPROCEVALUATE
 4710 H$=""
 4720 FOR N=1 TO 5:PP(N)=0:NEXT
 4730 FOR N=1 TO 5
 4740 IF N=1 THEN 4800
 4750 IF VAL(LEFT$(S$(N),2))=2 AND Z1$="Y" THEN 48
00
 4760 FOR VV=1 TO N-1
 4770 IF VAL(LEFT$(S$(VV),2))=VAL(LEFT$(S$(N),2))
THEN PP(N)=10
 4780 NEXT
 4790 IF PP(N)=10 THEN PP(N)=0:GOTO 4870
 4800 X=N+1
 4810 REPEAT
 4820 IF N=5 THEN X=6:GOTO 4860
 4830 IF Z1$="Y" AND VAL(LEFT$(S$(X),2))=2 THEN 48
50
 4840 IF VAL(LEFT$(S$(N),2))=VAL(LEFT$(S$(X),2)) T
HEN PP(N)=PP(N)+1
 4850 X=X+1
 4860 UNTIL X=6
 4870 NEXT N
 4880 B=0
 4890 FOR N=1 TO 5
 4900 IF Z1$="Y" AND VAL(LEFT$(S$(N),2))=2 THEN PR
OCDEUCE
 4910 NEXT
 4920 KK=1
 4930 FOR N=1 TO 5:B=B+PP(N)
 4940 IF PP(N)>0THEN H$(KK)=S$(N):KK=KK+1
 4950 NEXT
 4960 ENDPROC
 4970 DEFPROCWEIGHTING
 4980 CC=0:Q2=0
 4990 IF B>3 THEN B=3
 5000 IF B=0 THEN B=2
 5010 Q1=VAL(LEFT$(H$(1),2))
 5020 ON B GOTO 5030,5080,5180
 5030 IF Q1=1 THEN 5060
 5040 IF Q1=1 THEN 5060
 5050 CC=CC+(f-(f1*(26-(Q1*2))))
 5060 CC=CC+e+d
 5070 GOTO 5180
 5080 IF KK=2 THEN 5120
 5090 IF Q1=1 THEN 5110
 5100 CC=CC+(d-(d1*(13-Q1)))
 5110 GOTO 5180
 5120 IF Q1=1 THEN5140
 5130 CC=CC+(e-(e1*(13-Q1)))
 5140 Q2=VAL(LEFT$(H$(2),2))
 5150 IF Q2=1 THEN 5170
 5160 CC=CC+(e-(e1*(13-Q2)))
 5170 CC=CC+d
 5180 B=3.44-CC
 5190 IF B<1 THEN B=1
 5200 IF Z1$="N" THEN 5240
 5210 FOR X=1 TO 5
 5220 IF VAL(LEFT$(C$(X),2))=2 THEN B=B+1
 5230 NEXT
 5240 ENDPROC
 5250 REPORT:PRINT" at line ";ERL
 5260 *FX11,15
 5270 END
 5280 DEFPROCNOISE
 5290 FORI=1TO2:SOUND1,-10,50,1:SOUND1,-10,55,1:NE
XTI
 5300 ENDPROC
 5310 DEFPROCMUSIC
 5320 LOCALI,B,C,T
 5330 ENVELOPE1,4,1,0,0,0,0,0,121,-10,-5,-2,70,0
 5340 RESTORE5440
 5350 FORI=1TO29
 5360 A$=INKEY$(0):IF ( A$="Y" OR A$="N" ) THEN EN
DPROC
 5370 READ B,C
 5380 IF I/6 = INT(I/6) THEN FOR T=1 TO RND(5):SOU
ND1,1,B,1:SOUND2,1,B+1,1:SOUND3,1,B-1,1:SOUND1,0,0
,0:SOUND 2,0,0,0:SOUND3,0,0,0:NEXT T
 5390 SOUND1,1,B,C:SOUND2,1,B+1,C:SOUND3,1,B-1,C
 5400 NEXT
 5410 TIME=0
 5420 REPEAT A$=INKEY$(0): UNTIL ( TIME >300 OR A$
="Y" OR A$="N" )
 5430 ENDPROC
 5440 DATA 0,4,48,2,40,7,60,7,68,7,77,8
 5450 DATA 77,4,82,2,77,7,49,7,49,7,69,8
 5460 DATA 69,3,61,3,40,3,48,3,56,3,48,3,56,3,70,3
,88,7,80,9
 5470 DATA 70,3,61,6
 5480 DATA 77,3,61,3,77,3,88,3,108,12
```

The Perfect Partnership

The features of this superb desk include

- ★ Effective and economical space saving design
- ★ Attractive Teak effect finish
- ★ Ample desk top space to house the latest Home Computers
- ★ Rear cut-outs for connecting cables and leads
- ★ Raised section to accommodate portable T.V.
- ★ Large shelf and drawer for storage
- ★ Stability
- ★ Durability
- ★ Practical, beautifully made and truly functional
- ★ Delivered flat-packed with full instructions for easy assembly
- ★ Specifications:
  Overall: 34½" H, 38¼" W, 18¾" D.
  Platform: 6" H, 18½" W, 9" D.
- ★ Excellent value

ONLY £39.95 +P&P

POST THIS COUPON TO:

DISCOUNT Computer Supplies

14 SANDPIT PLACE, LONDON SE7 8HE

Please send me ..... Grenville Computer Desks at £39.95 + £10 P&P. I enclose a CHEQUE/POSTAL ORDER for £..... payable to DISCOUNT Computer Supplies.

Name .....

Address .....

.....

Signature .....

Overseas – check for carriage

Please allow 28 days for delivery

ORIC AND SINCLAIR COMPUTERS

Oric 1 computer 48K £143 (£141) £151. Oric colour printer £134 (£123) £140. Sinclair Spectrum 48K £131 (£131) £143. Spectrum 16K £101 (£105) £117. 32K memory upgrade kit for 16K Spectrum (issue 2 and 3 only) £31 (£28) £30. Fuller Master Unit for the Spectrum including speech unit, sound synthesizer, amplifier and joystick port £56 (£56) £62. Fuller full sized FDS keyboard for the Spectrum with proper space bar £52 (£52) £62. Spectrum printer interfaces with cables:- Centronics £57 (£53) £58m RS232 £35 (£33) £38. ZX printer with 5 free rolls paper £41 (£46) £56. ZX printer alone £36 (£38) £50. 5 printer rolls £13 (£16) £21. Special offer pack ZX81 computer + 16K ram pack + four game tapes £55 (£61) £71. ZX81 16K ram packs £31 (£28) £30.

COMMODORE COMPUTERS

Commodore 64 £204 (£184) £204. Vic 20 £104 (£97) £117. Convertor to allow most ordinary mono cassette recorders to be used with the Vic 20 and Commodore 64:- built £9.78 (£9) £11, kit £7.47 (£7) £9. Bargain package:- cassette convertor + compatible cassette recorder £37 (£38) £44. Commodore cassette recorder £43 (£44) £50. Centronics printer interfaces:- for Vic 20 £42 (£41) £46, for Commodore 64 £45 (£44) £50. 1541 Disc drive £233 (£209) £234. 1525 Printer £235 (£220) £245. 1526 Printer £350 (£330) £360.

ACORN COMPUTERS

Electron £203 (£209) £229. BBC Model B £424 (£388) £408. Kenda double density disk interface system for beeb £139 (£125) £135. We stock the whole range of Cumana disc drives for the beeb e.g. 100K single £230 (£220) £240, Double 2 x 400K £625 (£560) £580.

PRINTERS

Epson RX80 £326 (£309) £340. Epson RX80F/T £346 (£316) £346. Shinwa CTI CP80 £293 (£271) £312. Epson FX80 £440 (£480) £438. Epson MX100/3 £494 (£465) £495. Seikosha GP100A £234 (£219) £254. Oki Microline 80 £223 (£207) £248. Oki Microline 84 £831. The Ultra 21 combined daisy wheel and electric typewriter £438 (£415) £445. The brother EP22 combined matrix printer and electric typewriter £173 (£166) £186. Juki 6100 proportional daisy wheel printer £423 (£404) £434. MCP40 colour printer £134 (£123) £140. Star STX80 thermal printer £165 (£159) £169. Interfaces to run the above printers from Commodore machines:- Vic 20 £42 (£41) £46, Commodore 64 £45 (£44) £50. We can supply interfaces to run the above printers from Sharp computers £58 (£52) £55.

SWANLEY ELECTRONICS

Dept CT, 32 Goldsel Road, Swanley, Kent BR18 8EZ, England.
TEL: Swanley (0322) 64851

Nothing extra to pay. All prices are inclusive. UK prices are shown first and include post and VAT. The second price in brackets is for export customers in Europe and includes insured air mail postage. The third price is for export customers outside Europe (include Australia etc) and includes insured airmail postage. Official orders welcome.

COMPUTER ROBOTICS TRAINING

FULL TIME COLLEGE COURSE

SUITABLE FOR APPLICANTS WHO WISH TO ENTER COMPUTER SERVICE OR RELATED INDUSTRIES — HIGH PERCENTAGE OF PRACTICAL COURSE WORK

15 MONTHS
B TEC Certificate in Computing Technology

9 MONTHS
B TEC Higher Certificate in Computing Technology

Subjects: Foundation Electronics, Digital Techniques, Microelectronics, Microprocessors, Microcomputer Based Systems, Industrial Robotics, Machine Code & High Level Programming.

Shortened courses can be arranged for applicants with previous knowledge.

Courses commence Jan, April and Sept (Higher Cert Sept only). Prospectus from:

LONDON ELECTRONICS COLLEGE (Dept C5/6)
20 Penywern Road,
Earls Court, London SW5 9SU.
Tel: 01-373 8721

Kuma

~a unique pool of expertise

EPSON

HARDWARE PLUS
Software from the home of the Deskmaster series — the major source for HX-20 Software

SHARP

HARDWARE PLUS
A wealth of Software from the originators of Sharp approved Software for the MZ700 — also MZ80A, MZ3541 and PC1500

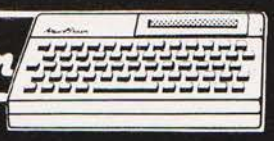
commodore 64

HARDWARE PLUS
Accounts, Database and Toolkit Software

SAGE

HARDWARE PLUS

NewBrain



68,000 based Forth

HARDWARE PLUS
Pascal, Forth and much, much more

SIRIUS 1 apricot

HARDWARE PLUS
Kuma Forth, Mailbox and much more

ELECTRONIC MAIL BOX
Software, advice, modems and acoustic couplers from experienced users

PHONE NOW
PANGBOURNE
(073 57) 4335
or SEND FOR DETAILS

PRINTERS
A great range of Matrix and Daisywheel

Kuma Computers Ltd Unit 12, Horseshoe Park, Horseshoe Road, Pangbourne, Berks RG8 7JW COMET: KUMA
Telex 849462 TELFAC KUMA TELECOM GOLD: KUM ØØ1

Please tick box ☑ for information required, and send coupon to:
Kuma Computers, Ltd. 11 York Road, Maidenhead, Berks SL6 1SQ

Hardware	Software	Hardware	Software
☐ SIRIUS	☐	☐ EPSON HX-20	☐
☐ SAGE	☐	☐ EPSON QX-10	☐
☐ SHARP 700	☐	☐ NEWBRAIN	☐
☐ SHARP A	☐	☐ COMMODORE 64	☐
☐ SHARP 3541	☐	☐ ELECTRONIC MAILBOX	☐
☐ SHARP PC1500	☐	☐ PRINTERS	☐

NAME .....
ADDRESS ..... CT
POST CODE .....

Henry Budgett

SCOPE GRAPHICS LANGUAGE

Wouldn't it be nice to be able to get the speed of machine code without the programming headaches? This new language for the Spectrum offers a simple implementation of arcade-type graphics.

New languages seem to arrive on the micro scene with almost as predictable a frequency as new computers. Many of these so-called 'new' discoveries are really just microcomputer implementations of existing mainframe and mini languages that have been around for years. All the fuss over FORTH fades slightly when you realise that it has been around since 1970 and was originally written on an IBM 1130! However, some of the more recent offerings do provide facilities previously unavailable on the mainstream languages such as BASIC and Pascal.

One such product is SCOPE, the acronym standing for Simple Compilation of Plain English, and its prime area of application is the programming of games and simple graphics displays. SCOPE doesn't replace BASIC but is co-resident both during development and execution so programs can be in both languages, taking advantage of their respective strengths.

In launching the product the publishers of SCOPE, ISP Marketing of Godalming, have been at considerable pains to point out that it is not just another menu-driven games creator. You really do have to design and program all the results yourself and these can be quite impressive, despite the limitations of just 31 commands.

OVERVIEW

SCOPE is claimed to be a "compiled, multitask language" and "easier to understand than BASIC yet writes in machine code"! Behind this lurks the basis of what SCOPE is really all about. The problems connected with writing fast graphics in BASIC are generally well-known: just try writing a game like 'Defender' in ZX BASIC and see how fast it goes if you aren't convinced! Equally well appreciated is the fact that machine code programming is something best tackled by those who gain a degree of masochistic pleasure from sapping their brains. Mere mortals like to hang onto their sanity.

What SCOPE sets out to provide is a fast and easy way to create machine code graphics without the programmer actually realising what is happening. To this end all the programming is done with a small set of commands that look and sound like BASIC we are all familiar with. However, what you are actually doing when you write a SCOPE program is to create links to ready-written routines within the SCOPE language. Anyone even remotely familiar with assembly language programming will spot the similarities straight away.

Because all these routines are in machine code they run many times faster than their equivalents in BASIC, a speed increase which has to be seen to be appreciated. Unfortunately the claims made for SCOPE in its promotional literature are now looking slightly ragged. If you really were producing machine code directly from your SCOPE source program you shouldn't need to have the SCOPE language there at all when you RUN the 'compiled' version. I haven't been able to fully track down the inner workings of the language but it would appear that it actually consists of a suite of macros, machine code procedures in effect, which are linked together by the 'compiler' as it interprets the source code.

Regardless of how it actually works internally, it does genuinely do the job, rather too well in some ways! Whether it is "easier to understand than BASIC" is a matter for the user to decide. I found programs somewhat tedious to construct; the obligatory punctuation is easy to miss, especially with the ZX Spectrum's keyboard. What can also be a problem is the fact that the SCOPE syntax checker doesn't prevent out-of-range commands. While these cause minor frustration in BASIC they are fatal in the machine code environment of SCOPE.

GETTING STARTED

Getting SCOPE into the ZX Spectrum is no problem at all: the normal "LOAD" sequence works just fine. A short loader program boots the main machine code program into memory. What happens when loading is complete is, at first, somewhat surprising. The manual says that the screeen and low memory will be cleared ready for you to start. What actually happens is that the screen goes black and then returns to the initial turn-on display. With muffled cursing you do it again, assuming that the cat ate the power lead or your central heating is playing games with you.

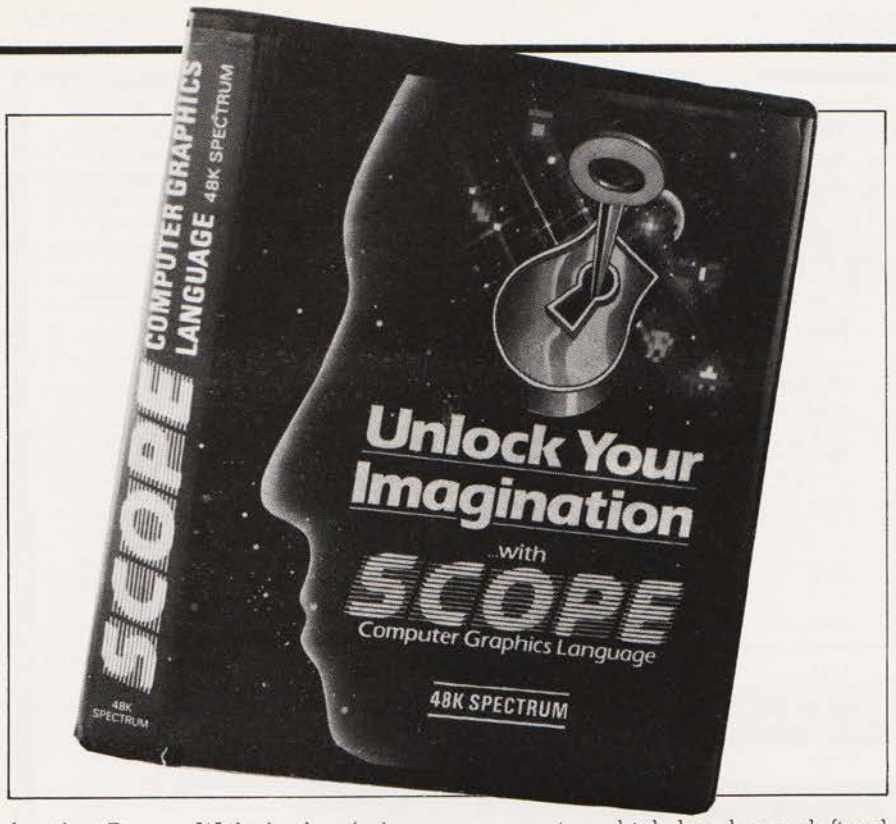
Needless to say, the same thing happens again. Well, I fell for it and there's no reason to suppose that others won't! Of course, when you sit and think about it, you would expect it to do a warm reset but quite why they don't tell you is beyond me.

BACK TO BASICS

All SCOPE programs have to be written in BASIC. Well, every line of SCOPE has to start with a line number and the BASIC command REM. This, of course, is done so that the Spectrum's interpreter doesn't try to disentangle SCOPE statements as BASIC. What comes after the REM is checked by the SCOPE interpreter for syntax so it is not practical to build SCOPE source code into BASIC programs and 'compile' the lot. One of the most tiresome things about writing a SCOPE program is the fact that you have to type line numbers and the keyword REM for each line — an auto numbering/REM facility would be much appreciated.

The first command in any SCOPE program must be Org; which defies the area of memory where the SCOPE program will reside in 'compiled' form and the area where routines will be stored. Typically these are 40000 and 50000 respectively, although they can be shuffled around at will. The figures above allow you 16K for source code with 10K reserved for 'compiled' SCOPE program and another 10K for any SCOPE routines.

Because of its low-level nature SCOPE requires that anything you want to refer to as a variable must be given a name. There are two variable types: eight-bit integers called Vars and 16-bit integers called Bvars. You can have up to 52 of each, the letters a to z and A to Z being used to name them. Labels within a program also use these letters but while a Bvar and Var may both have the name A and be different, labels can only be used once. This restricted naming facility has to be used carefully or the program rapidly becomes unreadable. I found that one easy solution was to use the little letters for the Vars and the BIG letters



for the Bvars. With both of these working from A towards Z you can use the little letters from z towards a for normal labels and capital letters from Z to A to label routines.

STRUCTURES

There are no luxuries provided in the SCOPE instruction set. Impassioned devotees of structured programming are going to have to grit their teeth and remember how to write real programs all over again! Subroutines can be created and labelled uniquely and there are two conditional commands as well as an unconditional Jump.

Routines are delimited by the commands Routine; and End; and the code produced as a result is stored in a separate area of memory set aside by the Org; statement. The first real statement within any routine should be a Label; so that the routine can be called from within the main program: jumps to a routine label are not prevented by SCOPE but disasters will occur if you try it.

The simplest conditional statement is Lim; which allows a maximum value to be set for a Bvar. When the Bvar reaches this value the programs jumps to the specified label rather than continuing in the normal way. This test can be put anywhere within a loop and effectively acts either as a FOR. . .NEXT or a DO. . .UNTIL depending on whether the test is made before or after the Bvar changes value.

As much more complex testing is likely to be needed a second conditional statement called Test; is provided. This tests a Var against a number and then branches to the defined label depending on the test parameter which has been defined. It is possible to check for the two values being 'equal', 'not equal', 'less than' and 'greater than or equal to' each other and the branch may be either a Jump; or a Call;.

The Test; structure can be regarded as approximating to most of the possible combinations of IF. . .THEN GOTO or IF. . .THEN GOSUB. Quite why the designers of SCOPE left out tests for 'greater than without being equal to' and 'less than or equal to' I've no idea. Finding ways around these deficiencies is not exactly difficult, just a matter of remembering which way round to put things in the test sequence.

One thing that is likely to befall you sooner or later, and from experience I'd say sooner, is that you'll create an infinite loop. Unlike good old BASIC you cannot Break out of a running SCOPE program, it's the power switch or nothing! A helpful hint here is to always stick a Get; somewhere in your program. Make the tested key something really obscure like the copyright symbol and you'll have an escape route during development.

SCORING ON SCREEN

Because SCOPE is generally going to be used to produce graphics displays the screen handling facilities really break down into two sections. The mundane tasks of putting words and numbers on the screen are directly handled by the commands Put; and Num;. The Put; command places text or graphics characters on the screen at the defined row and column position. The colour of the text is governed by the first parameter in the command which defines both the PAPER and INK colours.

All the SCOPE commands which affect the colour of the display use a single number to control both foreground and background colour. The number is calculated by adding the ink number to eight times the paper number. If you want the result to be BRIGHT add 64 to the result, for FLASH add 128 and for both add 192. Clever and simple but in the case of both Put; and Num; the colour number is fixed, it cannot be a variable. Equally odd is the fact that while the row and column position of Put; can be variables those of Num; cannot. Presumably the designers reckoned that scores should never move around!

The Num; command is much the same as Put; (except for the restrictions just mentioned) but it causes the contents of a Bvar to be printed on the screen. No equivalent command exists to place the contents of a Var on the screen nor, apparently, can you transfer the value of a Var to a Bvar. Why, you may ask, did they bother with Var at all? Well, the answer must surely be speed. It takes twice as long to process a two-byte number as it does to process a one-byte one so use Var for all your internal counters and loops and save Bvar for results and scores.

Other facilities provided by SCOPE include Bdr; which changes the BORDER colour, Chg; which clears the screen and changes the PAPER and INK colours, Wipe; which erases a specified number of lines from the bottom of the screen and Scr; which scrolls the screen upward.

SOUNDING OUT GRAPHICS

As you would expect from a programming language designed to make games faster, SCOPE offers a reasonable range of graphics facilities. The Plot;, Draw;, and Attr; commands are nearest equivalents to their BASIC counterparts apart from the syntax. To ensure that lines crossing one another don't cause problems the designers also included Over; which acts exactly the same as BASIC's OVER. Unfortunately they left out any form of circle drawing function and because you cannot get at BASIC's SIN and COS functions very easily from within SCOPE, this could prove a little bit of a problem.

Probably the single most useful graphics facility provided within SCOPE is the Fscr; command. When invoked this performs a pixel-step scroll in the specified direction. The numbers required relate to the cursor key values so it's easy to remember

which way they work. Because of the speed it is possible to achieve a very smooth diagonal scroll by following each vertical axis scroll command by the required horizontal movement.

It would have been very nice to see some extra facilities provided by the language: a Fill; command would have obvious uses and all the user-defined graphics need to be set up by a BASIC program before SCOPE can use them. Obviously any co-resident language has to compromise between facilities that can be handled by the native BASIC and those that would take up more memory if they were included in the new language.

SCOPE augments its graphic displays with the Sound; command. Provided with two parameters, the function creates sounds from two octaves below Middle C to three above. The duration of the sound should be linked to the pitch or you end up wondering if the noise is ever going to stop! Although it is only intended to be used for laser bursts and similar noises the speed of the command allows some really quite pleasing effects to be created, assuming you can actually hear them on the ZX Spectrum's tiny speaker.

Table 1. Scope Instruction Set.

System Commands:	
Note	Comment, ignored by SCOPE
Org	Defines the addresses in memory where the SCOPE program and its routines will be stored when compiled
Exit	End of the SCOPE program, returns control to BASIC
Structures:	
Add	Adds given constant to specified Bvar
Bvar	Defines a Bvar and its initial value
Call	Call a Routine by its Label, equivalent to a procedure call in BASIC
Dec	Subtract given constant from specified Var
End	End of Routine, equivalent to ENDPROC in BASIC
Get	Get the code value of the last key pressed
Halt	Wait on current frame for specified number of frame scans
Inc	Add given constant to specified Var
Jump	Jump to label but NOT a routine. Equivalent to GOTO in BASIC
Label	Defines a label
Lim	Sets limit for Bvar and the label to Jump to when it equals that limit
Minus	Subtract given constant from specified Bvar
Rnd	Generate a random number within specified limits
Routine	Start of a Labelled SCOPE routine
Test	Compares a Var with given constant and Jumps or Calls specified label depending on the condition set. Possibilities are 'equal to', 'not equal to', 'less than' and 'greater than or equal to'
Var	Define a Var and its initial value
Display:	
Bdr	Sets BORDER colour
Chg	Clears screen and sets PAPER and INK colours
Num	Displays Bvar in specified colour at given screen position
Scr	Scrolls screen upwards by specified number of lines
Wipe	Clears specified number of lines from bottom of screen
Graphics:	
Attr	Reads the colour attributes of the specified screen co-ordinate
Draw	Draws a line from the last cursor position to the new relative position in the specified colour
Fscr	Fine scrolls the screen in pixel steps. Directions supported are Up, Down, Left and Right and their diagonal combinations
Over	Allows overprinting of existing screen image without destroying it
Plot	Plots a point at given co-ordinates in specified colour
Sound:	
Sound	Produces note of specified pitch for specified period

FOLLOWING THE BOOK

The Reviewer's prayer must surely be "Oh Computer Maker, please let there be a good manual written just once in my career!". Unfortunately the SCOPE Instruction Manual wasn't to be it: I think I had mine about four years ago when I reviewed the HP-85. It is probably fair to say that my manual was an 'early version'. At least, I hope it was for the sake of the future SCOPE user. Quite apart from its content, which I'll come to in a moment, the presentation leaves a little to be desired. I actually had to break the spine before I could fold the pages flat in order to read them: the booklet is covered in very tough plastic which tends to spring it shut just as you find the place you wanted.

Out of the 48 pages of text — each is roughtly one-third the size of a page of this magazine — 19 are devoted to listing the language's command set. Once you have stumbled your way through this section, it's actually not at all bad, but you are then faced with the problem of implementing the commands as part of a program. Apart from the slender examples at the end of the booklet which give the most cursory explanation of their operation, you are on your own. This total lack of tutorial on the various structures and how to create them is, to put it bluntly, where the whole concept starts to fall apart. If SCOPE really is to be considered "easier to understand than BASIC" (their words, not mine) then the manual makes rather too many assumptions. If you've ever written programs in assembler or FORTH then SCOPE is not going to be any problem, but the first-time BASIC programmer is going to be out of his depth.

Probably the most terrifying omission from this manual is any real explanation of how to combine a compiled SCOPE program with BASIC. The glib explanation given in the manual simply doesn't go into the process in any depth. Remember, this is a product which writes machine code programs and it is supposed to be used by people who don't want to learn machine code!

THE FINAL ACT

The concept of SCOPE is remarkable: it certainly provides programs which run at speeds no BASIC routines ever could. Furthermore it encourages the programmer to learn structured programming techniques from first principles — real assembly language programming will be a simple step for a proficient SCOPE coder. Whether all the claims made for the language are really justified I'm not sure: it'll take some hard talking to convince me that it really is a compiler and the use of the word 'multitask' is definitely out of order.

The fact that SCOPE is a real language means, in theory, that any game which you might program in BASIC will run faster if you write it in SCOPE. Probably the best way for a novice to tackle the process would be to write it all in BASIC and then re-write the slow bits in SCOPE. Because SCOPE always has to be resident you lose about 8K of the memory space, but as the compiled code is more efficient this is not necessarily going to be a problem.

The real question is whether you want to program in a half-way language. With products like Picturesque's Editor/Assembler readily available, assembly language, while not easy, is something that will offer better results in the end than SCOPE. For the first-time BASIC programmer who still regards machine code as a total mystery SCOPE will at least instill some of the basic concepts needed for the next step. What that first-time user doesn't get from the product is the documentary support that he or she is likely to need.

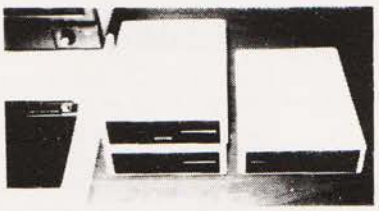
BBC Microcomputer System

OFFICIAL BBC COMPUTER DEALER

This 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 =	**£399.00**
BBC Mod B + disk interface	£409 + VAT =	**£469.00**
BBC Mod B + Econet interface	£389 + VAT =	**£447.35**
BBC Mod B + disk and Econet interfaces	£450 + VAT =	**£517.50**
BBC 100K disk drive	£230 + VAT =	**£264.00**
BBC dual 800K disk drive	£699 + VAT =	**£803.85**
Torch Z80 disk pack including Z80 2nd processor, 64K RAM and CPN operating system + Free Perfect Software	£699 + VAT =	**£803.85**
BBC Teletext receiver (Aug)	£196 + VAT =	**£225.40**
BBC cassette recorder and lead	£26 + VAT =	**£29.90**
Disk interface kit (free fitting)	£86 + VAT =	**£96.60**
Mod A to Mod B upgrade kit	£50 + VAT =	**£57.50**
Fitting charge for A to B upgrade kit	£20 + VAT =	**£23.00**
16K memory upgrade kit	£20 + VAT =	**£23.00**
Games paddles	£11 + VAT =	**£12.65**
12" Monochrome monitor incl. cable	£89 + VAT =	**£102.35**
16" Colour monitor incl. cable	£209 + VAT =	**£240.35**
User guide	£10 + VAT =	**£10.00**
Econet interface (free fitting)	£60 + VAT =	**£69.00**
Speech interface (free fitting)	£47 + VAT =	**£54.05**
BBC disk manual + formating disk	£30 + VAT =	**£34.50**
Parallel printer cable	£10 + VAT =	**£11.50**
BBC word processor (view)	£52 + VAT =	**£59.80**
BBC Fourth language cassette	£15 + VAT =	**£17.25**
BBC Lisp language cassette	£15 + VAT =	**£17.25**

100% BBC COMPATIBLE MITSUBISHI AND TEAC SLIMLINE DISK DRIVES



These drives are supplied ready cased with all the necessary cables, formating program and user disk system guide.

There are some useful utilities included, e.g. Epson Screen Dump Program, Memory Dump, Free, Duplicate, Merge and Relocate. Power consumption of these drives is very low (0.2A typ. at + 12V, 0.4V typ. at + 5V per drive). Power is taken from the BBC computer.

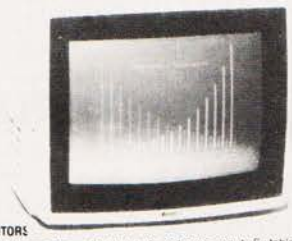
Single drive 100K 40 tracks	£169 + VAT = **£194.35**
Dual drive 200K 40 tracks	£329 + VAT = **£378.35**
Single drive 400K 80 tracks	£239 + VAT = **£274.35**
Single drive 400K 40/80 tracks switchable	£259 + VAT = **£297.85**
Dual drive 800K 80 tracks	£449 + VAT = **£516.35**
Dual drive 800K 40/80 tracks switchable	£469 + VAT = **£539.35**

✻ COMPLETE WORD PROCESSOR FOR ONLY £1,099 + VAT ✻

This package consists of BBC Microcomputer, View wordprocessor, 400K Slimline disc drive, High resolution 12" Green monitor, Juki 6100 18CPS Daisy Wheel printer and all the necessary cables and documentation. The above package can be supplied with components of your own choice: e.g. 800K disc drive or a different printer. Please phone us for a price for your particular requirement.

Special package deal £1,099 + VAT = **£1,263.85**

PROFESSIONAL MONITORS



GREEN MONITORS

12" Green screen monitors with composite and sync. input. Suitable for most computers.

✻ 18 MHz band width, high resolution	£89 + VAT =	**£102.35**
✻ 15 MHz band width, normal resolution	£69 + VAT =	**£79.35**

COLOUR MONITORS

✻ MICROVITEC RGB input 14" monitor supplied with RGB lead for BBC	£209 + VAT = **£240.35**
✻ SANYO SCM 14" Normal res. 14", 400 dots, RGB input supplied with RGB lead	£199 + VAT = **£228.85**
✻ SANYO SCM 14M Medium res. 14", 600 dots, RGB input supplied with RGB lead	£299 + VAT = **£343.85**
✻ SANYO SCM 14H High res. 14", 800 dots, RGB input supplied with RGB lead	£399 + VAT = **£458.85**

EPSON FOR RELIABILITY

EPSON FX80: 80 column, 160 CPS, normal, italic and elite characters, 256 user definable characters, superscript, subscript, 11 x 9 matrix, bi-directional logic seeking, hi-res bit image printing (960 x 8 dots line), friction and tractor feed, 9 international character sets, Centronic parallel interface.

FX80 PRICE £349 + VAT = **£401.35**

EPSON RX80: 80 column, 100 CPS, normal, italic and elite characters, 11 international character sets, hi-res bit image printing, bi-directional logic seeking, 4 to 10" adjustable pin feed, Centronic parallel interface.

RX80 PRICE £239 + VAT = **£274.84**

MX-100: 136 column, 100 CPS, friction and tractor feed, up to 15" adjustable carriage, hi-res bit image printing, true descenders, Centronic parallel interface.

MX-100 PRICE	£399 + VAT = **£458.85**
RS232 Interface for all above printers	£55 + VAT = **£63.25**
EPSON RX80FT (friction & tractor)	£269 + VAT = **£309.35**
FX100	£479 + VAT = **£550.85**
Roll holder for FX80	£12 + VAT = **£13.80**
Ribbon for MX80, FX80, RX80	£8 + VAT = **£9.20**
Ribbon for MX100	£12 + VAT = **£13.80**

SEIKOSHA DOT MATRIX PRINTERS WITH HIGH-RES GRAPHICS

GP-100A 80 column, 50 CPS, dot addressable hi-res graphics, 10" wide, fully adjustable, tractor feed, 7 x 5 print matrix, Centronic parallel interface.

GP-100A 50CPS PRICE £175 + VAT = **£201.25**

GP-250X 80 column, 50 CPS, 10" wide, fully adjustable, tractor feed, true descenders, 64 user definable characters, double height and or double width printing, 8 x 5 print matrix, Centronic parallel and RS232 (serial) interfaces both included.

GP-250X PRICE £219 + VAT = **£251.85**

NEW GP-700A 7 COLOUR PRINTER

This latest addition to Seikosha range gives you print in seven colours, 10" wide carriage, friction and tractor feed, 50 CPS print speed, dot addressable high-res graphics, 4 hammer printing mechanism, 10 CPI or 13.3 CPI, special Quite printing mode, Centronic parallel interface.

GP-700A SPECIAL INTRODUCTORY PRICE £349 + VAT = **£401.35**

GUARANTEED LOWEST PRICES

We guarantee that our prices are the lowest on the market. If you can find any item advertised and in stock at less than our price we will match that price.

NEW LOW PRICES ON STAR

ONLY £239 + VAT

The most cost effective quality matrix printers to be launched this year. DP510 and DP515 features include friction and tractor feed and roll holders as standard. 100 CPS print speed, bi-directional logic seeking, 9 x 9 matrix gives true descenders, 2.3K buffer as standard, hi-res bit image plus block graphics, sub and super script, italic printing, auto underlining, vertical and horizontal tabulation, left and right margins set, skip over perforation, back space and self test.

STAR GEMINI 10 10" carriage, 80 columns, 120 cps

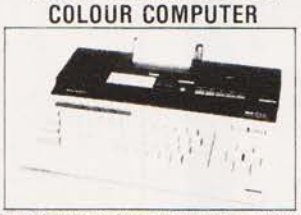
SPECIAL PRICE £239 + VAT = **£274.85**

STAR DP515 15" carriage 136 columns

SPECIAL PRICE £279 + VAT = **£320.85**

RS232 INTERFACE FOR ABOVE £50 + VAT = **£57.50**

SHARP MZ 700 SERIES COLOUR COMPUTER



The new Sharp MZ 700 has the flexibility to run programs in BASIC, FORTRAN, MACHINE CODE, PASCAL, ASSEMBLER and many other languages. The 64K memory caters for advanced programming, and the MZ 700 gives you access to a wide choice of software.

£60 worth of software FREE — ten games

SPECIAL PRICE £199 + VAT = £228.85

TEXAS INSTRUMENTS TI 99/4A

This microcomputer is based on TMS9900 16-bit microprocessor. It includes 16K RAM, 16 colour high resolution graphic (192 x 256). The screen display is 32 characters, 24 lines TI-BASIC. Full-size keyboard. For Software there are about 1000 programs to choose from. There are a lot of peripherals available, e.g. Disk Drives, Disk Interface, Speech Synthesizer, Extra RAM. Additional Language (PASCAL, TI-LOGO, ASSEMBLER).

TI HOME COMPUTER HARDWARE

Title	Description	Price inc VAT
T199/4A	Complete with UHF modulator and power supply + free cassette lead	£89.95
PERIPHERALS		
Speech Synthesizer	When used with selected modules will reduce electronic speech	£41.95
Peripheral Expansion System	This unit takes all card peripherals and on-internal disk drive	£79.95
Disk Drive – Internal	92K formatted drive, mounts internally in peripheral expansion system	£149.95
Disk Controller Card	Controls up to 3 disk drives, complete with disk manager command module	£109.95
Disk Drive Double Sided	92K formatted capacity per side acts as 2 drives DSK1 & DSK2 total capacity 184K bytes	£219.95
Disk Drive External	Complete with own case, power supply & connecting cables	£259.95
RS232 Expansion Card	Provides 2 serial RS232 ports, and one parallel port for interfacing	£79.95
RAM Expansion Card	Adds 32K bytes extra RAM bringing total capacity to 48K bytes	£79.95
P-Code Card	Includes the UCSD-PASCAL P-code interpreter	£129.95
Matrix Printer	80 column matrix printer pinter GP-100A - cable	£219.95
Matrix Printer GP250X	80 column matrix printer with RS232 and Centronic parallel interface	£273.95
Epson RX80 Printer	80 column, 100 CPS matrix printer	£320.85

Please send S.A.E. for software prices

THE AFFORDABLE DAISYWHEEL PRINTER

ONLY £369 + VAT

- ★ 18 CPS – Bi-Directional Logic Seeking
- ★ 10, 12, 15 CPI – Proportional Spacing
- ★ "Drop in" Daisywheel – Triumph Adler Compatible
- ★ Supports all Wordstar features
- ★ Diablo protocols – IBM Selectric ribbon
- ★ 2K Buffer as standard – 100 character Daisywheel

SPECIAL OFFER JUKI 6100 DAISYWHEEL £369 + VAT = £424.35

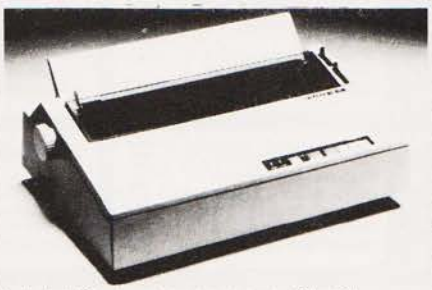
THE CP80 QUALITY PRINTER

ONLY £239 + VAT

- ★ 80 CPS – Bi-Directional Logic Seeking 80 Column
- ★ Friction and Adjustble Tractor Feed
- ★ Patented Square Needles up to 9 x 13 matrix
- ★ Hi-Res Graphics and Block Grphics

SHINWA CP80 PRINTER	£239 + VAT = **£274.85**
SPARE RIBBON FOR CP80	£5 + VAT = **£5.75**

BROTHER HR-15 PRINTER



The Brother HR-15 high specification daisywheel printer. RRP is £500

SPECIAL OFFER

3 FREE daisywheels supplied!

4 FREE spare ribbons

ONLY £399 + VAT = £458.85

Akhter Instruments Limited

Dept.

28 BURNTMILL, HARLOW, ESSEX CM20 2HU. UK.

TEL: HARLOW (0279) 443521 OR 412639

TELEX 995801 REF — A18

ORDERING INFORMATION

All orders which accompany a cheque, cash or postal orders are *CARRIAGE FREE* (UK only). Please make cheques and postal orders payable to "AKHTER INSTRUMENTS". A carriage charge of 3% of invoice total is applicable to BARCLAYCARD and ACCESS orders. We accept official orders from Government and Educational establishments. We accept VAT FREE EXPORT orders, please phone or write to check shipping cost.

OPENING HOURS: MON-FRI 9am-5.30pm, SAT 10am-2pm.

We welcome callers, no parking problems.

Access

VISA

B. Dorricot

GENIE UTILITIES

Machine code offers speed and facilities beyond BASIC, but a solitary USR function is a trifle limiting. Extend your capabilities on the Genie with this short routine.

Have you ever wished for more than the solitary user machine code function call? Ever wished for two or more arguments instead of one? Ever envied the BBC Micro's VDU command for fast and easy output of ASCII code characters to the display? Ever wanted to convert from hexadecimal to denary, octal to denary or binary to denary? This simple machine code utility program will solve all these problems in only 334 bytes.

USER MACHINE CODE

The normal BASIC command is:

k = USR (y)

where y is an integer and the address of the routine has been POKEd into addresses 16526 and 16527. This utility simplifies this rigmarole to the simple command:

DEF USR n = address

where n is the number of the routine in the range 0 to 9 inclusive and the address can be any legal address in the processor's 64K of memory (it could even call a ROM routine directly). These routines could be called up by the function:

x = USRn (y)

Again n is a number in the range 0 to 9 inclusive and corresponds to n above.

The first argument must be the single integer found in the brackets of the USR (y) function. This can be obtained from BASIC by calling a ROM routine located at 0A9A hex or 2714 decimal. The integer will be returned in the HL registers as 16 bits.

Further arguments may be appended to the USRn (y) routine in BASIC by a line such as:

x = USRn (y), a$, z, t$

These arguments may be 16-bit integer, string or even single or double precision floating point and may be mixed in any order. Any calculations may also be included in these arguments, ie string operators such as MID$, LEFT$ and CHR$ and arithmetical operators such as *, /, ABS, EXP, SIN and so on. The important thing to remember is that a call for a string or number from the user machine code routine must find a string or number in the corresponding position after the USRn (y) statement.

To acquire these arguments the utility supplies two routines called GETNUM and GETSTR. GETNUM will get a 16-bit binary number into the HL registers and GETSTR will set the address of the string into the HL registers and its length into the B register. In the user's machine code routine, calls to GETNUM and GETSTR should take place at the beginning of the code and the results stored for any calculations that are required. If any items are saved on the stack by the user machine code before any of the subroutines are called, the result will be unpredictable but will always crash the system.

The machine code routine can return any of the four types of variable:

Integer Simply set the 16-bit binary number into the HL register pair and jump to location 0A9A hex or 2714 decimal at the end of the routine.

String Any length up to 255 characters may be returned by jumping to the utility routine PUTSTR. On entry to the routine, the HL registers equal the actual address of the string in memory and register B is the length of the string.

Single and double precision numbers A single precision number in Video Genie BASIC requires four bytes and a double precision number eight bytes. This byte number, either 4 or 8, must be loaded to memory location 40AF hex or 16559 decimal. The actual number should be loaded somewhere in a table at 41D1 hex or 16669 decimal and the subroutine should return to BASIC with a RET instruction.

PUT IT THERE

The BBC Micro has the command:

VDU x, y, . . . , z

which will output the ASCII codes of the numbers following (ie x, y, . . . z) to the display. To mimic this command the utility uses PUT instead of VDU, if:

PUT x, y, . . . , z

will output the characters x, y, . . . , z to the display. Characters can also be output to the printer with this utility by inserting the code #2, #1 being the video display and also the default output device. For example:

PUT #2, 80, 85, 84, 13

will output the word "PUT" to the printer followed by a carriage return character.

THE CONVERSION ROUTINE

The routine in the utility will convert hexadecimal (base 16), octal (base 8) and binary (base 2) to denary (base 10). The conversion is activated by the commands:

&Hx for hexadecimal
&Ox for octal

and &Bx for binary

where the number following is in the base specified. If an incorrect character is found in the string (eg 3 for binary) then the computer will terminate the conversion at that point.

This routine will allow a statement such as

DEF USR0 = &H7F00

to set up routine number 0 to call at location 7F00 hex or 32312 decimal. It may be executed by:

X = USR0(Y)

and the computer will jump to and execute the code at 7F00 hex. Note that this command could cause the system to crash if no user machine code routine has been POKEd into the memory from 7F00 hex.

THE UTILITY PROGRAM

Listing 1 is the utility program itself. It is located at 0F000 hex or 61440 decimal to allow the user's machine code programs to be assembled at or above 0F200 hex or 61952 decimal, thus protecting them from the BASIC interpreter. It could easily be reassembled to another location. The START routine 'plugs' all the machine code extensions to the BASIC interpreter and also protects itself. This it does by placing its start address minus one (0F7FF hex) into two memory locations, 40B1 hex (16561 decimal) and 40D6 hex (16598 decimal). The amount of string space is reset to 100 bytes (top of BASIC RAM top minus 100) and is put in location 40A0 hex (16544 decimal). The screen is cleared and the initialise routine returns to BASIC. The rest of the utility is well documented and uses BASIC ROM routines which are explained in full. A BASIC progam in memory when the utility program is inititalised is not destroyed but its variables are.

If you crash the system and the utility is not destroyed, you may re-initialise it by typing SYSTEM then /61440.

Listing 2 is a BASIC program to load the utility into memory. It should be initialised, once RUN, by typing SYSTEM and then /61440. The last five-digit number in line 600 acts as a checksum, so if you make a mistake entering the data this will detect it.

Listing 1. The Genie utility program.

