AN ARGUS SPECIALIST PUBLICATION

MAY 1985 £1.00

COMPUTING Today

The National Computing Centre

Shedding light on LCDs

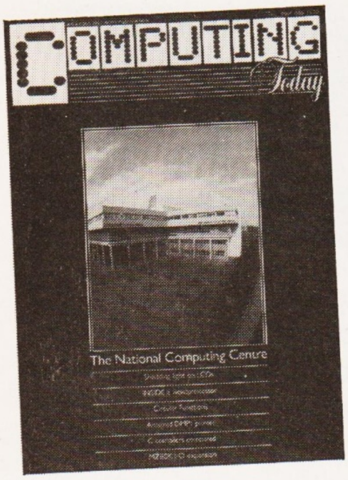
INSIDE a wordprocessor

Circular Functions

Amstrad DMP1 printer

C compilers compared

MZ80K I/O expansion



COVER

The National Computing Centre was established in 1966 to promote an increased and more effective use of computers in every field of National and commercial activity. Now an independent limited company, NCC provides many services to industry an government departments alike, including training, exhibitions, publishing, and a host of other information-related utilities. Cover story, page 34.

Editor: Don Thomasson
Assistant Editor: Jamie Clary
Technical Illustrator: Jerry Fowler
Additional Illustration: Grant Robertson
Advertisement Manager: Anthony Shelton
Advertisement Copy Control: Sue Couchman, Lynn Collis
Publishing Director: Peter Welham
Chairman: Jim Connell
Origination and Design: Design International
Cover Design: MM Design

Member of the Audit Bureau of Circulation
ISSN 0142-7210

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

Subscription notes; UK (£16.20) including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, Infonet Ltd., Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB England (phone 0442 48432).

CONTENTS

VOL 7 NO 5 MAY 1985

REGULARS

SERIES

GENERAL FEATURES

SPECIAL FEATURES

EDITORIAL & ADVERTISEMENT OFFICE: 1 Golden Square, London W1R 3AB. Telephone: 01-437 0626. Telex: 8811896.

Subscriptions

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

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

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

COMPUTING TODAY Subscriptions,

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

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

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1985 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

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

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

SUBSCRIPTION RATES (tick ☐ as appropriate)

£16.20 for 12 issues UK ☐
£18.70 for 12 issues Overseas Surface ☐
£51.20 for 12 issues Overseas Air Mail ☐

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

BARCLAYCARD VISA

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) ....
delete accordingly
ADDRESS ....
....
.... POSTCODE ....
Signature ....
Date ....

CT May '85

PHOTOCOPIES

A COMPUTING Today Reader Service

To take advantage of this simple service, just fill in the required information and send it (or a photocopy) to:

COMPUTING TODAY Photocopies,
No. 1, Golden Square,
London W1R 3AB.

together with your money and we'll do the rest!

CT PHOTOCOPIES

NAME ....

ADDRESS ....

....

.... POSTCODE ....