```
                00100 ;    ****************************
                00110 ;    **      BASIC UTILITY       **
                00120 ;    **  WRITTEN BY              **
                00130 ;    **          B. DORRICOTT    **
                00140 ;    ****************************
                00150 ;
                00160 ; ADDS THE FOLLOWING COMMANDS TO BASIC:
                00170 ;       DEF USR N = ADDRESS
                00180 ;       X = USR N ( Y )
                00190 ;       X = USR N ( Y ), X$, Z, M$
                00200 ;       PUT #N, Z, Y, ... Z
                00210 ;       &H X    - HEXADECIMAL
                00220 ;       &O Y    - OCTAL              CONVERSIONS
                00230 ;       &B Z    - BINARY
                00240 ;
                00250 ; AND ADDS THREE MACHINE CODE ROUTINES FOR THE
                00260 ; DISPOSAL OF THE USER:
                00270 ; GETNUM          - GET A NUMBER FOLLOWING THE USR
                00280 ;                   STATEMENT INTO THE HL REGISTER PAIR.
                00290 ;                   DESTROYS ALL REGISTERS (INCLUDING ALTERNAT
E)
                00300 ;                   STACK MUST NOT BE ALTERED.  INVOKED BY
                00310 ;                   THE COMMAND:
                00315 ;                         CALL GETNUM
                00320 ; GETSTR          - GET THE STRING FOLLOWING THE USR
                00330 ;                   STATEMENT INTO THE HL REGISTER PAIR AND
                00340 ;                   ITS LENGTH INTO THE B REGISTER.
                00350 ;                   DESTROYS ALL REGISTERS. STACK MUST
                00360 ;                   NOT BE ALTERED.  INVOKED BY:
                00370 ;                         CALL GETSTR.
                00380 ; PUTSTR          - RETURN THE STRING POINTED TO BY HL
                00390 ;                   WITH THE LENGTH IN B TO BASIC.
                00400 ;                   INVOKED BY:
                00410 ;                         JP PUTSTR.
                00420 ;
                00430 ; FOLLOWING ARE BASIC ROUTINES/FLAGS USED BY THIS
                00440 ; UTILITY WITH AN EXPLANATION OF THIER FUNCTION.
                00450 ;
                00460 ;
 19A2           00470 ERROR   EQU     19A2H   ;ERROR ROUTINE OF BASIC.
                00480                         ;ENTER WITH ERROR NUMBER
                00490                         ;(N-1)*2 IN THE E REGISTER.
                00500                         ; IF E=8 THEN ILLEGAL FUNCTION
                00510                         ; CALL ERROR
 1997           00520 SYNTAX  EQU     1997H   ;A SPECIAL JUMP TO CAUSE
                00530                         ;A SYNTAX ERROR.
 408E           00540 STORE   EQU     408EH   ;WHERE THE ADDRESS OF THE
                00550                         ;MACHINE CODE ROUTINE WAS
                00560                         ;STORED IN BASIC. (16526 DECIMAL)
 2B01           00570 GETNO   EQU     2B01H   ;SOLVE THE EQUATION IN PROGRAM
                00580                         ;AT THE HL REGISTER AND
                00590                         ;RETURN: HL = END OF ROUTINE:
                00600                         ;DE = 16 BIT BINARY NUMBER.
                00610                         ;IF THE NUMBER IS OUT OF RANGE
                00620                         ;RANGE AN OVERFLOW ERROR
                00630                         ;IS PRODUCED BY THIS ROUTINE.
 40AF           00650 NTF     EQU     40AFH   ;NUMBER TYPE FLAG
                00660                         ;SET UP SO BASIC KNOWS WHAT TO
                00670                         ;EXPECT.
                00680                         ;NTF = 2 = INTEGER
                00690                         ;      3 = STRING
                00700                         ;      4 = SINGLE PRECISION
                00710                         ;      8 = DOUBLE PRECISION
 409C           00720 ODF     EQU     409CH   ;THIS IS THE OUTPUT DEVICE FLAG.
                00730                         ; = 0 = VDU
                00740                         ; = 1 = PRINTER
                00750                         ; =-1 = CASSETTE
 2B1C           00760 GETA    EQU     2B1CH   ;GET AN 8-BUT NUMBER FROM THE
                00770                         ;EXPRESSION POINTED TO BY HL
                00780                         ;INTO THE A REGISTER.  IF OUT
                00790                         ;OF RANGE (IE LESS THAN 0 OR
                00800                         ;GREATER THAN 255) THEN CREATE
                00810                         ;ERROR AND RETURN TO BASIC. AT THE
                00820                         ;END OF THE ROUTINE HL POINTS
                00830                         ;TO THE END OF THE EXPRESSION.
 032A           00840 DEVICE  EQU     032AH   ;OUTPUT THE ASCII CHARACTER IN
                00850                         ;THE A REGISTER TO THE DEVICE
                00860                         ;SPECIFIED BY THE ODF FLAG.
 4121           00870 ARITH   EQU     4121H   ;PLACE IN THE BASIC ARITHMETIC
                00880                         ;TABLE WHERE A POINTER TO THE
                00890                         ;ADDRESS OF STRING DATA OR A
                00900                         ;16-BIT BINARY NUMBER IS
                00910                         ;STORED.
 2337           00920 FIND    EQU     2337H   ;FIND THE STRING ROUTINE.
                00930                         ;ENTER: BC = HL = PLACE OF THE
                00940                         ;EXPRESSION (IE "HELLO" OR
                00950                         ;"HI"+B$). EXIT. HL = END OF THE
                00960                         ;EXPRESSION AND STRING POINTER
                00970                         ;SET AT 4121H:
                00980                         ;(4121H)+0: LENGTH OF STRING
                00990                         ;         +1: LSB AND
                01000                         ;         +2: MSB OF THE ACTUAL
                01010                         ;ADDRESS OF THE STRING IN MEMORY.
                01020 ;
                01030 ;
 F000           01040         ORG     0F000H  ;THIS ADDRESS MAY BE ALTERED.
                01050 ;
                01060 ;
                01070 ; ROUTINE TO 'PLUG' INTO BASIC.
                01080 ;
                01090 ;
 F000 2A5C41    01100 START   LD      HL,(415CH);GET RETURN ON ERROR ADDRESS.
 F003 223DF0    01110         LD      (INST+1),HL;FOR UNKNOWN CHARACTER AFTER DEF
 F006 2139F0    01120         LD      HL,DEFINE
 F009 225C41    01130         LD      (415CH),HL;NOW DEFINE ROUTINE PLUGGED IN.
 F00C 215FF0    01140         LD      HL,USR  ;USR CALCULATION ROUTINE
                01150                         ;INSERT NOW.
 F00F 22AA41    01160         LD      (41AAH),HL
 F012 3EC3      01170         LD      A,0C3H  ;JUMP OP-CODE
 F014 32A941    01180         LD      (41A9H),A;COMPLETE PLUG IN
 F017 2179F0    01190         LD      HL,PUT  ;SORT OUT ENTRY TO THE PUT
 F01A 228341    01200         LD      (4183H),HL;FROM BASIC
 F01D 21E8F0    01210         LD      HL,CONVER;ALSO DO THE SAME FOR THE
 F020 229541    01220         LD      (4195H),HL;CONVERSION ROUTINE.
 F023 21FFEF    01230         LD      HL,START-1;NOW PROTECT THIS UTILITY
 F026 22B140    01240         LD      (40B1H),HL;
 F029 22D640    01250         LD      (40D6H),HL
 F02C 119CFF    01260         LD      DE,-100 ;AMOUNT OF STRING SPACE TO SAVE
 F02F 19        01270         ADD     HL,DE
 F030 22A040    01280         LD      (40A0H),HL
 F033 CDC901    01290         CALL    01C9H   ;CLEAR SCREEN
 F036 C3CC06    01300         JP      06CCH   ;RETURN TO BASIC
                01310 ;
                01320 ;
                01330 ; DEFINE ROUTINE FOR THE DEF USRN(Y)=ADDRESS
                01340 ; ROUTINE
                01350 ;
                01360 ;
 F039 FEC1      01370 DEFINE  CP      193     ;193 = CODE FOR "USR"
                01380                         ;IF NOT USR CODE THEN RETURN
 F03B C20000    01390 INST    JP      NZ,0    ;DUMMY VALUE TO BE FILLED
                01400                         ;IN BY THE START ROUTINE
 F03E D7        01410         RST     16      ;GET NEXT CHARACTER.
                01420                         ;TO SEE IF CHARACTER IS NUMERIC.
 F03F 1E08      01430         LD      E,8     ;ILLEGAL FUNCTION CALL IF NOT
                01440                         ;NUMERIC.
 F041 D2A219    01450         JP      NC,ERROR;RETURN TO BASIC WITH ERROR
 F044 D630      01460         SUB     48      ;GET THE ASCII CODE OF THE
                01470                         ;NUMBER INTO THE RANGE
                01480                         ;0 =( A =( 9.
 F046 08        01490         EX      AF,AF'  ;SAVE FOR LATER.
 F047 D7        01500         RST     16      ;GET NEXT CHARACTER INTO A.
 F048 FED5      01510         CP      213     ;TEST FOR EQUALS SIGN.
 F04A C29719    01520         JP      NZ,SYNTAX;SYNTAX ERROR IF NOT EQUALS
                01530                         ;SIGN.
 F04D CD012B    01540         CALL    GETNO   ;GET THE ADDRESS OF THE
                01550                         ;ROUTINE INTO DE.
 F050 E5        01560         PUSH    HL      ;SAVE POSITION IN BASIC PROGRAM
                01570                         ;BEING DECODED.
 F051 08        01580         EX      AF,AF'  ;GET BACK ORIGINAL A.
 F052 87        01590         ADD     A,A     ; A = A * 2
                01600                         ;EACH ADDRESS IS TWO BYTES LONG.
 F053 4F        01610         LD      C,A     ;GET READY FOR ADD TO FIND
 F054 0600      01620         LD      B,0     ;PLACE IN TABLE
 F056 2151F1    01630         LD      HL,TABLE;TABLE OF USR ROUTINE ADDRESSES.
 F059 09        01640         ADD     HL,BC   ;HL NOW POINTS TO WHERE TO
                01650                         ;STORE THE USER SUBROUTINE
                01660                         ;ADDRESS.
 F05A 73        01670         LD      (HL),E  ;SAVE LSB IN TABLE.
 F05B 23        01680         INC     HL
 F05C 72        01690         LD      (HL),D  ;SAVE MSB IN TABLE.
 F05D E1        01700         POP     HL      ;RETURN POSITION IN BASIC
                01710                         ;PROGRAM.
 F05E C9        01720         RET             ;RETURN TO BASIC NOW DONE.
                01730 ;
                01740 ;
                01750 ; THIS IS THE USR ROUTINE PLUGGED INTO BASIC
                01760 ;
                01770 ;
 F05F D7        01780 USR     RST     16      ;GET NEXT CHARACTER WHICH IS
                01790                         ;THE SUBROUTINE NUMBER.
 F060 1E08      01800         LD      E,8     ;ILLEGAL FUNCTION CALL IF THE
                01810                         ;CHARACTER WAS NOT A DIGIT.
 F062 D2A219    01820         JP      NC,ERROR;GO IF ERROR (IE C NOT SET)
 F065 D630      01830         SUB     48      ;GET ASCII CODE IN A REGISTER
                01840                         ;INTO RANGE 0 =( A =( 9
 F067 87        01850         ADD     A,A     ;A = A * 2
                01860                         ;EACH ENTRY IS TWO BYTES.
 F068 5F        01870         LD      E,A     ;FOR CALCULATION TO FIND ADDRESS
 F069 1600      01880         LD      D,0     ;OF USER ROUTINE.
 F06B E5        01890         PUSH    HL      ;SAVE POSITION IN THE PROGRAM
                01900                         ;FOR BASIC.
 F06C 2151F1    01910         LD      HL,TABLE;TABLE OF SUBROUTINE ADDRESSES.
 F06F 19        01920         ADD     HL,DE   ;NOW HL IS POINTING TO THE LSB
                01930                         ;OF THE ADDRESS OF THE ROUTINE
                01940                         ;REQUIRED.
 F070 5E        01950         LD      E,(HL)  ;GET THE ADDRESS INTO DE
 F071 23        01960         INC     HL
 F072 56        01970         LD      D,(HL)  ;MSB LAST
 F073 ED538E40  01980         LD      (408EH),DE;PUT THE ADDRESS HERE FOR
                01990                         ;THE BASIC ROUTINE TO PICK UP
 F077 E1        02000         POP     HL      ;RETURN POSITION IN PROGRAM
 F078 C9        02010         RET             ;RETURN TO BASIC.
                02020 ;
                02030 ;
                02040 ; THIS IS THE PUT ROUTINE TO OUTPUT THE
                02050 ; ASCII CODE OF CHARACTERS FOLLOWING THE
                02060 ; COMMAND NAME TO VDU.
                02070 ;
 F079 FE23      02080 PUT     CP      '#'     ;TEST IF THE USER HAS SPECIFIED
                02090                         ;THE OUTPUT DEVICE.
 F07B 200E      02100         JR      NZ,PUT1 ;IF NOT SPECIFIED THE DO DEFULT
                02110                         ;TO VIDEO DISPLAY.
 F07D D7        02120         RST     16      ;GET NEXT CHARACTER TO THE A
                02130                         ;REGISTER
 F07E FE31      02140         CP      '1'     ;TEST IF VDU WANTED
 F080 280A      02150         JR      Z,PUT2  ;JUMP IF WAS VDU OUTPUT
 F082 FE32      02160         CP      '2'     ;TEST IF OUTPUT TO PRINTER
 F084 2808      02170         JR      Z,PUT2  ;JUMP IF PRINTER
 F086 1E08      02180         LD      E,8     ;THIS IS AN ILLEGAL FUNCTION CALL
 F088 C3A219    02190         JP      ERROR   ;DO ERROR.
                02200 ;
 F08B 2B        02210 PUT1    DEC     HL      ;RETURN FOR COMMA ROUTINE
 F08C AF        02220 PUT2    XOR     A       ;A = 0 = OUTPUT TO VIDEO
 F08D 01        02230         DEFB    1       ;THIS IS THE CODE FOR
                02240                         ;LD BC,NNNN AND SAVES A BYTE
 F08E 3E01      02250 PUT3    LD      A,1     ;A = 1 = PRINTER OUTPUT
                02260                         ;IF PUT2 WAS EXECUTED THEN THESE
                02270                         ;TWO BYTES WILL BE LOADED INTO
                02280                         ;THE BC REGISTERS AND WILL NOT
                02290                         ;AFFECT THE CONTENT OF THE A
                02300                         ;REGISTER.
 F090 329CA0    02310         LD      (ODF),A ;THIS IS THE BASIC OUTPUT
                02320                         ;TYPE FLAG.
 F093 D7        02330 PUT4    RST     16      ;GET THE NEXT CHARACTE
 F094 FE2C      02340         CP      ','     ;TEST IF A COMMA FOLLOWING
 F096 28FB      02350         JR      Z,PUT4  ;IF IT IS THEN SKIP IT.
 F098 CD1C2B    02360         CALL    GETA    ;THIS IS A BASIC ROUTINE THAT
                02370                         ;GETS A NUMBER OF 8-BITS INTO
                02380                         ;THE A REGISTER.
 F09B CD2A03    02390         CALL    DEVICE  ;THIS IS THE BASIC ROUTINE TO
                02400                         ;OUTPUT A BYTE TO THE OUTPUT
                02410                         ;DEVICE SPECIFIED AT THE
                02420                         ;MEMORY LOCATION 409CH.
 F09E 7E        02430         LD      A,(HL)  ;GET THE CHARACTER.
 F09F FE2C      02440         CP      ','     ;ANY MORE ARGUMENTS?
 F0A1 28F0      02450         JR      Z,PUT4  ;GO FOR NEXT IF THERE ARE.
 F0A3 C9        02460         RET             ;RETURN TO BASIC NOW.
                02470 ;
                02480 ;
                02490 ; THIS IS THE ROUTINE TO BE CALLED BY A ROUTINE
                02500 ; TO GET A STRING FROM BASIC.
                02510 ; THE STRING SHOULD FOLLOW THE USR COMMAND:
                02520 ;         X = USR 0 ( Y ) , X$ , Y$ , .... , Z$
                02530 ; (THERE MAY BE NUMBERS INTERMIXED).
                02540 ;
 F0A4 C1        02550 GETSTR  POP     BC      ;GET RETURN ADDRESS AND SAVE
 F0A5 D1        02560         POP     DE      ;GET RETURN ADDRESS FOR BASIC
 F0A6 E1        02570         POP     HL      ;GET PLACE IN PROGRAM
 F0A7 D5        02580         PUSH    DE      ;SAVE RETURN ADDRESS FOR BASIC
 F0A8 C5        02590         PUSH    BC      ;SAVE RETURN ADDRESS ON STACK
 F0A9 D7        02600 GETST1  RST     16      ;GET NEXT CHARACTER IN PROGRAM
 F0AA FE2C      02610         CP      ','     ;IF IT IS A COMMA THEN TRY AGAIN
 F0AC 28FB      02620         JR      Z,GETST1
 F0AE 44        02630         LD      B,H     ;GET HL INTO THE REGISTERS
 F0AF 4D        02640         LD      C,L     ;BC FOR THE BASIC ROUTINE
 F0B0 CD3723    02650         CALL    FIND    ;THIS WILL GET THE STRING INTO
                02660                         ;MEMORY LOCATIONS 4121H.
 F0B3 C1        02670         POP     BC      ;NOW RESTORE POSITION IN
 F0B4 D1        02680         POP     DE      ;GET RETURN ADDRESS FOR BASIC
 F0B5 E5        02690         PUSH    HL      ;BACK ON STACK FOLLOWED BY
 F0B6 D5        02700         PUSH    DE      ;PUT BACK RETURN ADDRESS FOR BASIC
 F0B7 C5        02710         PUSH    BC      ;THE RETURN ADDRESS OF THE
                02720                         ;BASIC.
 F0B8 2A2141    02730         LD      HL,(ARITH);THIS IS WHERE THE ADDRESS
                02740                         ;OF THE STRING POINTER IS.
 F0BB 46        02750         LD      B,(HL)  ;FIRST IT POINTS TO THE LENGTH
                02760                         ;OF THE STRING
 F0BC 23        02770         INC     HL
 F0BD 5E        02780         LD      E,(HL)  ;THEN THE LSB OF THE ACTUAL
                02790                         ;ADDRESS OF THE STRING
 F0BE 23        02800         INC     HL
 F0BF 56        02810         LD      D,(HL)  ;THEN THE MSB OF THE ADDRESS
                02820                         ;OF THE STRING.
 F0C0 EB        02830         EX      DE,HL   ;GET THIS ADDRESS INTO HL
                02840                         ;FOR USER.
 F0C1 C9        02850         RET             ;AND RETURN
                02860 ;
                02870 ;
                02880 ; THIS ROUTINE WILL RETURN A STRING TO BASIC.
                02890 ; THE LENGTH IS IN B AND THE ADDRESS OF THE
                02900 ; STRING IN THE HL REGISTER.
                02910 ;
                02920 ;
 F0C2 3E03      02930 PUTSTR  LD      A,3     ;THIS IS TO INDICATE A STRING
 F0C4 32AF40    02940         LD      (NTF),A ;SAVE IT IN THE NUMBER TYPE
                02950                         ;FLAG FOR BASIC.
 F0C7 EB        02960         EX      DE,HL   ;SAVE HL FOR LATER
 F0C8 214EF1    02970         LD      HL,TAB  ;SPECIAL TABLE FOR THIS ROUTINE
 F0CB 222141    02980         LD      (ARITH),HL;THIS IS WHERE THE TABLE FOR
                02990                         ;THE POSITION OF THE STRING
                03000                         ;IS IN MEMORY.
 F0CE 70        03010         LD      (HL),B  ;LOAD FIRST WITH LENGTH OF
                03020                         ;THE STRING.
```

```
F0CF 23         03030           INC     HL
F0D0 73         03040           LD      (HL),E  ;LOAD THE SECOND WITH THE LSB
                03050                           ;OF THE STRING
F0D1 23         03060           INC     HL
F0D2 72         03070           LD      (HL),D  ;AND LAST WITH THE MSB OF THE
                03080                           ;STRING.
F0D3 C9         03090           RET
                03100 ;
                03110 ;
                03120 ; GET A NUMBER FROM BASIC AFTER THE USRN(Y)
                03130 ; STATEMENT AND LOAD THE VALUE INTO THE HL
                03140 ; REGISTER.  NB THE STACK SHOULD NOT BE ALTERED
                03150 ; BY THE USER ROUTINE.
                03160 ;
                03170 ;
F0D4 D9         03180 GETNUM    EXX             ;SAVE THE RETURN ADDRESSES IN
                03190                           ;THE ALTERNATE REGISTER SET AS
                03200                           ;THEY ARE NOT USED BY BASIC.
F0D5 C1         03210           POP     BC      ;ADDRESS TO RETURN TO AS GIVEN
                03220                           ;BY THE USER CALL STATEMENT.
F0D6 D1         03230           POP     DE      ;ADDRESS FOR USER ROUTINE TO
                03240                           ;RETURN TO GIVEN BY BASIC.
F0D7 D9         03250           EXX             ;SAVE
F0D8 E1         03260           POP     HL      ;GET PLACE IN BASIC INTO HL
                03270                           ;REGISTER
F0D9 D7         03280 GETN1     RST     16      ;GET NEXT CHARACTER IN A
F0DA FE2C       03290           CP      ','     ;IF COMMA THEN TRY AGAIN
F0DC 28FB       03300           JR      Z,GETN1
F0DE CD0022B    03310           CALL    GETNO+1 ;NOW DECODE THE NUMBER INTO DE.
F0E1 E5         03320           PUSH    HL      ;SAVE WHERE WE ARE IN BASIC
                03330                           ;PROGRAM.
F0E2 D9         03340           EXX             ;NOW GET BACK THE TWO RETURN
                03350                           ;ADDRESSES
F0E3 D5         03360           PUSH    DE      ;BASIC RETURN ADDRESS
F0E4 C5         03370           PUSH    BC      ;USER RETURN ADDRESS
F0E5 D9         03380           EXX             ;GET BACK NORMAL REGISTERS
F0E6 EB         03390           EX      DE,HL   ;NOW GET NUMBER INTO HL
                03400                           ;REGISTER FOR USER.
F0E7 C9         03410           RET             ;RETURN
                03420 ;
                03430 ;
                03440 ; FOLLOWING IS THE HEXADECIMAL, OCTAL OR BINARY TO
                03450 ; DECIMAL CONVERSION ROUTINE.
                03460 ;
                03470 ;
F0E8 D7         03480 CONVER    RST     16      ;GET NEXT BYTE IN THE STRING
                03490                           ;INTO A REGISTER.
F0E9 110000     03500           LD      DE,0    ;CONVERSION WILL BE PUT INTO
                03510                           ;THIS REGISTER PAIR.
F0EC FE48       03520           CP      'H'     ;SEE IF HEXADECIMAL CONVERSION
F0EE 280D       03530           JR      Z,HEX   ;WANTED. IF SO JUMP
F0F0 FE4F       03540           CP      'O'     ;SEE IF OCTAL CONVERSION
F0F2 2828       03550           JR      Z,OCTAL ;WANTED. IF SO JUMP
F0F4 FE42       03560           CP      'B'     ;SEE IF BINARY CONVERSION
F0F6 2838       03570           JR      Z,BINARY;WANTED. IF SO JUMP.
F0F8 1E08       03580           LD      E,8     ;GENERATE AN ILLEGAL FUNCTION
F0FA C3A219     03590           JP      ERROR   ;CALL IF NOT ONE OF ABOVE.
                03600 ;
                03610 ; THIS IS THE HEXADECIMAL CONVERSION ROUTINE.
                03620 ; IT WILL CONVERT ALL ASCII UNTIL A NON-HEXADECIMAL
                03630 ; CHARACTER IS FOUND, WHEREUPON IT WILL JUMP
                03640 ; TO THE TIDY UP ROUTINE.
                03650 ;
F0FD D7         03660 HEX       RST     16      ;GET A CHARACTER
F0FE D630       03670           SUB     '0'     ;SET UP FOR RANGE 0 TO 9
F100 3841       03680           JR      C,NUM1  ;IF LESS THAN THE CHARACTER
                03690                           ;'0' THEN MUST BE END OF NUMBER
                03700                           ;SO JUMP AWAY.
F102 FE0A       03710           CP      10      ;TEST IF IN LEGAL LIMIT OF
                03720                           ;0 TO 9
F104 3806       03730           JR      C,HEX1  ;JUMP IF IT IS.
F106 D611       03740           SUB     17      ;NOW TEST FOR DIGITS A-F
F108 3839       03750           JR      C,NUM1  ;IF TOO LOW THEN INCORRECT
                03760                           ;SO JUMP AWAY
F10A C60A       03770           ADD     A,10    ;ALLIGN TO RIGHT RANGE AGAIN
F10C FE10       03780 HEX1      CP      16      ;TEST IF TOO LARGE
F10E 3033       03790           JR      NC,NUM1 ;GO IF IT IS.
F110 4F         03800           LD      C,A     ;GET THE CHARACTER INTO
F111 0600       03810           LD      B,0     ;THE BC REGISTER FOR THE ADDITION
                03820                           ;INTO HL.
F113 EB         03830           EX      DE,HL   ;EXCHANGE SO HL IS NOW THE NUMBER
                03840                           ;REACHED SO FAR.
F114 29         03850           ADD     HL,HL   ;HL = HL * 2
F115 29         03860           ADD     HL,HL   ;HL = HL * 4
F116 29         03870           ADD     HL,HL   ;HL = HL * 8
F117 29         03880           ADD     HL,HL   ;HL = HL * 16
F118 09         03890           ADD     HL,BC   ;ADD IN NEW NUMBER
F119 EB         03900           EX      DE,HL   ;GET BACK PLACE IN PROGRAM AND
                03910                           ;SAVE NUMBER REACHED SO FAR.
F11A 18E1       03920           JR      HEX     ;GO - GO NEXT CHARACTER.
                03930 ;
                03940 ; THIS ROUTINE IS THE OCTAL TO DENARY CONVERSION
                03950 ; ROUTINE WHICH TAKES THE ASCII AND CONVERTS IT INTO
                03960 ; BINARY IN DE.
                03970 ;
F11C D7         03980 OCTAL     RST     16      ;GET THE NEXT CHARACTER.
F11D FE38       03990           CP      '8'     ;TEST - IS IT OVER?
F11F 3022       04000           JR      NC,NUM1 ;YES - END OF NUMBER THEN - JUMP.
F121 D630       04010           SUB     '0'     ;GET INTO RANGE 0 - 7.
F123 381E       04020           JR      C,NUM1  ;IF TOO SMALL THEN END OF NUMBER.
F125 4F         04030           LD      C,A     ;GET READY TO ADD INTO THE DE
F126 0600       04040           LD      B,0     ;REGISTER.
F128 EB         04050           EX      DE,HL   ;GET NUMBER SO FAR AND SAVE PLACE
                04060                           ;IN THE PROGRAM.
F129 29         04070           ADD     HL,HL   ; HL = HL * 2
F12A 29         04080           ADD     HL,HL   ; HL = HL * 4
F12B 29         04090           ADD     HL,HL   ; HL = HL * 8
F12C 09         04100           ADD     HL,BC   ;ADD IN NEW NUMBER.
F12D EB         04110           EX      DE,HL   ;GET PLACE IN PROGRAM
F12E 18EC       04120           JR      OCTAL   ;GO FOR NEXT CHARACTER
                04130 ;
                04140 ; BINARY TO DENARY ROUTINE.  WILL ACCEPT ONLY THE
                04150 ; ASCII '0' AND '1' AS LEGAL AND WILL END NUMBER
                04160 ; DECODING IF THE ASCII IS OUT OF THE RANGE.
                04170 ;
F130 D7         04180 BINARY    RST     16      ;GET NEXT CHARACTER
F131 FE30       04190           CP      '0'     ;MUST BE DE = DE + 2 ONLY SO JUMP
F133 2805       04200           JR      Z,BIN1
F135 FE31       04210           CP      '1'     ;MUST BE DE = DE + 2 + 2
F137 2006       04220           JR      NZ,BIN2 ;JUMP IF NOT AS END OF NUMBER.
F139 13         04230           INC     DE
F13A EB         04240 BIN1      EX      DE,HL   ;NOW DO DE = DE + 2
F13B 29         04250           ADD     HL,HL
F13C EB         04260           EX      DE,HL
F13D 18F1       04270           JR      BINARY  ;GO FOR NEXT CHARACTER
F13F CB0A       04280 BIN2      SRL     D       ;DIVIDE DE BY TWO TO GET IT
F141 CB1B       04290           RR      E       ;INTO THE RIGHT RANGE.
                04300 ;
                04310 ; THIS IS THE TIDY UP SECTION OF THE CONVERSION ROUTINE.
                04320 ;
F143 3E02       04330 NUM1      LD      A,2     ;THE CONVERSIONS ARE INTEGER ONLY
F145 32AF40     04340           LD      (VTF),A ;TELL THIS TO BASIC INTERPRETER.
F148 ED5B2141   04350           LD      (ARITH),DE;GIVE THE NUMBER TO BASIC AS
                04360                           ;WELL.
F14C 7E         04370           LD      A,(HL)  ;GET THE CHARACTER FOR BASIC
                04380                           ;AS GENERATES ERROR WITHOUT IT.
F14D C9         04390           RET
                04400 ;
                04410 ;
F14E 00         04420 TAB       DEFB    0       ;THE LENGTH OF THE STRING WILL
                04430                           ;GO HERE.
F14F 0000       04440           DEFW    0       ;WHERE STRING IS WILL GO HERE.
                04450 ;
                04460 ; TABLE OF TEN ADDRESSES OF ROUTINES.
                04470 ; PRESET TO JUMP TO THE ERROR ROUTINE IF CALLED
                04480 ; IN ERROR
                04490 ;
                04500 ;
F151 9719       04510 TABLE     DEFW    SYNTAX
F153 9719       04520           DEFW    SYNTAX
F155 9719       04530           DEFW    SYNTAX
F157 9719       04540           DEFW    SYNTAX
F159 9719       04550           DEFW    SYNTAX
F15B 9719       04560           DEFW    SYNTAX
F15D 9719       04570           DEFW    SYNTAX
F15F 9719       04580           DEFW    SYNTAX
F161 9719       04590           DEFW    SYNTAX
F163 9719       04600           DEFW    SYNTAX
                04610 ;
F000            04620           END     START
00000 TOTAL ERRORS
```

Listing 2. The corresponding BASIC program to load the utility into memory.

```
10 REM - Program two - Load utility into memory
20 REM - (Written by B.Dorricott - 1983)
30 REM
40 FOR ADR=-4096 TO -3740
50 READ DAT
60 LET NUM=NUM+DAT
70 POKE ADR,DAT
80 NEXT ADR
90 READ CHECK:IF CHECK<>NUM THEN PRINT"Error in input data":END
100 CLS:PRINT"The utility has been loaded and checked."
110 PRINT"To initialise the utility enter the following:":PRINT
120 PRINT,"SYSTEM"
130 PRINT
140 PRINT,"*? /61440"
150 PRINT
160 END
200 DATA 042,092,065,034,060,240,033,057,240,034
210 DATA 092,065,033,095,240,034,170,065,062,195
220 DATA 050,169,065,033,121,240,034,131,065,033
230 DATA 232,240,074,149,065,033,255,239,034,177
240 DATA 064,034,214,064,017,156,255,025,034,160
250 DATA 064,205,201,001,195,204,006,254,193,194
260 DATA 045,001,215,030,008,210,162,025,214,048
270 DATA 008,215,254,213,194,151,025,205,001,043
280 DATA 229,008,135,079,006,000,033,081,241,009
290 DATA 115,035,114,225,201,215,030,008,210,162
300 DATA 025,214,048,135,095,022,000,229,033,081
310 DATA 241,025,094,035,086,237,083,142,064,225
320 DATA 201,254,035,032,014,215,254,049,040,010
330 DATA 254,050,040,000,030,008,195,162,025,042
340 DATA 175,001,062,001,050,156,064,215,254,044
350 DATA 040,251,205,028,047,205,042,003,126,254
360 DATA 044,040,240,201,193,209,225,213,197,215
370 DATA 254,044,040,251,068,077,205,055,035,193
380 DATA 209,229,213,197,042,033,065,070,035,094
390 DATA 035,086,235,201,062,003,050,175,064,235
400 DATA 033,078,241,034,033,065,112,035,115,035
410 DATA 114,201,217,193,209,217,225,215,254,044
420 DATA 040,251,205,002,043,229,217,213,197,217
430 DATA 235,201,215,017,000,000,254,072,040,013
440 DATA 254,079,040,040,254,066,040,056,030,008
450 DATA 195,162,025,215,214,048,056,065,254,010
460 DATA 056,006,214,017,056,057,198,010,254,016
470 DATA 048,051,079,006,000,235,041,041,041,041
480 DATA 009,235,024,225,215,254,056,048,034,214
490 DATA 048,056,030,079,006,000,235,041,041,041
500 DATA 009,235,024,206,215,254,048,040,005,254
510 DATA 049,002,006,019,235,041,275,024,241,003
520 DATA 058,203,027,062,002,050,175,064,227,065
530 DATA 033,065,126,201,000,000,000,151,025,151
540 DATA 025,151,025,151,025,151,025,151,025,151
550 DATA 025,151,025,151,025,151,025
600 DATA 39225:REM - Check number
```

Listing 3. The denary to hexadecimal conversion routine.

```
10 REM - Program three - Denary to Hexadecimal
20 REM - (Written by B.Dorricott - 1983)
30 REM
40 FOR ADR=(&HF200) TO (&HF22F)
50 READ DAT
60 LET NUM=NUM+DAT
70 POKE ADR,DAT
80 NEXT ADR
90 READ CHECK
100 IF CHECK<>NUM THEN PRINT"Error in input data":END
110 DEF USR0=&HF200:REM - Set up start of subroutine
120 CLS:PRINT"Denary to Hexadecimal conversion"
130 PRINT:PRINT:PRINT
140 INPUT"Enter the denary number ";DENARY
150 IF DENARY(-32768 OR DENARY)32767 THEN PRINT"Number out of ra
nge - Try again.":GOTO 140
160 PRINT "Hexadecimal is ";USR0(DENARY)
170 GOTO 130
500 DATA 205,127,010:' USER     CALL 007FH    :Get argument
510 DATA 017,048,242:'          LD DE,&PACE   :Where to put hex
520 DATA 124        :'          LD A,H        :Get MSB
530 DATA 205,022,242:'          CALL CONVER   :Do conversion MSB
540 DATA 125        :'          LD A,L        :Get LSB
550 DATA 205,022,242:'          CALL CONVER   :Do conversion LSB
560 DATA 033,048,242:'          LD HL,&PACE   :Get where HEX is
570 DATA 006,004    :'          LD B,4        :Length of HEX
580 DATA 195,194,240:'          JP PUTSTR     :Send to BASIC,return
590 DATA 079        :' CONVER   LD C,A        :Save for later
600 DATA 203,063    :'          SRL A         :Get MS nibble to
610 DATA 203,063    :'          SRL A         :to convert  first.
620 DATA 203,063    :'          SRL A
630 DATA 203,063    :'          SRL A
640 DATA 205,037,242:'          CALL TEST     :Convert and save
650 DATA 121        :'          LD A,C        :Get to convert LSN
660 DATA 230,015    :'          AND 15        :Set to 1. nibble.
670 DATA 198,048    :' TEST     ADD A,'0'     :First set ASCII no.
680 DATA 254,058    :'          CP 3AH        :Test if should A-F
690 DATA 056,002    :'          JR C,TEST1    :Go if not
700 DATA 198,007    :'          ADD A,7       :Now range A-F
710 DATA 018        :' TEST1    LD (DE),A     :Save in buffer
720 DATA 019        :'          INC DE        :Point to next
730 DATA 201        :'          RET
740                 :' &PACE    DEFM '        :Buffer of four
750 DATA 5850:REM - Check number
```

CHANGING BASES

Program three is a machine code routine to convert from 16-bit integer number to a four-digit hexadecimal string using the utility. Once the program has been run, entering USR0 (y) will act the same as the function HEX$ (y) found on some machines.

Type in, SAVE and RUN the program. After the few seconds it takes to load the machine code into memory at 0F200 hex, a decimal number will be requested. The hexadecimal of this number will then be printed out.

This routine will only be destroyed if the start address is changed or the machine is switched off. Hence you could type NEW and still be able to use this function.

THE INTERNATIONAL VIDEO GAME OF THE YEAR COMPETITION HERALDS THE "NEW AGE" IN VIDEO/COMPUTER GAMES

$175,000 TO BE WON

International Video Game of the Year

Create a brilliant, new video game and you could be on your way to becoming a millionaire. This fantastic competition, organised by I.R.P. (The International Register of Independent Computer Programmers Ltd) and the famous Mark McCormack International Management Group, offers programmers and inventors the opportunity of a life time. There are huge, immediate cash prizes and the on-going revenue of 10% of the sales of all games to distributors throughout the world, plus the chance to appear on an international TV show. Your skill and imagination could bring you fame and fortune!

$100,000 FIRST PRIZE! PLUS FIVE $15,000 RUNNER-UP PRIZES!

Devise a totally original new video game in one of these categories: SPORTS, SIMULATORS, ARCADE, STRATEGY, ADVENTURE/FANTASY or a special section which covers programmes that are not necessarily games but have outstanding Educational or Entertainment merit. We'll also be announcing a number of 'MERIT' awards which will be entitled to carry the message 'An International Video Game of the Year MERIT AWARD' on their retail packaging. It's a great challenge. And the rewards, both financially and in terms of prestige, are tremendous. This is the most exciting competition ever for creative computer and video enthusiasts.

YOU'RE A TV STAR TOO!

All six winning games and their inventors will be featured on an internationally distributed, spectacular TV special. That's going to make your name!

HOW TO ENTER

Just send in your game, or games, programmed on cassette for any popular home computer. Use the coupon, today, and we'll send you all the facts you need.

CLOSING DATE FOR ENTRIES IS 31st MAY 1984

To: IRP Limited, Pinewood Film Studios, Iver, Bucks, England.

Name .....

Address .....

.....

.....

.....CT1

Programmers: we offer you The World (which world is up to you!)

Argus Press Software (APS) is seeking to expand it's range of products into all areas of computer software, and we need talented programmers to help us. We already market the highly successful ASP range of adventure and strategy games and will be announcing new ranges in other fields in the near future.

APS is a member of the Argus Press group of Companies and is a sister company to Argus Specialist Publications, Europe's largest computer magazine publishers, (Computing Today, ZX Computing, Games Computing, Personal Computing Today, etc, etc). As such we can offer all the benefits of a large company whilst retaining the flexibility of a smaller, highly motivated unit.

We are looking for both product and people. If you have a piece of software for any of the popular micros that you think is marketable, send it to us at the address below. Our commission structure is second to none and our generation procedure is designed to offer programmers the maximum flexibility.

Alternatively - if you think you have the talent at the keyboard to help us realise our designs - write and tell us what you can do and upon which machine you can do it!

Write now for details to:-
Ron Harris
Argus Press Software
No 1 Golden Square
London W1R 3AB

LEARNING FORTH PART 5

In this month's article we look at input and output and how data may be stored using the FORTH RAM-disc.

The ease and flexibility with which information can be input and output within a program often gives a good indication of how 'usable' and 'friendly' a language may be. This month we will see how FORTH provides the building blocks for some powerful I/O (input-output) commands. We shall also see how information may be moved about between keyboard, screen, memory and disc storage, and discover a little more about how FORTH handles the disc. With the aid of a 'memory map' we shall also try to get our image of the complete FORTH system a little clearer and I shall explain a few of the oddities that I've been putting off till now!

A MAP OF THE MEMORY

Figure 1 is a diagrammatic representation of how the memory in the Spectrum micro is divided up when you are running Abersoft FORTH. The principles are the same for many FORTH implementations, but I shall deal specifically with this one.

This version of FORTH is loaded as a machine code progam which starts at memory location 24128. 'Below' that location (ie towards low memory) is the usual Spectrum ROM and system RAM including the video display RAM, printer buffer and so on. The details of these are given in Chapter 24 of the Spectrum manual.

The dictionary. Above this point (ie towards high-memory) is our FORTH system. The first section of the memory contains the dictionary of definitions of all the precompiled FORTH words and any you may have added yourself. The 'top' of this area can be found by executing the word HERE. This leaves on top of the stack the first free byte of memory above the dictionary. The dictionary expands as you add your own definitions or use words such as , and C, to add numbers to the dictionary.

The pad. At a fixed offset from the top of the dictionary is an area called the pad. This is a temporary 'scratch-pad' that can be used to hold text or numbers which we may want to move about in memory. We will deal with the use of the pad later in this article. The size of the pad is not fixed, as the distance between the pad and the parameter stack changes as definitions are compiled or while programs are running which put values on the stack. The position of the first byte of the pad can be found using the word PAD which leaves the address of the pad on the stack.

The parameter stack. Well above the pad is the parameter stack. This stack actually starts from a fixed position and as values are added the top of the stack moves downwards towards low memory. The starting point of the stack can be found using S0 @ where S0 is a system variable, described below.

The current memory location of the top of the stack is held in another system variable. The position of the top of the stack can be found using the word SP@.

Although I have described the stack as a push-down pop-up structure, the only thing that changes is the top of the stack and the stack pointer which tells you where the top is. As you can see from Fig. 2, when we put a number on the stack what really happens is that the pointer is decremented, so that it points to the next location lower in memory, then our number is stored where the pointer is pointing. When we remove a number from the stack, the number is fetched from the location where the pointer is pointing, then the pointer is decremented. Any numbers below our stack pointer (ie towards low memory) are meaningless.

As the stack pointer can be fetched using SP@ we could 'index into the stack', ie work out where, say, the tenth element was in memory and fetch that value, but this is not considered good practice. (The memory map is not identical for each machine and, besides your program being confusing, it would also not be transportable.)

The bottom of the stack is pointed to by a variable called S0 (pronounced S-zero).

Terminal input buffer. Above the stack is a region of memory which stores the text you enter from the keyboard. The starting address of this region is held in a variable called TIB. I will show later that we can use the fact that input from the keyboard is stored here to manipulate text.

Return stack. Not very far above the Terminal Input Buffer is the return stack. The return stack starts from a position contained in R0 (R-zero) and grows downwards towards low memory.

The return stack is used to hold the 'return address' of any word that is currently being executed. For example, if we had two definitions:

```
: BILL  CR  ." Hello Bill" ;
: FRED  BILL  ." I'm not Bill!" ;
```

when we execute FRED the first word in the definition is BILL. However before BILL is executed the FORTH address interpreter (which I have mentioned in a previous article) calculates the memory address of the next instruction in FRED and puts the value on the return stack. So when BILL is finished executing the address interpreter can use the value on the return stack to know where to

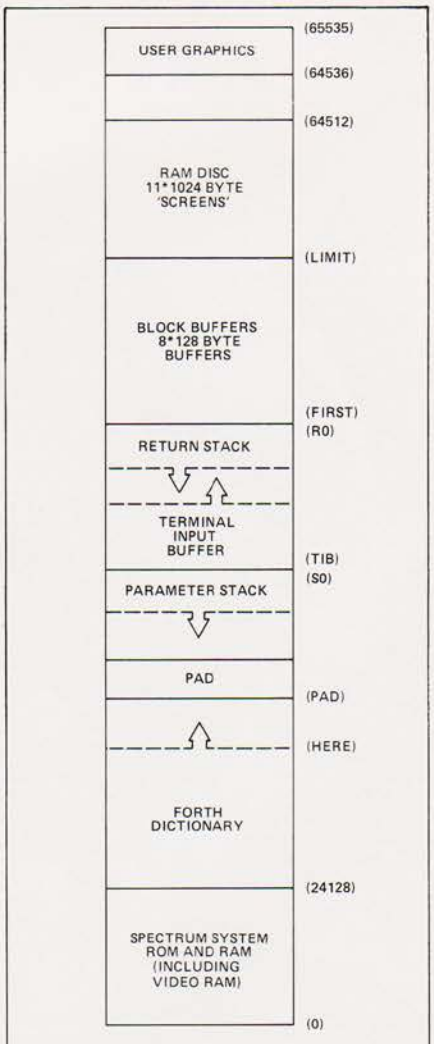


Fig. 1 The memory map for a 48K Spectrum running Abersoft FORTH.

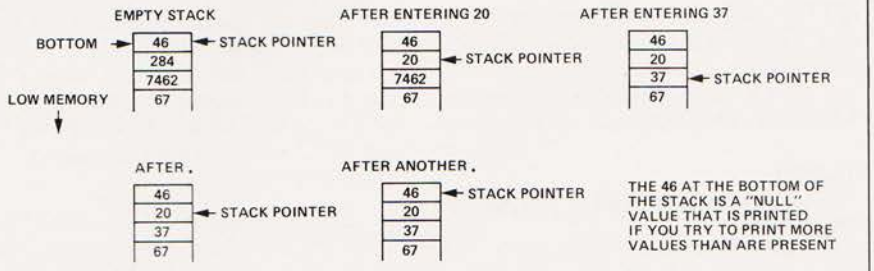


Fig. 2 The stack and its pointer.

continue from in the execution of FRED.

The return stack is also used by the DO-LOOP construction. DO takes two values off the stack, limit and starting point, and puts them on the return stack as a 'temporary storage' place. The word LOOP adds one to the count stored on the top of the return stack and returns to DO if the limit is greater than the count.

We have seen that within a DO-LOOP construction such as:

```
:  COUNTUP
10  0  DO
CR  I  .
LOOP
;
```

we can use the word I to copy the current value of the count from the return stack onto the parameter stack. There is another word I' which copies the second number from the top of the return stack. Within a DO-LOOP this position holds the number for the limit of the loop.

There are a couple of other words which manipulate the return stack:

```
>R    (n  --)
```

takes a value off the parameter stack and puts it on the return stack.

```
R>    (--  n)
```

takes a value off the return stack and puts it on the parameter stack.

Last month I showed the use of >R and R> (in Listing 17) to use the return stack to hold two values which I needed to use frequently and where manipulating the parameter stack would be difficult to follow. CAREFUL! You must remove by using R> any values you put on the return stack using >R and the occurrences of R> and >R must be equal within the same definition or DO-LOOP construction. These commands should be used with great care!

One small problem with Abersoft's FORTH is that the return stack is not very large. It is quite sufficient for most purposes except recursive definitions (ie a word that contains a call of itself). Again, in last month's definition of SEEKEXIT which, depending on the shape of the maze, can call itself a couple of hundred times, I used the word R> at the beginning of SEEKEXIT and >R at the end of SEEKEXIT. This simply held the 'return address' on the data stack while the word was being executed and put the address back on the return stack just before the word finished executing.

Block buffers. Above the return stack is a region of memory containing the 'block buffers'. When FORTH is communicating with the disc (whether it be an actual disc or, in this case, a RAM-disc) information is not taken directly from the disc but is first loaded into a memory buffer. The use of the buffer will be explained in the section on input and output. The starting point of the buffers can be found using the word FIRST which leaves the address on the stack. The end of the buffers can be found using the word LIMIT.

RAM-disc. Above the block buffers is the RAM-disc. This consists of 11 pages or screens of 1024 bytes. It is important to note that in a FORTH program you should not read or write directly to the RAM-disc space, as I shall explain.

The RAM-disc in Abersoft's version of FORTH for the Spectrum ends at location 64512. Between this and the address 65368 (which marks the beginning of the user-defined graphics area) is a space where we can put a small machine code routine. Next month I'll show you how to handle machine code in FORTH and how to create a soft-reset routine, one that will break out of a FORTH program without destroying the memory contents.

User graphics area. Space is provided right at the top of memory for 21 of your own user-defined graphics. The commands for this are given in the Abersoft manual.

Hopefully the memory map will give you a clearer idea of where things are going when we start shuffling information around the system.

BASIC I/O TECHNIQUES

So far we have assumed that all the values needed for a word will either be on the stack or contained within a variable of some sort. We have also only printed text contained in a definition using the ." (dot-quote) command. FORTH does provide many more commands so that you can make your FORTH programs truly 'interactive'.

CHARACTER INPUT/OUTPUT

The simplest of all the character output words is EMIT which will print the character whose ASCII code is on top of the stack. (All ASCII codes for your machine should be contained in your computer manual). For example:

```
66  EMIT
```

will print B ok as 66 is the ASCII code for B.

There is a 'reverse' operation for EMIT called KEY. KEY suspends execution of a program and waits for you to press a key. The ASCII code of the key pressed will be left on the stack.

```
:  WAIT
CR  ." Press any key to continue"
KEY  DROP
;
```

In this case we discard the ASCII code as WAIT does not need it. We could, however, insist that key Y is pressed before continuing.

```
:  YWAIT
BEGIN
CR  ."  Press Y to continue"
KEY
89  ( ASCII for Y)  =
UNTIL
;
```

You can print strings using EMIT by repeatedly using the word, for example:

```
89  EMIT  69  EMIT  83  EMIT
```

would print 'YES', but using ." is easier. EMIT is useful for printing control characters that cannot be enclosed in a dot-quote string.

```
:  CR  13  EMIT  ;
```

We can use KEY in the definition of GETINITS below to enclose a fixed number of characters into the parameter field of a word such as INITIALS. For example:

```
0  VARIABLE  INITIALS  1  ALLOT
```

This sets up a dictionary space for a

variable and then allots to it one more byte of memory so that three characters can be stored in the parameter field of INITIALS. Note: In future, 'parameter field address' will be referred to as PFA. It's easier!

```
:  GETINITS
CR  ." Input 3 initial letters"
INITIALS        ( put PFA of
                  INITIALS on stack)
3  Ø  DO        ( for three letters)
KEY DUP EMIT    ( echo key pressed)
OVER  C!        ( store letter in
                  dictionary space)
1+              ( add one to PFA)
LOOP
DROP  ;         ( discard address)
```

The string, once stored, could be output using a new word TYPE as follows:

```
TYPE    (addr, n  --)
```

This word expects on the stack the starting address of a string and the number of characters in the string, for example:

```
INITIALS  3  TYPE
```

We could instead use a loop containing EMIT which would allow us to change each character before outputting it:

```
:  PRINTINITS
INITIALS
3  Ø  DO
DUP  C@  32  +  EMIT
1+
LOOP
DROP  ;
```

The 32 + in PRINTINITS adds 32 to the character code before printing it. If the initial were in upper case, this would convert it to lower case.

The FORTH programmer has two methods for string input to programs. One is to halt the program and wait for a line of input to be typed into the keyboard (the way that INPUT in BASIC does). The other method is to take the string input from the original input stream.

The second of these two methods uses the FORTH command WORD which acts as follows:

```
WORD    (ASCII code  --)
```

WORD reads one word from the input stream, using the character whose ASCII code is given as a delimiter (ie to mark the end of the word). It then moves the string to the dictionary buffer with the character count in the first byte. Note: WORD will only work from within a colon definition. The start address of the dictionary buffer can be found by typing HERE.

A pretty useless example of WORD follows, but it does illustrate the point without any complications.

```
:  NEXTWORD
32  WORD  HERE  COUNT  TYPE
;
```

NEXTWORD when executed will print out the word that follows it in the input stream. For example:

```
NEXTWORD  example
```

will print

```
example ok
```

To follow all the actions of this word, when you press the return key the two words in the input buffer will be

```
NEXTWORD  example
```

NEXTWORD is executed and its effect is that 32 WORD takes the next word from the input buffer. WORD uses the fact that there is a space (or a return character) to mark the end of the next word (32 is the ASCII code for space). WORD moves the word example into the WORD buffer, the start of which is given by the expression HERE. WORD's buffer will currently contain

```
7example
```

where '7' is the number of bytes in the string. The stack will hold the address of the first byte of the buffer.

COUNT is a FORTH word that uses this address to fetch the contents of the first byte and leaves the address+1 and the count on the stack, that is:

```
COUNT    (addr,  --  addr+1, count)
```

The stack is now ready for the word TYPE which we have seen already. TYPE simply prints out the string.

If we want to input a line of text that contains a space then we can change the delimiter for WORD thus:

```
:  NEXTWORDS
34  WORD  HERE  COUNT  TYPE
;
```

Now we can print out a string containing spaces as long as the string is ended with " (quote) — as 34 is the ASCII code for " — or a Return character. So:

```
NEXTWORDS many words at a time"
```

will print

```
many words at a time ok
```

A similar word to WORD is

```
TEXT    (ASCII code  --)
```

TEXT will accept the following string from the input buffer, delimited by the character code on the stack, but instead of putting the string into WORD's buffer, it puts it into the pad (see memory map). TEXT first fills the pad with blanks (spaces) then moves the string into the pad and leaves the count of the number of characters in the first byte. This example shows that TEXT can be used outside a colon definition:

```
32  TEXT  Hello
```

If the computer responds ok, then type

```
PAD  COUNT  TYPE
```

and Hello ok will be printed.
PAD returns the position of the first byte of the pad.

Now comes the point where we can see how to enclose some text in the dictionary space. First let's set aside some space in the dictionary:

```
Ø VARIABLE STRINGSPACE 38 ALLOT
```

This sets up a dictionary header and allots 40 bytes in total as the parameter field space.

Now let's have two words to fill the string and print it out:

```
:  FILLSTRING
STRINGSPACE  4Ø  32  FILL
32  WORD  HERE  COUNT
STRINGSPACE  SWAP  CMOVE
;
```

Two new words are used in this definition.

```
FILL    (addr, n, char  --)
```

This word fills the 'n' bytes of memory starting at 'addr' with the character 'char'. So, in our example above, STRINGSPACE 40 32 FILL simply fills the STRINGSPACE with spaces.

The word CMOVE is a 'character block move' operation.

```
CMOVE  (from, to, n  --)
```

The 'n' bytes at address 'from' are moved to address 'to'. In the above example, CMOVE moves the string in WORD's buffer into our dictionary space STRINGSPACE. Note: STRINGSPACE will now contain the string found by WORD but it will not have a count as the first byte.

We can use FILLSTRING as follows:

```
FILLSTRING  one-word
```

We can now print out the string with an expression such as

```
STRINGSPACE  4Ø  TYPE
```

Notice here, though, that all 40

characters of STRINGSPACE are typed, even when they are blanks. We can suppress the trailing blanks with a word −TRAILING like this:

```
:  PRINTOUT
STRINGSPACE  40  -TRAILING  TYPE
;
```

PRINTOUT will now only print the leading characters in STRINGSPACE as −TRAILING adjusts the count to miss out any trailing spaces.

As an aside, here's a quick demonstration to show the speed of some FORTH words. Try this:

```
0  16384  6912  CMOVE
```

This moves 6912 bytes of information from the Spectrum ROM into the screen RAM. Although not very pretty, it is quick!

The other method of string input in FORTH is to halt the execution of the program and accept input from the keyboard. Two words that are used for this are QUERY and EXPECT with EXPECT being the most general word. It has the stack effect (address, n —). EXPECT halts execution and waits for n characters from the keyboard, storing them starting from the address given. For example:

```
PAD  20  EXPECT
```

would wait for us to type 20 characters at the keyboard (or fewer if terminated with Newline/Return key) and would store them in the pad.

The other word, QUERY is similar to EXPECT but QUERY expects up to 80 characters and stores them in the terminal input buffer. Combining QUERY with WORD allows us to input up to 80 characters and then split the input text into individual words.

```
0  VARIABLE  FIRSTWORD  8  ALLOT
0  VARIABLE  SECONDWORD  8  ALLOT

:  ENTERWORDS
FIRSTWORD  10  32  FILL
SECONDWORD  10  32  FILL
        ( clear each string space)
CR  ." Enter your two words"
        ( print a prompt)
QUERY    ( wait for input)
32  WORD  HERE
        ( find first word)
COUNT  FIRSTWORD  SWAP  CMOVE
        ( enter into FIRSTWORD)
32  WORD  HERE
        ( find second word)
COUNT  SECONDWORD  SWAP  CMOVE
        ( enter into SECONDWORD)
;
```

Executing ENTERWORDS would look something like this:

```
ENTERWORDS
Enter your two words  Hi there ok

FIRSTWORD 10 TYPE   Hi         ok
SECONDWORD 10 TYPE  there      ok
```

I have underlined the output from the computer to distinguish it from your typing.

STRING HANDLING

We can combine our knowledge of defining words with the use of WORD to provide a more general purpose set of string handling commands. So let us define a 'defining word' which could be used in the form:

```
n  STRING  name
```

which would set up a dictionary space called name, capable of holding a character string n bytes long.

```
:  STRING
<BUILDS
DUP  C,  0  C,
ALLOT
DOES>
2+
DUP  1  -  C@
;
```

This definition would set up a string space that can hold any string up to the maximum length n. When the new word, name, is used the run-time action leaves the address of the string and the count of the number of bytes on the stack. This is what is required by TYPE. So to set up a 20 character string space called SURNAME enter:

```
20  STRING  SURNAME
```

Now, SURNAME TYPE will print the contents of SURNAME, if any.

Before our new string space is useful, though, we need an associated word that enables us to input characters to it.

```
:  INPUT$
DROP  1  -
DUP  1  -  C@
CR  ." ? "  QUERY
1  WORD
HERE  C@
<  IF
." String too big "
DROP  QUIT
THEN
HERE  DUP  C@  1+
ROT  SWAP  CMOVE  ;
```

You can now use INPUT$ in a program in much the same way as you use the BASIC equivalent.

```
10  STRING  FORENAME

:  STORENAME  CR
." Please enter your first name "
FORENAME  INPUT$  ;
```

Notice here that you put the name of the string space before the word INPUT$.

NUMBER INPUT/OUTPUT

So far we have only input numbers onto the stack prior to a calculation, or we have stored them in a variable. However, we can create a new word to behave like INPUT in BASIC. We do this using WORD with another FORTH word NUMBER, thus:

```
:  INPUT
CR  ." ? "  ( print a prompt)
QUERY       ( get a line of input)
1  WORD  HERE
            ( copy word into
              buffer)
NUMBER      ( convert to a number)
DROP        ( convert to single
;             length)    .
```

The new word NUMBER has the following action:

```
NUMBER    (addr  --  d)
```

This is a very powerful word. It takes an ASCII string starting at 'addr' and will convert it to a 'double length' number on the stack.

INPUT above halts execution of a program and uses WORD and NUMBER to convert what you type in into a 'single length' integer, which it leaves on the stack. We have not come to the use of double length numbers yet, but it is enough for now to know that a double length number is stored on the stack using four bytes of memory. If the value of the number is within the single length range (−32768 to +32767) then the top two bytes of the stack are zero. DROP at the end of INPUT simply drops this top zero to leave a single length integer.

FORTH AND THE DISC

We have only used our RAM disc (Abersoft FORTH) so far to hold definitions for us before we compile them. However, we can use the disc to store data of any kind, as it is possible for the FORTH system to treat disc space as an extension to memory.

FORTH traditionally divides the space on a disc into numbered blocks of 1024 bytes each. When loading or editing a definition screens are fetched one at a time into an area of RAM called the 'block buffers'. Programs can extend over many blocks and a block may contain a command to load successive blocks.

Abersoft FORTH works a little differently in that there are eight buffers in RAM, each of which is 128 bytes long, and it also treats the RAM ▶

disc space as 'virtual memory' ie pretends it is really a disc drive. But each block in the RAM disc is only 128 bytes long. This arrangement is quite transparent to the user when you are loading and editing programs as you can address screens of RAM disc of 1024 bytes each. However, if we want to use the disc to store data then we must be aware of the actual configuration.

So, with Abersoft FORTH the RAM disc is divided up into 88 blocks of 128 bytes each. The block buffers are an area of RAM that can hold up to eight blocks at any one time. The reason we can call our RAM disc virtual memory is that any block of the disc can be brought into the buffers, used for reading from or writing to and then replaced in the disc space when it is finished with.

A command that accesses the disc is BLOCK which acts thus:

```
BLOCK   (n  --  addr)
```

loads a given block, n, from disc into a buffer then leaves the address of the start of the buffer on the stack.

Try this: as each 'screen' of disc is 1024 bytes then each screen actually spreads across eight blocks. The blocks are numbered from 0 to 87 so our screen 1 starts at block 8 and ends on block 15. If you already have some text edited onto screen 1 then type

```
8  BLOCK  128  TYPE
```

This will bring the first block of screen 1 into a block buffer and leave the address of the buffer ready for TYPE which will list it out. This is the top two 'lines' of screen 1.

This is effectively what LIST does, but LIST lists all eight blocks of a screen and puts line numbers in front of each 64 byte line.

Here's a word that will print line zero of all 11 screens in the RAM disc (these usually contain comments).

```
:  LISTLINES
88  0  DO  CR
I  BLOCK  64  TYPE
8  +  LOOP  ;
```

This steps through blocks 0, 8, 16,.., 80 which represent the first block of each screen and then prints the first 64 bytes (one line) of each of these blocks. (If you have not used line 0 on some blocks they may show up as a series of '?').

Another handy word for Abersoft FORTH users is given below. When you list a block to a screen it is quite difficult to read, as each 'line' in FORTH spreads over two Spectrum screen lines. The definition below will list a block in 32-byte slices such that if you connect your ZX Printer to the Spectrum, it will print the first 32 characters of all the lines and then print the last 32 characters of all the lines. If you cut your printout and lay the two halves next to each other you have a 64-character wide listing!

```
:  SPLITLIST  ( screen num  --)
1  LINK       ( turns on printer)
CR
8  *  DUP  DUP
8  +  SWAP  DO
I  BLOCK  32  TYPE
I  BLOCK  64  +  32  TYPE
LOOP
DUP  8  +  SWAP  DO
I  BLOCK  32  +  32  TYPE
I  BLOCK  96  +  32  TYPE
LOOP
0  LINK       ( turn off printer)
;
```

WE WANT INFORMATION

Finally this month we'll see how you can use the disc to store and retrieve information. As I've explained, whenever you LIST, LOAD or EDIT a screen of the disc, the information is first transferred into the buffers prior to use. The idea of this is that once a particular block is in memory, it does not have to be re-read from disc every time you want to use that block, which is quite slow when you are using a real disc system. So once a block is loaded it remains in memory until the buffer is required for something else. If the old block has been changed then it is rewritten to disc before that buffer can be used for anything else. Each buffer has associated with it an 'update flag' which is altered whenever the contents of a buffer have been changed by editing commands; this ensures that the updated contents are written back to disc when you have finished editing.

We can use the disc space to 'load' and 'save' any form of data, but if we were to simply move data from memory locations, for example numeric data, then a listing of the block would be quite unintelligible and, in some cases, would crash the system. For this reason the disc space is usually only used to store characters and strings. These can be converted to numbers if necessary.

For example, suppose we wanted to store a list of names and addresses on disc. We'll use screen 10 for this as it is the last one. This contains blocks 80 to 87 inclusive. On each line of the disc we'll store just one name and one address, 64 characters in total. That means we could get two names and addresses per block or 16 for the screen. Edit onto screen 10 a few suitable names and addresses, but make sure that the address starts from the 32nd character (ie second half of the line).

So now our screen will contain

name 1	address 1
name 2	address 2
name 3	address 3
..	..
..	..

You can use as many spaces in between words as necessary but each name or address must not exceed 32 characters.

Now we require a simple word that will take a number off the stack in the range 0 to 15 and will print out the name and address on the screen, suppressing any unnecessary trailing spaces.

```
80  CONSTANT  STARTBLOCK
2  CONSTANT  NAMES-PER-BLOCK

:  PRINTDETAILS   (n  --)
DUP  DUP  0<  SWAP  15  >
OR  IF  ." Out of range for file!"
DROP  QUIT  THEN
( calculate block)
NAMES-PER-BLOCK  /MOD
                 ( offset in blocks)
STARTBLOCK  +    ( actual block)
BLOCK            ( address of first
                   byte)
SWAP  IF  64  +  THEN
DUP  CR
32  -TRAILING  TYPE  SPACE
32  +  32  -TRAILING  TYPE SPACE
;
```

The method of finding the correct address is quite simple. If the number is odd then NAMES PER BLOCK /MOD will leave the number of blocks from the start on top of the stack and a '1' second on the stack if it is the second name in the block. Once the right block is found then the two fields are printed out, separated by a SPACE.

WRITING TO THE DISC

If you want your program to write to space on the disc, you can do it by fetching the particular disc block into a buffer using BLOCK. For example:

```
8  BLOCK  ( top 2 lines of
            screen 1)
64  32  FILL
          ( fill top line with
            spaces, ASCII 32)
```

Now having altered the buffer you must execute the word UPDATE to ensure that, when the buffer you used is needed next, the updated contents are written to disc. So now if you list screen 1:

```
1  LIST
```

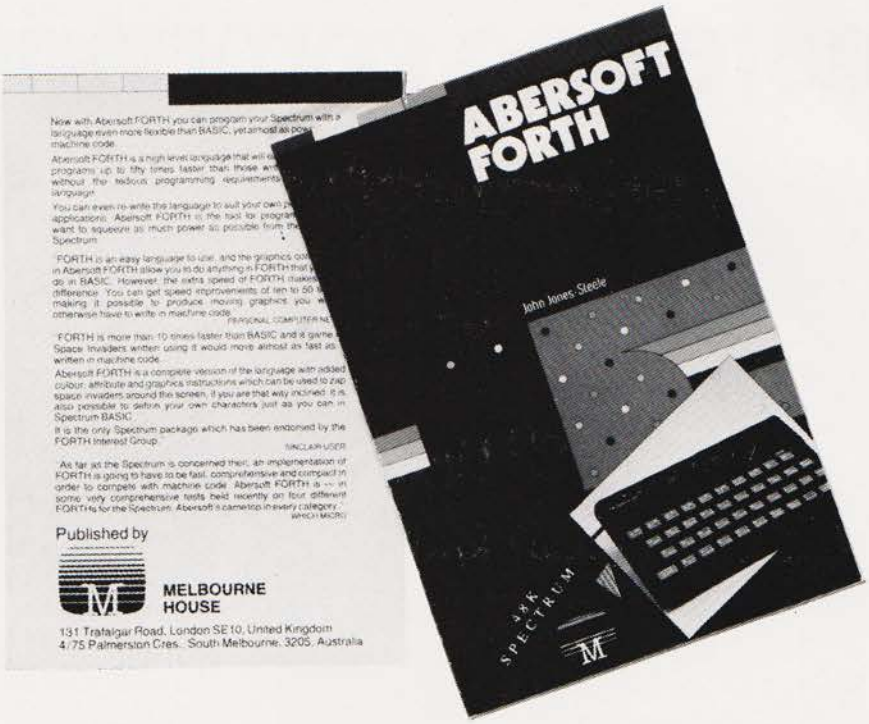
you will see that the top line has been filled with spaces. BLOCK is quite an 'intelligent' word in that if a block required is already in one of the buffers then it does not need to be re-read from disc: the address of

the buffer holding the block is simply left on the stack.

You will notice that this month I have not given any instructions 'This is the way to achieve input in FORTH', but I have simply tried to give you a few ideas. You can manipulate input in FORTH to do exactly what you decide; it is possible to make your FORTH I/O behave exactly like your favourite other language such as BASIC or Pascal or whatever you choose: but with FORTH, user-friendliness is easy to achieve.

COMING SOON

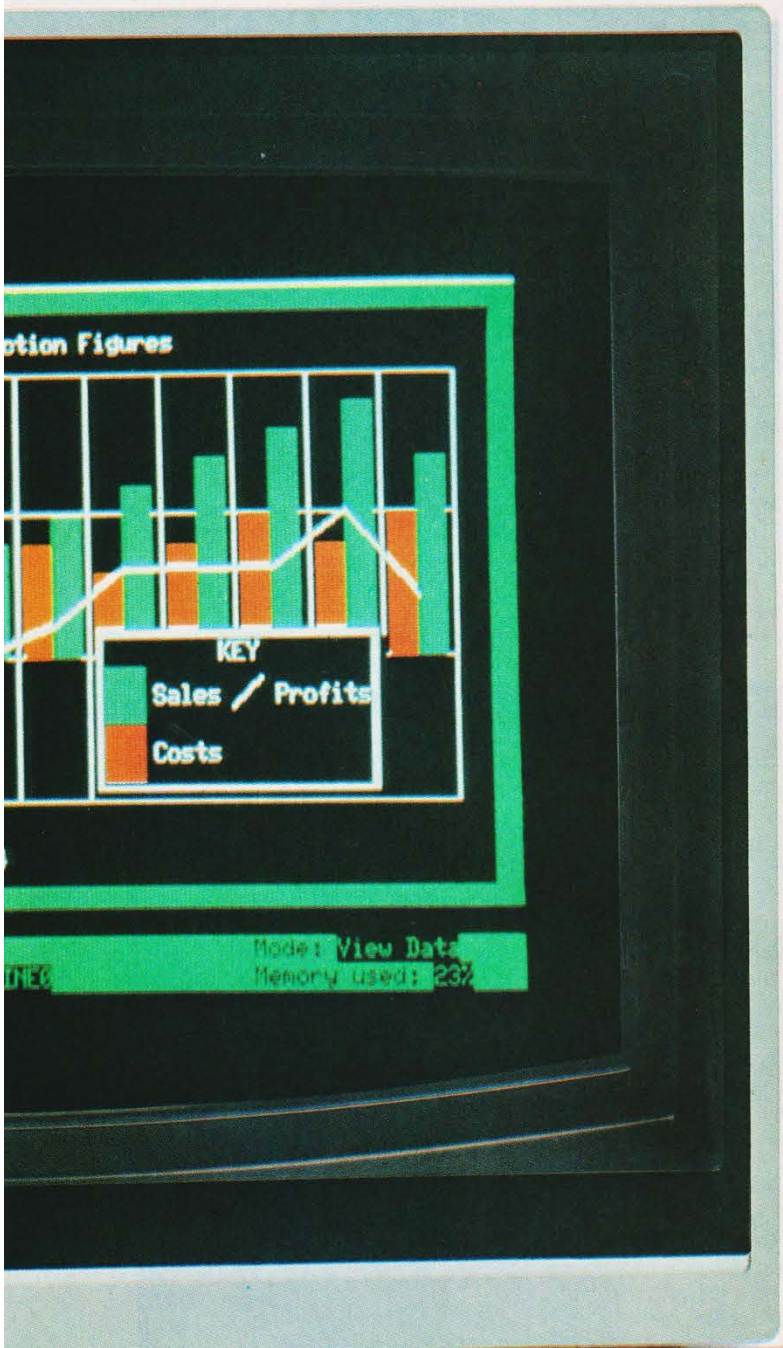
Next month I'll be talking about some very diverse aspects of FORTH. I'll spend some time on handling various length numbers and 'pictured' output of numeric information (like 13.10.83 for dates and such like). We'll see how FORTH can input and output numbers in different bases like binary, hex, octal, even base 255! I'll provide some information on how you can implement machine code routines within FORTH, and for Abersoft users an interrupt routine that will let you break out of a 'stuck' program. Finally we'll see how to extend the compiler to implement the sort of structures that are usually missing, like the CASE structure.



The Abersoft FORTH used in this series is now available from Melbourne House, who have acquired all publishing rights.

New-Sinclair QL

There's no comparison chart, b



The Sinclair QL is a new computer.

Not just a new Sinclair computer, but a totally new sort of computer – nothing like it exists anywhere.

It's not just a bit better than this, or a bit cheaper than that – it's a computer that's very hard to compare with anything. Just check the features below – and if you don't agree, take up the challenge at the end of the advertisement.

If you do agree, there's only one course of action you can take... get yourself a Sinclair QL at the earliest possible moment.

The Sinclair QL has 128K RAM. Big deal?

Several micros offer 128K RAM, or more, as standard. The 'What Micro?' table for December 1983 lists over 50 of them – but 40 of the 50 micros listed cost over £2,500!

The Sinclair QL offers you 128K RAM for under £400, and an option to expand to 640K. That's a lot of bytes to the pound!

The Sinclair QL has a 32-bit processor. Who else?

Under £2,700, nobody. Even the new generation of business computers, such as the IBM PC, are only now beginning to use 16-bit processors.

At prices like this, the Motorola 68000 family – widely regarded as the most powerful microprocessors available – will remain a luxury.

Yet with the Sinclair QL, the 32-bit Motorola 68008 is available for less than £400.

You can also be sure that the QL will not become outdated. 32-bit architecture is future-proof.

32-bit processor architecture, 128K RAM, and QDOS combine to give the QL the performance of a mini-computer for the price of a micro.

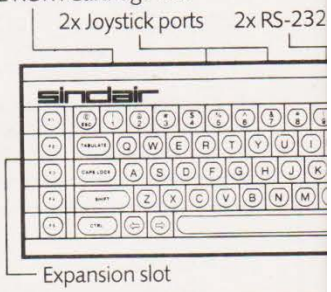
Exclusive: new QDOS operating system

No competition! QDOS sets a new standard in operating systems for the 68000 family of processors, and may well become the industry standard.

QDOS is a single-user, multi-tasking, time-sliced system using Sinclair's new SuperBASIC as a command language.

One of its most significant features is its very powerful multi-tasking capability – the ability to run several programs individually and simultaneously. It can also display the results simultaneously in different portions of the screen. These are features not normally available on computers costing less than £7,000.

Eleven input/output ports



New professional keyboard

The QL keyboard is designed for fast input of data and programs.

It is a full-size QWERTY keyboard, with 65 keys, including a space bar; left-and right-hand shift keys; five function keys; and four separate cursor-control keys – key action is positive and precise

A membrane beneath the keyboard protects the machine from dust (and coffee!), and for users who find an angled keyboard more comfortable, the computer can be raised slightly at the back by small detachable feet.

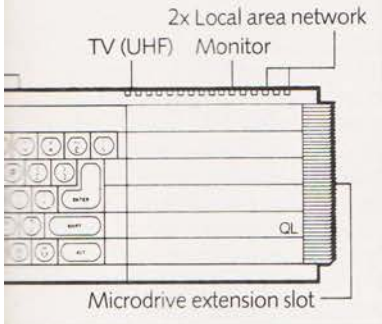
£399

ecause there's no comparison!

Advanced new friendly language – Sinclair SuperBASIC

The new Sinclair SuperBASIC combines the familiarity of BASIC with a number of major developments which allow the QL's full power to be exploited.

Unlike conventional BASIC, its procedure facility allows code to be written in clearly-defined blocks; extendability allows new procedures to be added which will work in exactly the same way as the command procedures built into the ROM; and its constant execution speed means that SuperBASIC does not get slower as programs get larger.



Two 100K microdrives built in

The Microdrives for the Sinclair QL are identical in principle to the popular and proven ZX Microdrives, but give increased capacity (at least 100K bytes each) and a faster data-transfer rate. Typical access speed is 3.5 seconds, and loading is at up to 15K bytes per second. The Sinclair QL has two built-in Microdrives. If required, a further six units can be connected.

Four blank cartridges are supplied with the machine.

Included – superb professional software

The suite of four programs is written by Psion specially for the QL and incorporates many major developments. All programs use full colour, and data is transportable from one to another. (For example, figures can be transferred from spreadsheet to graphics for an instant visual presentation.)

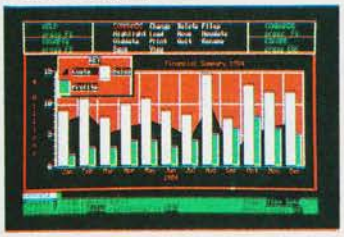
Word-processing

Certain to set a new standard of excellence, QL Quill uses the power of the QL to show on the screen exactly what you key in, and to print out exactly what you see on the screen.

A beginner can be using QL Quill for word-processing within minutes.

QL Quill brings you all the facilities of a very advanced word-processing package.

Business graphics



QL Easel is a high-resolution colour program so easy to use you probably won't refer to the manual! It handles anything from lines, shaded curves or histograms to overlapping or stacked bars or pie charts. QL Easel does not require you to format your display before entering data; it handles design and scaling automatically or under your control. Text can be added and altered as simply as data.

Spreadsheet

QL Abacus makes simultaneous calculations and 'what if' model-construction easier than they've ever been. Sample applications are provided, including budget-planning and cash-flow analysis. QL Abacus allows you to refer to rows, columns and cells by names, not just letters and numbers. Function keys can be assigned to change a variable and carry out a complete 'what if' calculation with a single key-stroke.

Database management

QL Archive is a very powerful filing system which sets new standards, using a language even simpler than BASIC. It combines ease of use for simple applications – such as card indices – with huge power as a multi-file data processor.

An easy-to-use labelling facility means that you don't have to ask for your file by its full name – a few letters are enough.

New – the Sinclair QLUB

The QLUB is the QL Users Bureau. Membership is open to all QL owners. For an annual subscription of £35, QLUB members receive one free update to each of the four programs supplied with the QL, and six bi-monthly newsletters. Sinclair has also made exclusive arrangements for QLUB members to obtain software assistance on QL Quill, Abacus, Archive or Easel by writing to Psion.

The Sinclair QL challenge

If you're seriously considering any other computer, post the coupon for a blow-by-blow comparison. We'll take a *published* comparison chart for the machine you're considering (not one we've created ourselves) and give you the Sinclair QL figures, detail by detail.

Take action today!

To order by mail

– complete the coupon and send it to the FREEPOST address below. For credit card holders it may be possible to extend your credit limit. Full details will be sent when we acknowledge your order.

To order by telephone

– phone Camberley (0276) 685311;
– have your credit card (Access, Barclaycard, Trustcard) number ready. It may be possible to extend your existing credit limit. Please ask our telephone staff for more details.
Please do not use this number for other enquiries.

For more information

Phone Camberley (0276) 686100, or use the coupon to get a QL brochure.

Please allow 28 days from receipt of order. Remember that Sinclair offers a 14-day money-back undertaking.

Send to: Sinclair Research Ltd, Computer Division, FREEPOST, Camberley, Surrey, GU15 3BR.

Qty	Item	Code	Item Price £	Total £
	Sinclair QL Computer	6000	399.00	
	QLUB membership (one year)	6100	35.00	
	Postage & packing (any order over £390)	6999	7.95	

Please tick the appropriate box-

☐ I enclose a cheque made payable to Sinclair Research Ltd for £_____ £_____

☐ Please charge my Access/Barclaycard Trustcard Account No. _____

Signature_____

Mr/Mrs/Miss _____

Address _____

_____ COT 403

☐ Please send me a chart comparing the Sinclair QL computer with a_____

☐ Please send me a Sinclair QL brochure

Sinclair Research Ltd, Stanhope Road, Camberley, Surrey, GU15 3PS

sinclair

Peter Green

PROCopinion

Suddenly there's a lot of activity down at the lower end of the computing price bracket. And some of the stuff coming your way is very exciting indeed.

It isn't getting any easier to decide which microcomputer you should buy. Despite the collapse, or near-collapse, of several computer companies, and the strategic withdrawal of Texas Instruments from the battlefield, there is still a host of new machines hitting the marketplace. Towards the end of last year Acorn launched their Electron, of which more anon; Memotech decided to go beyond making peripherals, and their MTX500 is reviewed next month; the Spectravideo is currently on our test bench and heralds the Japanese MSX invasion; today I saw Sir Clive unveil his latest bombshell; the Oric has been re-styled, re-ROMed and relaunched as the Atmos; and the Elan is lurking in the wings for an April launch. All these machines cost less than £400 and several less than £200, and all offer a spec for this price that was undreamed of a year or two ago.

Maybe the choice will be made easier by the supply problems that have now become an unwelcome industry standard. One reason for the great initial success of the Dragon 32 was that it was a machine you could actually go out and buy. Electrons are still a bit thin on the ground, and this is reflected in the fact that very few review cassettes are arriving at the offices with 'Electron' on the label. Or maybe it's the problems that are involved in writing good BASIC games in the meagre memory left over by the rapacious graphics of the Electron. I should know: I've been doing a series of conversions of games for ASP Software on our Electron over the past few weeks, and the missing Mode 7 with its very economical Teletext graphics has been sorely missed.

Planetfall, in its present form, simply will not fit, although I'm looking at ways to rewrite it and slim it down. The Valley runs in practically the same way on the Electron as on the Beeb, but the remaining games, such as Stockmarket and Demon Knight (a text Adventure not published in CT) cause more of a problem. Lacking graphics, these games need various combinations of colours in different text windows to jazz up their appearance. They also require 40-column screens. Anyone familiar with BBC/Electron BASIC knows that Mode 4 gives 40 columns and two colours, while Mode 1 gives 40 columns and four colours (Mode 6 is out because it gives non-continuous character cells which results in black stripes when you change background colour). Mode 1 leaves 8K free for programs, and I've been having great fun trying to pack programs into that.

If anyone's wondering why I don't close up the Mode 6 cells by re-programming the 6845 video chip, it's simple — the Electron doesn't have a 6845. Acorn have put in a ULA instead, and it's not programmable. It also handles the sound, instead of the General Instruments sound chip in the BBC, and this gave me a few problems too when trying to get explosions in the wargames. Using a short burst of white noise sounded fine in immediate mode, but putting it in the program followed by a call to a delay procedure resulted in a warbling tone. Strange. Obviously something in the delay routine was modulating the sound channel somehow, so I altered the TIME procedure to a REPEAT-UNTIL loop, then a FOR-NEXT loop. No change. Right, I thought, any sort of activity by BASIC seems to mess up the sound: so I tried CALLing a machine code delay loop that disabled the 6502 interrupts during its execution. Still no joy: what actually happened was that the SOUND call passed the parameters into the ULA, and the delay call stopped the interrupts, which stopped the ULA! — giving a half-second silence followed by that same warbling tone.

Just about to give up in desperation and do without sound effects, I remembered one last way of obtaining a delay on the Electron — the INKEY statement. Not expecting any success, I tried it, and it worked; a white noise explosion. Don't ask me how it works, but I pass it on in the hope it'll be of use to someone.

STR$ acts a bit funny too, so much so that it can only be due to a bug in the ROM. Using STR$ to allow print formatting of the bank balance in Stockmarket, I suddenly started getting figures like £5.01111111 on the screen, which suggested a faulty rounding routine in the program. In fact it turned out to be STR$ at fault. Let M=8.04: PRINT M gives 8.04, but PRINT STR$(M) gives 8.03999999. So much for nine-digit accuracy.

This sort of thing seems to have happened because Acorn haven't simply copied the BBC BASIC ROM and 'blanked off' the sections that require hardware the Electron doesn't possess: the operating system appears to have been completely re-jigged. I say this because I wanted to use the page 3 location containing the current graphics mode in a 'universal' routine I was writing, and on the BBC it's location &367. On the Electron it's a &355. Acorn probably have good reasons for doing all this, but it seems a bit dangerous to muck around with a field-tested OS and risk creating new bugs.

An Oric by any other name is an Atmos . . .

This is Sinclair's QL, a sleek-looking beast.

COMPETITION TIME

The Electron can do without this sort of problem, because it's going to be facing some stiff competition soon. Both MSX machines like the Spectravideo and the Memotech MTX 500 use the Texas Instruments 9900-series graphics chip, as used in the 99/4A itself and the Cortex kit computer published by ETI just over a year ago. This means they have 16 colours on screen simultaneously, plus up to 32 sprites, with no overhead on user RAM because separate video RAM is used. Admittedly the resolution is only 256 by 192 maximum, but most owners of a micro in this sort of price range are going to be using them with a domestic TV which cannot really handle higher resolutions.

The Oric, too, is looking healthier in its new incarnation as the Atmos. Re-styled in black and red with a new ROM (with most of the old bugs of the Oric-1 fixed), it also offers a full-pitch typewriter keyboard, several new commands such as PRINT @ and versatile cassette operations, and matching printer and 320K 3" micro disc drives. (Designer Paul Johnson's SF tendencies seem to have surfaced again: wasn't the Atmos the creature that lived under the city in the film **Barbarella?).** Even better is the news that the manual has been written by a team of writers working outside of Oric under the auspices of an independent publisher — Pan Books. Hopefully this will result in a more accurate and more readable manual than most other manufacturers can manage.

Visitors to the PCW Show last year will have seen the unusual-looking Elan on show, or rather a facsimile thereof. With a promised launch date of April, the BASIC reference manual has already been prepared and we've been having a look at it in the office. This is a real programmer's machine, without being too discouraging to the beginner because just about all the machine options have default values. Flexibility is the keynote of the design: for example, all the peripherals and the various parts of the machine such as the screen are specified as channels, and information can be shuttled around in any way you like by opening and closing the various channels.

The BASIC is designed along the principle of more is better, and there are keywords here you won't have seen anywhere else before — I count 201 of them. Some of them seem a bit over the top: do you really need three types of REM? (REM can only be used at the start of a line, ! can be used for a comment at the end of a line, and PROGRAM is a REM that you use to give your program a name (honest!)). There are three types of log; base 10, base 2 and base e. There are curious new keywords like LTRIM$ and RTRIM$, which remove leading and trailing spaces from strings, and words to allow the bitwise ANDing and ORing of strings. Like other BASICs there is user error-handling, but you can also define your own errors — for example, you could define alphabetical input to be an error if you only wanted numerical input, and the program would jump to your own error handling routine if invalid input was detected. The graphics are pretty impressive, pretty, and more versatile than the BBC's. The BASIC has lots of nice structures to encourage you to program more elegantly (CASE and DO-WHILE, for example). Then there's the built-in word processor that uses the keyboard joystick for cursor control. I'm looking forward to the delivery of our review model!

Finally, but by no means least, comes Sir Clive's new machine, the Sinclair QL. From the first impressions at the launch, I have to say that Uncle has done it again — this machine will RUN and RUN. QL stands for Quantum Leap, says the demon knight, and he's not far off the mark. Listen to the spec: a 32-bit 68008 microprocessor, 128K RAM as standard, expandable to 640K, twin built-in Microdrives with an increased capacity and faster data-transfer rate (100K bytes at 15K bytes per second), networking capability, a full-size QWERTY keyboard, monitor output, joystick ports, two RS-232C ports, ROM cartridge slot, two video modes (512 by 256 four-colour and 256 by 256 eight-colour), multitasking QDOS operating system utilizing screen windows, structured BASIC, small size and excellent styling. The cost is a staggering £399, which includes a suite of four utility programs on Microdrive cassette; a wordprocessor, a business graphics package, a spreadsheet and a relational database.

That's pretty formidable ammo in the coming battles. Despite a couple of faux pas — there's no cassette port, so all software must be on Microdrive, and the four colours in high-res mode are not selectable, being black, red, green and white — this computer should find a ready market. It's worth buying just for the bundled software, and will certainly appeal to the target market of students, scientists and businessmen. Despite the brake that the lack of cassette facilites will put on available software, it will do very well in the home market too. And it's still only January . . .

This could be a very interesting year.

WE WANT INFORMATION

The very welcome re-runs of The Prisoner on Channel 4 have reminded me of an incident that happened back when Maplin became computerised. For those who don't know, Maplin are a large mail-order company for electronic components who I've always found to be very efficient with very fast delivery. When their computer was installed, my next order came with a form telling me my new customer number, to be used in all future transactions. Because I have a peculiar sense of humour, I sent back the next order form with "I am not a number, I am a free man" written across it. The partially-filled order arrived three weeks later . . .

Computers definitely don't appreciate jokes.

WHO DO YOU DO

Doctor Who's back on the box, too, and I like the new style control console in the Tardis very much. But close observers of the video screens will have noticed a slap in the face for Uncle Clive. You might be able to control a power station with a ZX81, but it takes a BBC Micro to run a time machine!

Don Thomasson

THE LEGIBLE SPECTRUM

The ZX Printer is cheap, and (dare we say) a bit nasty. We look at ways of connecting high-quality parallel printers to the Spectrum.

The ZX Printer provides a low-cost means for generating hard copy from the Spectrum, but the quality of the result leaves something to be desired. Substituting a better printer involves a number of problems (which can be overcome in different ways):

- The Spectrum has no explicit parallel interface.
- The output to the ZX Printer is in the form of dot patterns, not character codes, and the dot patterns are not compatible with those required by graphic printers.
- The codes used within the Spectrum depart from standard ASCII in a number of respects.

Two elements are needed to overcome these problems, a hardware system and associated software, which is partly related to the hardware arrangement and partly dependent on the code conversions which are to be provided. An attempt is made here to cater for a number of possible requirements.

THE HARDWARE

It will be assumed that the printer to be used is equipped with a standard Centronics parallel interface. This is no place to start a debate on the relative merits of serial and parallel printer connections, but it should be noted that some serial interfaces inhibit some of the useful printer capabilities because they are not fast enough in the transfer of data.

A Centronics interface provides seven or eight data lines, a 'strobe' line to tell the printer to take data, a 'busy' line to indicate that the printer is unable to accept data, and an 'acknowledge' line to signal that data can now be accepted. The combined presence of 'busy' and 'acknowledge' can be confusing, and it should be understood that they are alternatives, only one being used in a given system.

The action cycle involves putting data on the data lines, and then making 'strobe' low for not less than ½μS. The printer accepts the data, making 'busy' high until the system is ready to take another data byte. For most transfers, 'busy' will remain high very briefly, just long enough for the data byte to be stored. But when actual printing begins 'busy' remains high until the printing is complete.

'Acknowledge', normally high, goes low for 5μS (approximately) at the time when 'busy' falls to low. It can thus be used to reset a bistable in the interface hardware, the bistable having been set by 'strobe'. In this case, the bistable effectively provides the 'busy' signal.

Of the two alternatives, 'busy' is more useful, on two counts. First, it makes the bistable unnecessary. Secondly, the line will appear to be high if the printer is not connected, and this can be used to avoid a 'hang-up' in those circumstances.

The hardware thus needs to handle 10 lines: eight data outputs (or in some cases only seven), a 'strobe' output and a 'busy' input. The 8255 PPI (Programmable Peripheral Interface) provides a useful basis, since it can implement three separately-addressable ports. The connections for an Epson MX80 are defined in Fig. 1. Note that the connections to the printer should either be in twisted pair lines or in ribbon cable with alternative lines earthed.

For those who want to keep constructional work to a minimum, Kempston Microelectronics produce such an interface, but without the printer connection cable. They also produce a complete printer interface, of which more anon.

Those who wish to squeeze the maximum performance from the system may care to note that a number of printer lines can be added:

Paper Out: Pin 12, with earth on pin 30. High when paper is exhausted. Connect to port C, bit 1 (pin 15 on 8255).

Select: Pin 13. High when printer selected. Connect to port C, bit 2 (pin 16 on 8255).

Error: Pin 32. Low if error condition found. Connect to port C, bit 3 (pin 17 on 8255).

Initialise: Pin 31. If this line is taken low, the printer is re-initialised. Connect to port B, bit 1 (pin 19 on 8255).

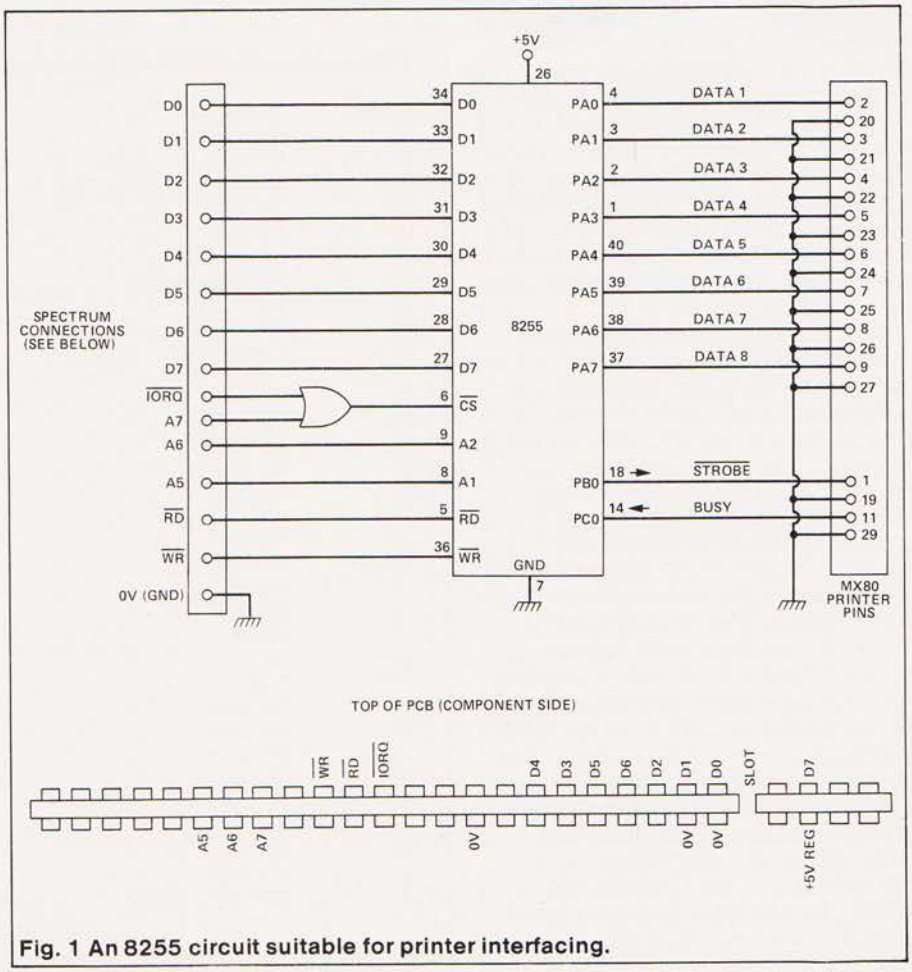


Fig. 1 An 8255 circuit suitable for printer interfacing.

Auto LF: Pin 14. While this line is low, an automatic line feed occurs after each line is printed. Connect to port B, bit 2 (pin 20 on 8255).

These extra connections complicate the software somewhat, and provisions for handling them will not be included in the routines given here. However, they do allow some interesting automatic actions. It may be useful, in any case, to earth the Auto LF line, since this avoids a need to generate LF following each CR.

SOFTWARE INTERFACE

The 8255 interface has to be initialised before use by the routine shown in Listing 1. The first output sets up the 8255 to make port A output, port B output, and port C input. The second output sets bit 0 of port B (strobe) high.

The complete Kempston interface requires a different initialisation procedure, shown in Listing 2. A complete driver routine on tape is provided with the hardware.

The next stage of initialisation must persuade the Spectrum to call the special interface routines instead of the usual printer driver. This involves changing the relevant 'channel data', a task performed by Listing 3.

Locations 5C4F/5C50 (23631/2 in decimal) hold the address of the start of channel data, which is in blocks of five bytes each. We wish to change the fourth block, which relates to the 'printer' channel, so we have to add 15 to find its start address. We load the address with the start address of the interface routines. If we wish to be completely sure, we can set the next three locations to C4 15 50, these being the original values, which will usually be unaltered.

All that remains is to determine the contents of the new printer driver, and those will depend on what we want to achieve. Table 1 shows the differences between ASCII and the internal code of the Spectrum, from which it is seen that a number of codes used by printers are not available in standard form. The question is, how many of these codes do we need? That will be determined by individual taste and necessity.

It will also be affected by the characteristics of individual printers. The response given above apply to the basic Epson MX80, but an MX80 is not always an MX80! No less than seven variant ROMs have been identified, each set making the printer work in a slightly different way. (Some stop it working altogether . . .) The Graftrax-80 ROMs, for example, make codes 0E and 0F unnecessary by providing alternatives based on the Escape prefix. This sort of thing makes it very difficult to be dogmatic about which codes are needed.

Almost inevitably, Escape will be required, and it can be generated by CHR$ 27. Other codes are less straightforward. Code 6 needs to implement a half-screen-width tab, or a return if the midway point has already been passed. Code 8 needs to implement an erasing backspace. Code 0DH needs to act normally, but it should also reset the column count.

Codes 10-15 are special, in that they are really prefixes, and the following code should not print, whatever it is. Similarly, codes 16 and 17 (AT and TAB) should take special action, spacing forward to a given position. Finally, codes A5H upwards are tokens, and need to be expanded by a Spectrum routine starting at 0C10.

A skeleton program for this is shown in Listing 4. Not all the process modules are shown, since the purpose of the listing is to demonstrate the necessary framework, and there may be differing ideas as to the best way to implent the code changes. The GRAPH routine, for example, might set a flag which causes both the current code and the next to be ignored, or it might change the channel data so that the code is re-routed to the appropriate Spectrum routines. Those who want to discover how to implement this will find the Kempston routines of considerable interest.

```
LD  A,89H
OUT (7FH),A
LD  A,1
OUT (3FH),A
```

Listing 1. 8255 initialisation.

```
LD  BC,0E3BFH
LD  A,81H
OUT (C),A
LD  A,0FH
OUT (C),A
```

Listing 2. Kempston initialisation.

```
LD  HL,(5C4F)
LD  BC,000FH
ADD HL,BC
LD  (HL),Lower byte of Address
INC HL
LD  (HL),Upper byte of Address

The Address is the start of
the printer driver routine.
```

Listing 3. Channel changing.

Table 1. Spectrum and ASCII codes.

Code	Spectrum	ASCII	Printer
00	Not used	NUL	—
01	Not used	—	—
02	Not used	—	—
03	Not used	—	—
04	Not used	—	—
05	Not used	—	—
06	PRINT comma	—	—
07	EDIT	Bell	Buzzer
08	Cursor left	—	—
09	Cursor right	HT	Horizontal Tab
0A	Cursor down	LF	Line Feed
0B	Cursor up	VT	Vertical Tab
0C	DELETE	FF	Form Feed
0D	ENTER	CR	Carriage Return
0E	Number	SO	Enlarged Characters
0F	Not used	SI	Condensed characters
10	INK	—	—
11	PAPER	DC1	Select
12	FLASH	DC2	Cancel condensed
13	BRIGHT	DC3	Deselect
14	INVERSE	DC4	Cancel enlarged
15	OVER	—	—
16	AT	—	—
17	TAB	—	—
18	Not used	Cancel	Clear buffer
19	Not used	—	—
1A	Not used	—	—
1B	Not used	ESCAPE	Prefix code
1C	Not used	—	—
1D	Not used	—	—
1E	Not used	—	—
1F	Not used	—	—
7F	Copyright symbol	Erase	Erase

```
        CP  6
        JR  C,QUERY         Codes 0-5: Print as ?
        JR  Z,COMMA         Code 6: Half-screen width tab
        CP  8
        JR  C,QUERY         Code 7: Print as ?
        JR  Z,BACK          Code 8: Erasing backspace
        CP  0DH
        JR  C,QUERY         Codes 9-0C: Print as query
        JR  Z,RETURN        Code 0D:  Return and reset column count
        CP  10H
        JR  C,QUERY         Codes 0E-0F: Print as query
        CP  16H
        JR  C,GRAPH         Codes 10-15: Graphics control
        CP  18H
        JR  C,POS           Codes 16,17: Position control (AT,TAB)
        CP  7FH
        JR  C,PRINT         Print codes 18-7EH unaltered
        JR  Z,COPYR         Convert 7FH to '@'
        CP  0A5H
        JR  C,SPACE         Print 80H - A4H as space
        SUB 0A5H            Convert token to number
        JP  0C10H           Convert token number to text
QUERY   LD  A,3FH           Print a query
        JP  PRINT
BACK    LD  A,(COLUMN)
        OR  A
        RET Z               Return if column 0
        DEC A
        LD  (COLUMN),A
        LD  A,7FH           Pass 'erase' to printer.
        JP  PRINT
RETURN  LD  HL,COLUMN)
        LD  (HL),0
        JP  PRINT
```

Listing 4. Code converter.

```
PRINT   PUSH AF
L1      IN   A,(5F)
        RRC  A
        JR   C,L1           Loop until not busy
        POP  A
        OUT  (1F),A         Output data
        LD   A,0
        OUT  (1F),A         Pulse strobe
        LD   A,FFH
        OUT  (1F),A
        RET
```

Listing 5. PRINT for the 8255.

```
PRINT   LD   BC,0E3BFH
        PUSH BC
        DEC  B
        LD   E,0EH
LOOP    IN   D,(C)
        SRL  D
        JR   C,LOOP         Wait for not busy
        DEC  B
        DEC  B
        OUT  (C),A
        POP  BC
        OUT  (C),E
        INC  E
        OUT  (C),E
        RET
```

Listing 6. PRINT for the Kempston interface.
NOTE: it is advisable to disable interrupts during execution of either print routine.

The 'filter' at the beginning of Listing 4 can be modified as necessary. For example, codes 0E-0F could be printed as controls instead of being converted to queries.

Some relevant entry points and data addresses in the Spectrum system may be useful in constructing the program.

As already noted, 0C10 is the start of the token expansion routine. 5C0E is the low byte of TVDATA, in which data on colour and print position are stored. Using this data is a bit complex. When a 'prefix' code is recognised, it is put into TVDATA(L), and the contents of 5C51/2 are read to determine the current channel. The address of an interpretive routine is set at the address so determined, and the routine returns.

The interpretive routine, in the present context, handles the next output byte according to the contents of TVDATA(L). If the value stored there is less than 16H, a jump is made to 2211 in the Spectrum ROM, to process the qualifying data as a colour instruction. Otherwise, the TAB routine is entered, and the data is processed as a TAB position. The interpretive routine must reset the current channel to the normal print driver.

Where the text to be printed is free from colour changes and tabs, this complication can be avoided.

PRINT DRIVER

The actual print routine is shown in Listing 5 for the 8255 type of interface, and in Listing 6 for the Kempston interface. It is required, in either case, to present the current data on the printer data lines, and pulse 'strobe' when the printer signals that it is no longer 'busy'.

And that, really, is about all that can be said without trespassing too far on copyright preserves, except to point out that the routines discussed will work with code generated by BASIC, but may not work with machine code routines. For example, the Picturesque Monitor and Editor Assembler normally achieve print action by calling 0ECD, the Spectrum routine which prints the contents of the Printer Buffer. To drive an MX80, it is ncessary to change this call to access the special printer driver, which can be relatively simple, since it will only be called upon to handle straightforward codes. Picturesque have a data sheet giving full details.

Interfacing the Spectrum to a printer is thus a slightly complex business, but the details which have been provided should make it possible. . .

NEW HOME COMPUTER CABINET

SPECIAL OFFER £79.95

incl. VAT & Delivery

Especially designed for your home computer system, whilst also accommodating your video recorder.

Our cabinet has been custom designed after intensive consultation with P.C. users to provide maximum comfort and convenience during those long hours spent slaving over a hot micro.

The cabinet includes many features not previously available, and as well as combining the very best aspects of modern furniture design, it provides an ergonomic and practical layout for all the major components of your system.

In fact, we are proud to claim that your cabinet will look perfectly at home in your lounge, dining room, or even bedroom.

32½" height, 36" width, 16¾" depth.

JUST CHECK THESE STANDARD FEATURES

1. Smooth sliding shelf locks in position just where you require it — and neatly glides away after use.
2. The computer shelf has been designed to allow adequate 'Knee room' for you to sit comfortably, and will avoid 'programmers backache'.
3. Adequate room to position tape recorder, printer, disc-drives, etc. alongside computer for ease of use.
4. All cables neatly and safely out of sight, while allowing all units to be permanently connected if desired.
5. Lots of storage space for cassettes, cartridges, books, listing paper, joysticks, etc.
6. With shelves and doors closed, instantly becomes an elegant piece of furniture.
7. The lower shelf accommodates all makes of video recorders currently available.
8. Smart teak effect finish.
9. Supplied as a flat-pack. Very simple to assemble, using just a screwdriver. Full instructions supplied.

TERMS OF OFFER UK Mainland Customers only. Please allow up to 28 days for delivery. Cash with order or charge to one of the credit card accounts specified. Money back guarantee if not satisfied provided the goods are returned undamaged within 72 hrs of the customer taking receipt.

ORDER FORM

Please send me one of your new Home Computer Cabinets as advertised. I enclose my cheque for £79.95 or please debit my

Access/Barclaycard No ..... Name (Block letters) ..... Signed .....

Address .....

.....

.....

MARCOL CABINETS, *PO BOX 69, Southampton SO1 OAJ* *TEL: (0703) 731168*

(24hr answering)

Wendy J. Palmer

CLUB CALL

MANCHESTER SINCLAIR USERS CLUB
Tel: 061-225 6997 or 061-445 6316

This club meets every Wednesday at Longsight Library, 519 Stockport Road, Longsight. The meetings start at 7.30 pm and everyone is welcome, regardless of age or experience. Just turn up or ring up one of the above numbers.

MILTON KEYNES MICROCOMPUTER USERS GROUP
Information Technology Training Centre
Erica Road
Stacey Bushes
Milton Keynes
Tel: 0908-311526

This group has been in existence for about two years and has grown from a handful to 250. The group meets on Tuesday nights from 7.30 pm to 10.00 pm and members help each other devise new programs or iron out problems in existing ones. They also have an organised programme of lectures. Membership ranges from seven to 70 and computer games are as much in evidence as more advanced programs. They would like to encourage more female members but everyone is welcome, so if you want to know more, why not visit the Centre?

TANDY MODEL 100 USER'S CLUB
Remsoft
18 George Street
Brighton BN2 1RH
Tel: 0273-602354

Following the launch of Tandy's new Model 100, this user's club has been formed. It is felt that since the Model 100 is so different to earlier Tandy products, users need their own club, where information can be exchanged. A quarterly newsletter is planned and a discount scheme for software is envisaged. The annual subscription is £12, payable to Remsoft, or send an SAE for more details. Contributions for the first newsletter are welcome — preferably on tape cassette using the text editor.

NORWICH AND DISTRICT BBC MICROCOMPUTER USER GROUP
Department of Electronics
Norwich City College
Ipswich Road
Norwich NR2 2LJ
Contact: Paul Beverley
Tel: 0603-60011 ext 231

Meetings are held at Norwich City College twice a month during term time. They are of two types — the first Tuesday at 7.00 pm in Room B9 are speaker meetings, and on the third Tuesday from 7.00 pm to 9.00 pm in Rooms A3 and A4a are workshops. Subscription is £3.00 for the 1983 calendar year. Students and OAPs £1.50; the first visit is free. By arranging bulk purchases, the club has been able to supply cassettes, discs, disc drives, and the Epson FX80 printer at almost trade prices. The Maths and Computer Studies Department runs evening courses on programming and the Department of Electronics runs courses on interfacing.

SOUTHPORT COMMODORE COMPUTER USERS GROUP
5/19 Huth Street
Labrador
Queensland
Australia 4215
Contact: Bill Fitzpatrick
Tel: 075-320061

This group meets every Monday at 7.00 pm at the Southport State School in Scarborough Street. So if you're in Queensland and you have a Commodore micro, why not contact Bill?

CORBY UNIVERSAL MICRO CLUB
26 North Cape Walk
Corby
Northamptonshire
Contact: P. Wilson
Tel: 0536-742622

This is a well-established gorup meeting twice monthly on alternate Wednesdays and Thursdays at Lodge Park Sports Centre, Corby (licenced bar and snacks available). Membership is open to anyone interested in microcomputing, with reduced subscriptions for juniors and families. A variety of machines are catered for and beginners as well as experienced users are welcome. The club maintains a software library of non-copyright material free to all members and the club has a small selection of hardware/add-ons including a printer. Visits to other clubs and national events are arranged together with in-club demonstrations and exhibitions. A newsletter is published.

NATIONAL COLOUR GENIE USERS' CLUB
Lowe Computers Limited,
Chesterfield Road,
Bentley Bridge,
Matlock,
Derbyshire DE4 5LE
Contact: Richard Peat
Tel: 0629-4995

Lowe Computers formed this national club in April; a special new club magazine 'Chewing GUM' (GUM is the Colour Genie Users' Magazine) was launched at the same time in full colour and fully backed by Lowe Computers. Exciting magazine features include 'Sticky Corners' to help users with programming problems and to give a facility for users to publish their solutions. News on hardware and software will also be published and the magazine will have a complaints column. Readers will be encouraged to develop their own programs and to distribute them through the Genie Users' Magazine's own label 'GUM Boot'. For more details on the club and the magazine write to Richard Peat at Lowe Computers.

SOUTHEND COMPUTER CLUB
128 Little Wakering Road
Great Wakering
Essex
Contact: R. Knight (Membership Secretary)
Tel: 0702-218456

The Southend Computer Club has changed its venue in the Esplanade Public House along the seafront, just past the pier. The club meets every Monday evening from 7.30 pm to 10.30 pm and the annual subscription is £6 for adults and £3

for junior members under 14 years of age. A wide range of micros are brought along to the club and many experienced members are there to give advice. Lectures on various subjects are arranged and films as and when they are available. The club has been established for three years and was formally known as the South East Essex Computing Society. They have held basic programming classes in the past and will arrange them again in the future if members are interested. New members are always welcome; go along and see the many types of micros before you decide to buy one.

FORTH INTEREST GROUP
PO Box 1105
San Carlos
CA 94070
USA
Tel: (415) 962 8653

FIG is a worldwide organisation devoted to the dissemination of information about the FORTH computer language and has embarked on a worldwide campaign to double its current membership of over 3500. FORTH Dimensions, the bimonthly non-profit publication of the FORTH Interest Group, will include special articles on music, graphics, voice synthesis, project management, FORTH in the laboratory, the history of FORTH and more. A one year subscription to FORTH Dimension is included in FIG membership of $15 (USA) and $27 (foreign)

ACORN COMPUTER USERS CLUB
BP 325
1000 Brussels 1
Belgium
Contact: Jean-Louis Meerts

The BBC Micro's fame has spread to Belgium; this club issues a monthly newsletter full of information and programs and has bimonthly meetings in Brussels and several provincial cities. Should some readers be interested, the Club will send a free copy of the latest newsletter: just send your name and address to the above address.

EAST LONDON AMATEUR COMPUTER CLUB
63 Millais Road
London E11 4HB
Contact: Jim Turner
Tel: 01-558 3681 (or Fred Linger, Chairman, 01-554 3288)

The club meets on the second and fourth Tuesdays of the month in the main hall of the Harrow Green Library, Leytonstone. They encourage people to visit them and to ask questions about the club and membership; visitors are requested to sign the Visitor's Book, apart from that there is no charge, although tea is extra. Meetings start at 7.00 pm and end at 10.00 pm. The first meeting in the month they try to have a lecture or talk on micro orientated subjects, the second meeting is a 'free' evening giving more of a chance for members to talk about problems, ideas and show off their 'toys' and clever programs. Micros of all types are shown; the last Thursday in the month is a special meeting for users' groups and lectures on BASIC for members. Same meeting place, but another room, upstairs. The normal meeting hall is on the ground floor and has easy access for disabled or elderly members to attend. The membership fee at the moment is £5 per year, half for juniors and OAPs; if members of other clubs would like to visit they are welcome.

HX20 USERS GROUP
25 Sawyers Lawn
Drayton Bridge Road
Ealing W13
Contact: Terence Ronson
Tel: 01-998 1494

This group has been set up in order to bring together the wide spectrum of Epson HX20 owners and users. Terence is gathering ideas constantly from the response he has had from other users and he will produce lists of programs, information, tips and advice in a monthly newsletter. The group is a non-profit organisation whose aim is to introduce users to this machine by pooling ideas. He has also had quit a lot of contact with users outside the UK and it is hoped that the group can help these people obtain software not available to them in their own countries.

NATIONAL ASSOCIATION OF VIC-20 OWNERS
20 Milner Road
Sherwood
Nottinghamshire
Contact: Stuart N Tomanek

For a membership fee of £6 members are entitled to services of the club such as a software library, monthly newsletter and software exchange service. In addition they offer advice to members on all matters concerning their machines.

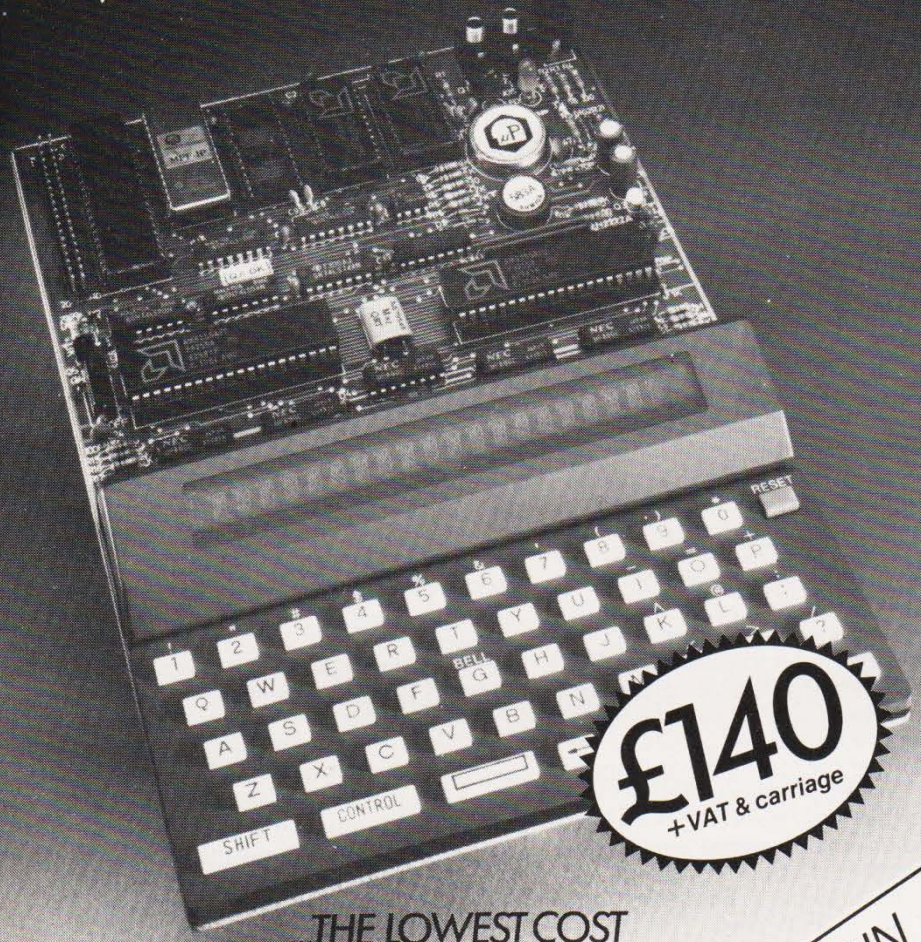
Readers should note that the Computer section of the Cornish Radio Amateur Club has a new Secretary: Mr S. T. S. Evans of Glengormley, Carnon Downs, Truro, Cornwall TR3 6JY.

Anyone that has an Oric and would like to get in touch with other users in the Harrow area should write to James Shields at 182 Welbeck Road, West Harrow, Middlesex, phone: 01-864 4360.

Mr J. R. Griffin would like to get in contact with other Dragon 32 users in the Slough area, with a view to forming a users' club. So anyone interested should write to him at 1 Garrard Road, Britwell Estate, Slough SL35 26X.

**Club Call,
Computing Today,
No. 1, Golden Square,
London W1R 3AB.**

THE NEW MPF1 PLUS...



£140 +VAT & carriage

...THE LOWEST COST Z80 SINGLE BOARD COMPUTER AVAILABLE WITH ALL THESE FEATURES!

BUILT-IN ASSEMBLER

The MPF1 PLUS incorporates the Z80 – the most widely used 8-bit microprocessor in the world, to form a Single Board Computer (SBC). Packed in a plastic bookcase together with three comprehensive manuals and power supply (to BS3651 standard), the MPF1 PLUS is a microprocessor learning tool for every application.

Teaching you in a step-by-step method the MPF1 PLUS helps the user fully understand the Software and Hardware of a microprocessor easily and conveniently – as opposed to micro-computers that aim to teach high-level languages instead of microprocessor systems fundamentals.

Not only is the MPF1 PLUS a teaching tool but with the available accessories it can also be used as a low-cost development tool or simply for OEMs.

FLIGHT Electronics Ltd.

Quayside Rd, Southampton, Hants SO2 4AD. Telex 477793. Tel. (0703) 34003/27721.

Micro-Professor is a trade mark of Multitech Industrial Corporation. Z80 is a trade mark of Zilog Inc

THE MPF1 PLUS

Just look at the specification:-

Technical Specification

CPU: *Z80A – 158 instructions*
Software:
- *Z80/8080/8085 machine code*
- *Z80 Assembler, line and 2 pass.*
- *8K BASIC interpreter (Extra)*
- *8K FORTH (Extra)*

ROM: *8K Monitor (full listing and comments)*
RAM: *4K CMOS (2 x 6116)*
Input/Ouput: *48 system I/O lines*
Speaker: *2.25" coned linear*
Display: *20 character 14 segment green phosphorescent*
Expansion:
- *Socket for 8K ROM*
- *Cassette interface*
- *Connectors 40 way, complete CPU bus*

Keyboard: *49 key. Full "QWERTY" real movement good tactile feedback*
Batteries: *4 x U11 for memory back-up (batteries not included)*
Serial Interface: *165 baud for read/write via audio cassette*

Manuals

1. *User's Manual. 8 chapters.*
   1. Over view and Installation.
   2. Specification (hardware and software). 3. Description of Operation. 4. Operating the MPF-1 Plus. 5. 44 Useful Sub-Routines.
   6. The Text Editor.
   7. Assembler and Disassembler.
   8. System Hardware Configuration.
2. *Experiment Manual. 16 experiments.*
3. *Monitor Program Source Listing with full commenting.*
4. *Also available the MPF-1 Plus Student Work Book (self-learning text).*

Accessories

- ***PRT-MPF-1P:*** *20 character printer. Ready to plug in. Memory dump.*
- ***EPB-MPF-1P:*** *Copy/list/verify 1K/2K/4K/8K ROMS. Ready to plug in.*
- ***SSB-MPF-1P:*** *Speech Synthesizer. Inc. 20 words and clock program. 1200 words available.*
- ***SGB-MPF-1P:*** *Sound Synthesizer Board.*
- ***I/O – MPF-1P:*** *Input/output board*

Yes! I now realise that I need an MPF1 PLUS and that it is the lowest cost Z80 SBC available with all these features.

I enclose £165.00 (£140.00 + £21 VAT plus £4 carriage). Overseas P.O.A.

Cheques payable to FLIGHT ELECTRONICS LTD.

Please debit my Barclaycard/Access VISA ☐ Access ☐

Account No.

An invoice will automatically be sent.

*Name*_____

*Address*_____

_____

_____

*Signature*_____

*Date*_____

CHRISTINE COMPUTING

TI 99/4A

BLAST IT

Disarm bombs with a wire guided robot, bombs must be disarmed in order within a time limit. 5 levels of play, high score table. **£4.95**

CORE!

A three part game, move left and right as you descend through the clouds, birds and aeroplanes. On to the caves and caverns then through the 3-D maze to the Core. **£4.95**

FRUIT MACHINE

One arm bandit simulation featuring random hold, nudge and gamble. 'Reels' are fixed and can be learnt but motion is random. **£4.95**

UFO

Blast the aliens as they appear from hyper-space, features 3 levels of play, refuel etc. **£4.95**

SPUDZ

Score points as you collect potatoes without getting caught in the shadow cast by the setting sun or falling into your trench. **£4.95**

COMMODORE 64

MUNCHER

100% machine code game making good use of the sound and graphic facilities of the 64. Munch your way around the maze avoiding the ghosts, etc. **£6.95**

Send Cheques and Postal Orders to:

CHRISTINE COMPUTING, 6 FLORENCE CLOSE, WATFORD, HERTS WD2 6AZ
Tel: Watford (0923) 672941
Please add 55p P&P per order.

DEALER CBM MICRODEALER

DEVON

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

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

ZIPPY ELECTRONICS
West Dorsets Nascom Dealer
Business and Personal Computer Systems, Printers, Software. Tel: Bridport (0308) 56539
Mail Order Service operated. Access/Barclaycard accepted.

HUMBERSIDE

TOMORROWS WORLD
The sole Nascom Agent for North Humberside.
15 Paragon Street, Hull HU1 3NA.
TEL: 0482 24887

NORTHERN IRELAND

NEWBURN BALLYCARRY CO ANTRIM
6 days until 8 pm. Gemini, Galaxy, Nascom, Acorn Computers. Discs, monitors, printers, servicing, hire. Industrial Control. Accounts, Word Processors.
WHITEHEAD 78330

SUFFOLK

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

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.

LANCASHIRE

COMMODORE BUSINESS SYSTEMS
Commodore Home Computers
Software and Hardware
01 - 228 1637
2/4 Oxford Road, Manchester M1 5QA (Opposite BBC)

LONDON

MAYFAIR MICROS
for a full range of Commodore equipment at very competitive prices. 5th Floor, 65 Duke Street, London W1. Tel: 01-629 2487

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, ICM, 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.

COMMODORE 64

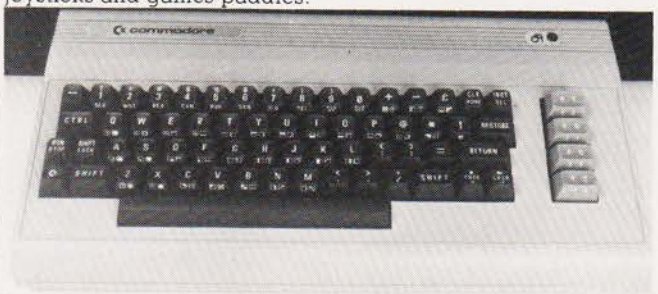
MEMORY	256K 20K ROM
LANGUAGE	Commodore BASIC
CASSETTE	300 baud
DISC	Twin in-built floppy drives
KEYBOARD	QWERTY ☑ CURSOR ☑ NUMERIC ☑ FUNCT ☑
DISPLAY	TV ☐ MONITOR SUPPLIED ☑
INTERFACE	PARA ☑ SERIAL ☑ BUS ☐
GRAPHICS	BLOCK ☑ USER ☐ LINE ☐ RES 80 by 25 COLOUR 16 TEXT 80 by 25
SOUND	**Three channels**

Notes. The Commodore 720 is the top model in the 700 range of business machines. It is built round the 6509 processor, but there is a dual processor (Z80 or 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. The dual disc drives are built-in to the main housing and use DMA transfer, increasing speed.

COMMODORE 720

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.

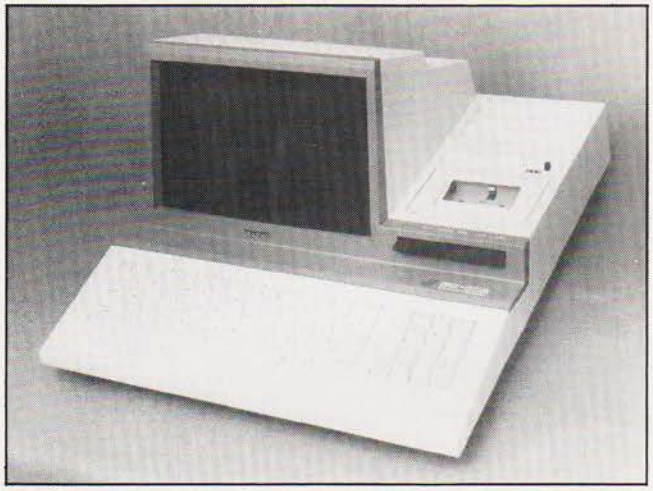


SHARP MICRODEALER

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.

COMPUTING TODAY

Lineage: 35p per word.

Semi display: £8.00 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:
MARK BECAREVIC
ASP LTD, 1 GOLDEN SQUARE, LONDON W1

ACCESSORIES

DUST COVERS

Vic 20/64 Deagon 32 164 Atari 800 1600XL BBC A & B, Electron, Spectrum Cass Unit C2N (old model) **C2N1530** (new model) **& Atari 410/1010** in natural quality vinyl. Just send **£2.95** (Spectrum £1.50) or **£3.95** to inc cass. cover, stating computer and cassette mode. Cassette cover only **£1.50** to:

ALLEN ENTERPRISES
Freepost, Dept CT, Luton LU2 8BR

BLANK CASSETTES!

TOP QUALITY PROFESSIONAL BRAND COMPUTER/AUDIO CASSETTES AT BUDGET PRICES

Packed in boxes of 10 cassettes complete with labels, inlay cards and library cases. Prices include VAT post & packing

Length	Box Price (10)	Qty	Amount
5 min	£4.85		
10 min	£4.90		
12 min	£4.95		
15 min	£5.00		
30 min	£5.20		
60 min	£5.80		
90 min	£7.50		

Cheque/Postal Order enclosed for £
NAME .....
ADDRESS .....
.....

PROFESSIONAL MAGNETICS LTD
Cassette House, 329 Hunslet Rd, Leeds.
Tel: (0532) 706066
TRADE ENQUIRIES WELCOME

CHILTERN ELECTRONICS

Amazing Value in Video Monitors
Professional quality equipment by AGC Corp ideal for your micro. Look at these features: ★ Superb resolution for graphics, 80 col. etc ★ Bandwidth 10hz-22Mhz ★ Green Screen P31 phorsphor ★ Antiglare tube ★ Attractively styled case ★ Composite video mains operation. Available in 9 " Model HM911 **£85.** 12" Model HM123 **£89.**

FANTASTIC KEYBOARD BARGAIN!!
Scoop purchase allows us to offer a 49 key QWERTY layout, ASCII encoded keyboard. Fully cased, used but guaranteed working. Full logic diagrams and information supplied. **ONLY £15.** Add carriage £1.85 and VAT. Educational discounts. Access welcome. Same day despatch. Call Gary Kent on 02407 71234. **High Street, Chalfont St Giles, Bucks HP8 4HQ. Telex: 262284**

MRL PRINTERS

EPSON Dot Matrix
RX80 Tractor (100 cps) **£248 + VAT**
FX 80 F&T (160 cps) **£370 + VAT**
Plus Star & Juki Printers available from **£235 + VAT** Paper, Ribbons & Refills.
RING 0506 31605 FOR DETAILS

MRL Printer Interfaces
Available to fit most micros. Electron, Spectrum, Atari, Commodore 64 incl. FREE Cable **ALL AT £39.95 inc.**
RING 0506 31605 FOR DETAILS
FREE DELIVERY

Selection of Disc Drives at bargain prices from **£135** (inc VAT) to fit BBC, Tandy, Nascom etc.

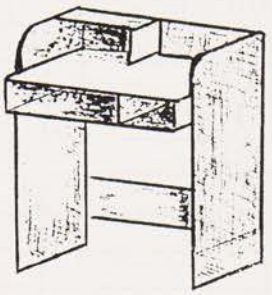
Micro Research Ltd (Freepost), Industrial Unit 6, Knightsridge East, Livingstone EH54 8RA, West Lothian.

Home Computer "CONSOLE" in teakwood finish

Simple self-assembly to house your:

computer, VDU/TV, cassette player, tapes, printer, magazines

SAFELY.



TO ORDER: Send **£49.00** plus £6.00 p&p (delivery 21 days) to:

FORDBURY PRODUCTS
PO Box 20, Hertford SG14 2LW

BOOKS AND PUBLICATIONS

PARAPHYSICS Journal (Russian Translations); Psychotronics, Kirlianography, Heliphonic Music, Telekinetics, Computer Software. SAE 4 x 9". Paralab, Downton, Wiltshire.

FOR SALE

NASCOM-2 cased with 48K RAM B AVC and extensive software. Offers over £550. Waterlooville 52314.

SERVICES

ELIMINATE FAULTY CASSETTES

DataClone is the first company in the UK established specifically for the duplication of data cassettes.

All other duplicating houses are audio orientated — only DataClone has a duplicating system designed from scratch purely to handle computer information.

The result?

Greatly improved reliability in data transfer rates from 300 to beyond 2400 baud — previously unattainable.

All formats catered for. Quantities from 100 to infinity.

Contact us now for brochure.

DataClone — the first specialist service for computer cassettes.

DATACLONE
Unit 1, Roslin Square, Roslin Rd., Acton, London W3.
Tel: 01-723 0477 Telex: 21879

WHICH COMPUTER? Hobbyist or Business. For individual assessment from non-dealer forward £3 together with name and address to: C & S Advisory Services, 144 Sutcliffe Avenue, Grimsby, South Humberside DN33 1AP.

ASP
01 - 437 0699

HOLIDAYS

ADVANCED COMPUTING HOLIDAYS

For Teenagers 12-18
Expand your knowledge and quiz our experts at 2 modern computer centres.
- Advanced BASIC software development
- Introduction to high-level languages
- Main frame and microcomputers

Please send me your Adventure Holiday brochure for 6-18's
Name:
Address:
Postcode:
Tel: Age:

PGL ADVENTURE
Free colour brochure.
Tel: (0989) 63511 or 64211
24 hr ansaphone
PGL Young Adventure Ltd,
689 Station Street, Ross on Wye HR9 7AH.

LIBRARIES

THE MZ80K/A SOFTWARE LIBRARY

Choose from 150 titles. These include exciting Arcade/Adventure games and Educational/Business applications. Once only membership fee. **£5.00**. Hire charges only **90p** per 14 day period. Plus your first 5 games FREE. Send now for free illustrated catalogue to:

The Yorkshire Software Library, 13 Park Top, Pudsey, Yorkshire LS28 6BY.

CARTRIDGE CITY will rent Atari 400 and 800 owners cartridges at low rates. Discounts on software purchases. Details from Cartridge City, 25 Gaitside Drive, Aberdeen AB1 7BH.

SOFTWARE EDUCATIONAL

ISIS VIDEO

Specialists in Educational Software, Software and Textbooks/Software Packages from ABC to A-Level. Many major publishing houses included in our comprehensive catalogue covering most popular home and school computers. Ring Fiona Brooks now on 01 - 549 9305, or write stating model to **Isis Video (Dept CT), Crown Works, Church Road, Norbiton, Kingston, Surrey KT1 3OB**

SOFTWARE APPLICATIONS

POOLS PREDICTION

"POOLSWINNER"
The most sophisticated Pools Prediction Aid available. Gives probabilities of score draws, draws, homes or aways, based on two databases holding over 20,000 matches (included). The databases are automatically updated as results are entered. This year's results are supplied with the package. Can be used in simple mode, or with parameter adjustments to develop your own unique forecast method. Fully documented, available now for Apple, Spectrum (48K), Dragon, ZX81 (16K), Commodore 64, BBC (B) (others — please enquire).

£15.00 (discs/tapes)

"POOLSDATA"
Complete record of all English Football League matches 1978-83. Team, scores and dates of 10,000 matches held in simple format, ready for your analysis. Starter analysis programs and full documentation included. Available for Apple, Spectrum, ZX81, BBC, Dragon, Commodore series.

5 yrs Data £15.00 — 2 yrs Data £7.50

SELEC SOFTWARE (PCN), 37 Councillor Lane, Cheadle, Cheshire. 061-428 7425

CORTEX — FORTH

For fig-forth with extensions for Cortex Colour Computer. 16K supplied in two 2564 Eproms. Replaces 1st two Basic Eproms.

£35 INCLUSIVE

LOMBARD SYSTEMS
18 Lombard Street, Lidlington, Bedford MK43 0RP.

DRAGON ADVANCED PROGRAMMER'S PACKAGE

The most comprehensive FORTH implementation on the market, suitable for 32K and 64K. Includes the language itself which can access Basic commands, powerful text editor, 6809 macroassembler, extensive documentation and the fig-FORTH Installation Manual. **Plus:** Dragon Companion Book containing full 6809 dissassembler, extensive memory map, many tips to get the most from your Dragon. Excellent value at only **£15** inc.
M & J Software, 34 Grays Close, Scholar Green, Stoke-on-Trent. (0782) 517876

COMMODORE 64. "Spriter". Enables single or multicolour sprites to be designed interactively, complied and saved to tape. Tape £4.60. 43 Chanctonbury Drive, Shoreham, Sussex BN4 5FR.

TRS-80 MODEL I
1500 BAUD CASSETTE SPEED-UP UTILITY

Load and Save Programs at 3 times normal speed!

- No hardware modes needed
- Works on BASIC & SYSTEM programs
- Displays program as it loads
- Programs can be up to 14.5K long (16K Level II only)

£7.50 inc p&p to:

D JACKSON (Software)
37 Sandhurst Road, Bexley, Kent.
Tel: 01 - 304 3991

NEWBRAIN

The professional micro computer for the price of a hobby machine.

EXPANSIONS, CP/M IN STOCK

Delivery free. Call us for very special offer. Mail Order. Access.

Call **STEVENAGE (0483 812439)**

anytime for the latest details of Newbrain Hardware, Software, Books. "Newbrain Dissected" Book **£9.00**, tape **£9.95** and both **£18.45**. "Getting More from your Newbrain" **£7.50**. All include p/p. Epson, CP80, Juki & KDC printers, monitors & recorders. Chess **£14.95**.

ANGELA ENTERPRISES
4 Ninnings Lane, Rabley Heath, Welwyn, Herts AL6 9TD.

SOFTWARE GAMES

GROVE SOFTWARE

Games for the Jupiter Ace with 16K

TBT 1 Mastermind + Bomber
TBT 2 Pairs + Submarines

£5.50 each inc p&p

SAE for Catalogue

Cheques and PO's to:

GROVE SOFTWARE
2 Grove Court, Oakfield Road, Penge, London SE20 8RG.

TO ADVERTISE YOUR SOFTWARE GAMES PHONE 01-437 0699

ASP CLASSIFIED
01 - 437 0699

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)

VISA — We welcome Access

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. .....

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.

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

DORSET

ZIPPY ELECTRONICS
West Dorsets Nascom Dealer
Business and Personal Computer Systems, Printers, Software. Tel: Bridport (0308) 56539
Mail Order Service operated
Access/Barclaycard accepted

HERTFORDSHIRE

NEWBRAIN SPECIALISTS
HARDWARE & SOFTWARE
Printers. Epson, Shinwa, Juki etc. Monitors, Tape Recorders. Books. Expansions. CP/M. **Mail Order, Access.** 2 new books — see classified ad for details. Chess in stock **£14.95**
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

NORTH KENT

MEDWAY COMPUTERS
141 NEW ROAD, CHATHAM.
Tel: 0634 826080
Open Mon-Sat 10 - 5. Closed Weds.
Most computers and software stocked.

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.

BUG-BYTE AUTHORISED DEALER
LEIGH COLOUR LABORATORY LTD
87 Chapel St., LEIGH, Tel: 0942 607661
Open: 9 — 5.30.

NSC COMPUTER SHOPS
29 Hanging Ditch, Manchester.
Tel: 061 832 2269
Open: Mon-Fri 9.30am-5.30pm.
Sat 10-5. Retail and Wholesale.

LINCOLNSHIRE

SHARP CENTRE
16 Melville Street, Lincoln.
Tel: Lincoln 32379.
Open: 9am-5.30pm closed Wed.

LONDON

HENRY'S COMPUTER SHOP
404-406 Edgware Road, London W2 1ED.
Tel: 01-402 6822
Open: 6 days a week. Order by phone or call in and see for yourself.

A. J. Duchesne
(Computer Consultants) Limited
Specialists in Small Business Computing
10-12 Creechurch Lane, London EC3A 5AY
Telephone: 01-621 0433

LOOKING FOR MICROCOMPUTER HARDWARE OR SOFTWARE? LOOK NO FURTHER THAN COMPUTAMART!

SOUTH LONDON

CROYDON COMPUTER CENTRE
CCC
29a Brigstock Rd., Thornton Heath Surrey. Tel: 01-689 1280
BBC, Acorn, NewBrain, Genie, Oric, Kaga Microvitek Zenith Monitors, OKI 80, 82A + 84 Printers. Paper Ribbons, Software etc. 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.

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

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

STAFFORDSHIRE

COMPUTER CABIN
24 The Parade, Silverdale, Newcastle.
Tel: 0782 636911
VIC-20, ZX81 hardware and software. BBC, Atom, Spectrum software.

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.

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

WALES

SIR Computers Ltd.
91 Whitchurch Road, Cardiff.
Tel: 0222 - 21341
BBC ACORN COMPUTER

STEVE'S COMPUTER COMPANY LTD
CASTLE ARCADE, CARDIFF
Tel: 0222 — 371578
OPEN: Mon — Sat. 9 — 5.30
FOR COMPONENTS TO COMPUTERS

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

CARVELLS OF RUGBY LTD
3/7 Bank Street, Rugby CV21 2QE
The Acorn/BBC Specialists
We also stock Texas T199, VIC 20, Dragon 32, Commodore 64, Sinclair ZX81/Spectrum, Oric, Atari, Disk drives, Epson, Juki, Shinwa & Olivetti Printers and Books
Tel: Rugby 65275/6

WARWICKSHIRE

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)

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!

Unlock Your Imagination

SCOPE

Computer Graphics Language

- ★ A VERY POWERFUL, TRUE MULTITASK LANGUAGE
- ★ AS FAST AS MACHINE CODE
- ★ SIMPLER THAN BASIC

Write Machine Code in a fraction of the time currently required

SCOPE is available from most good quality dealers and selected branches of WHSMITH

For details of how to get your games published commercially contact ISP Marketing

PRICE £11.95

TECHNICAL DETAILS SCOPE is a fully structured multitask language specifically for writing *graphics, animation* and *sound.* Being fully compatible with BASIC it is ideal for writing both arcade and adventure style games. Additionally, with SCOPE present in high memory it can be used as an assembler with SCOPE words as plain language mnemonics. Therefore no knowledge of machine code is required. The language is extremely easy to comprehend and is very powerful indeed providing many features not available from BASIC.

SCOPE HAS TO BE USED TO BE BELIEVED

48K SPECTRUM

CLIP THE COUPON NOW AND UNLOCK YOUR IMAGINATION

Post to: ISP Marketing Ltd, Crown House, 38b High Street, Godalming, Surrey
Tel: Godalming (04868) 24151 Please rush me ..... copies of the SCOPE computer graphics language tape and instruction manual at £11.95 (p&p free).
(Cheques payable to ISP Marketing Ltd)

Block capitals please

Name

Address

Special offer this issue only. Your Spectrum readers deduct **£2.00** *per tape.*

A

GIVE YOUR MICRO THE SOFTWARE IT DESERVES!

BRITISH BROADCASTING SOFT CORPORATION

These new programs from BBC Publications have been specially developed to exploit to the full the great versatility of the British Broadcasting Corporation Microcomputer.

Each pack contains a program cassette* and a comprehensive instruction booklet.

Making the Most of the Micro

Over 25 great programs from the BBC tv series, designed to put your Model B through its paces. They include a flying, buzzing bee; a buggy searching a maze; and graphics tablet software.

Book and software pack
£12.95

VU Type

An ingenious program that teaches you to touch-type without the need for a book, using exercises approved by Pitmans.
£16.10

Motorists' Log

Keep tabs on the cost of owning and running your car with this ingenious program from the Consumers' Association. It includes checks on fuel consumption and tyre wear, and advance warning of regular major expenses.
£10.00

Beyond Basic

A book and software pack explaining and demonstrating assembly language programming using the Model B's built-in BASIC assembler.

Book £7.25
Software Pack £11.50

Published jointly with National Extension College Trust Ltd.

White Knight Mark 11

The amazing chess program that won joint first prize in the home computer section of the P.C.W. Microcomputer Chess Championships.
£11.50

Structured Basic

This book and software pack will help you extend your knowledge of BASIC. Richard Freeman explains how to write well-structured programs which are crash proof and easy to amend. The software illustrates a wide range of programming techniques.

Available shortly

Published jointly with National Extension College Trust Ltd.

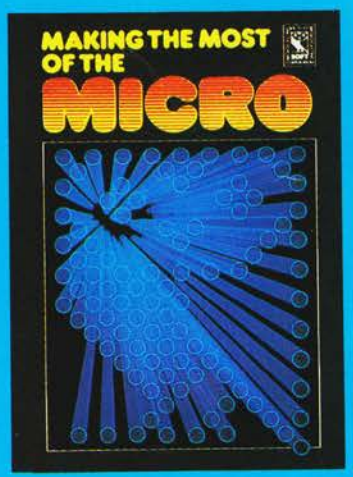
Record Keeper

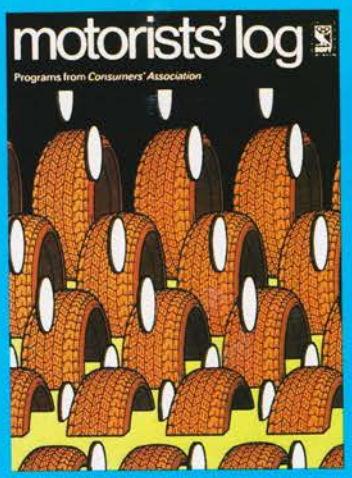
A very useful program enabling householders and small businessmen to keep track of their lists.
£13.80

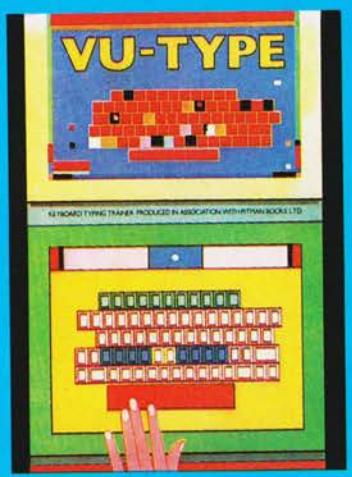
Ultracalc

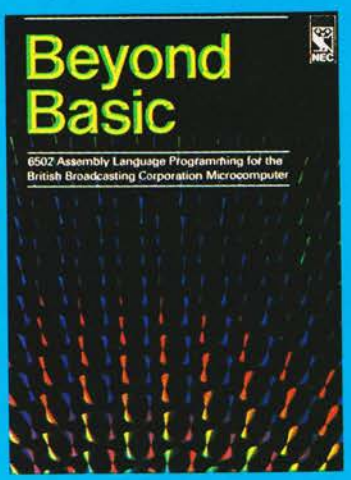
An extremely powerful electronic spreadsheet program. Enter figures and labels on the grid – rapidly total along rows or columns – alter some figures and see the results. Ideal for engineers, scientists, accountants, and everyone who needs ultra-fast calculations.

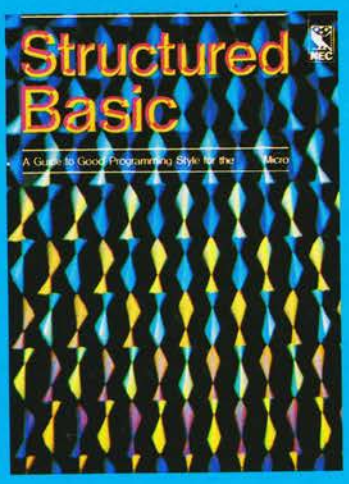
Firmware* £74.75

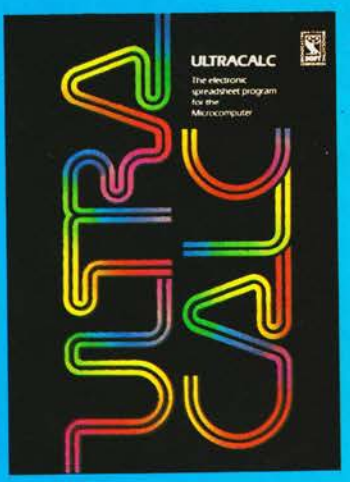












All packs are designed for the British Broadcasting Corporation Model B Microcomputer, operating system 1.0 onwards. Motorists' Log, VU Type and most of Beyond Basic will also run on 0.1. Prices include VAT. *Ultracalc is supplied on a ROM chip which should preferably be installed by a dealer.

FROM BOOKSELLERS AND MICRO DEALERS