Please send me Photocopies of the following items

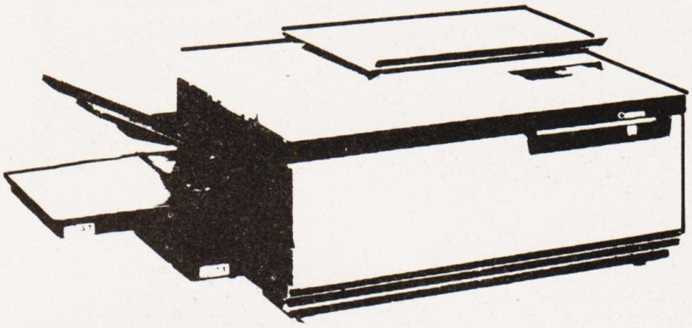
ISSUE	ARTICLE	PAGES

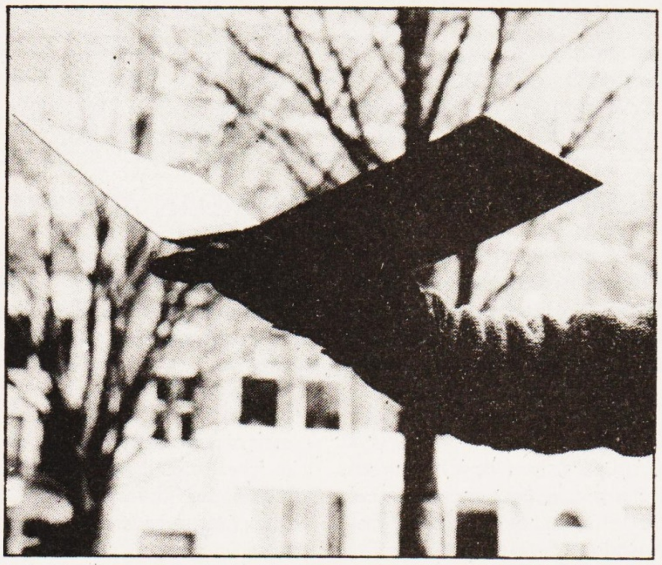
At £1.50 each, I enclose £ ....

Cheques and Postal Orders should be made payable to ASP Ltd.

Lost and can't be replaced? Well, if you've lost one of the early issues that cannot be replaced from our stocks of backnumbers, all is not quite lost.

If you know the article name and the issue it appeared in, we can supply you with a photocopy for the miserly sum of £1.50 including postage and packing. If you're stuck for the month, try checking in our annual index which is included in the December issue each year.





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

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

Cut out and send to:

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

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

Insert card no. | | | | | | | | | | | | | | | | |

Please use BLOCK CAPITALS and include post codes.

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

....

....

Signature ....

Date ....

NEWS

KAYPRO MAKE MICROS A THING OF THE PAST

Although we are becoming increasingly dependent on miros to provide us with the latest information in the office and at home, they have been used surprisingly little to tell us more about the past on excavation programmes etc. All that could change, however, when a team from the Institute of Archeology return from their latest expedition.

The Institute recently set out on a 4-month visit to Thailand, taking with them a Kaypro 4 portable personal computer. They aim to complete their excavation of Ban Don Ta Phet in Western Thailand, where some 60 graves dating back to AD100-200 have been discovered. During their first visit there in 1981, they soon found out that the area was exceptionally rich in discoveries, with literally thousands of glass, pottery and metal relics having been found.

They therefore decided that a portable microcomputer would be an essential aid to complete their research, so that the vast amount of information could be recorded and sent back on disk to the Institute's headquarters in London. They sought the help of the Kaypro Corporation of Solana Beach, California, who run a special scheme which donates machines for scientific research or to underdeveloped countries.

The leader of the excavation, Dr. Ian Glover, explained how they intend to use the microcomputer: "Our biggest problem during the original excavation was the sheer amount of material being discovered — far more than a typial British site. We therefore knew that a portable microcomputer such as the Kaypro would prove indispensable to store the information.

"The other reason why we are taking it is because the soil on this particular site is very poorly stratified. This means that there are not so many clues as to what belongs to what amongst the discoveries. We therefore need to record a lot more detail, and to do so in much more efficient manner than before.

DESIGN COUNCIL AWARDS SCHEME TO INCLUDE SOFTWARE

In response to the increasingly important role of computer technology in UK industry, The Design Council announces the addition of a separate category specifically for the computer software industry in its annual Design Council Awards scheme.

Computer hardware is already included within the Council's existing scheme for engineering products. However, the design of software products is a specialist area too broad and too complex to be considered within the existing categories. "Over the past year there has been a significant upturn in numbers of enquiries from industry regarding the eligibility of software products for Design Council Awards. We sought advice from a selection of experts in the industry, and although it was agreed that the task would not be an easy one, it was considered essential to our design promotion work to include software design", said Tony Key, awards manager.

As with the existing Design Council Awards categories, the aim of the software scheme will be to promote the importance of design to British industry, by identifying and highlighting outstanding examples and by publicising the designers and companies involved.

All products of the software industry are eligible to enter the scheme, as long as the entries have been designed and produced predominantly in the UK. Closing date for applications is Wednesday, 31 July.

A panel of judges drawn from the industry itself will be assessing the entries using a number of criteria including design innovation; attention to design for manufacture — including evidence of configurability, maintainability and testability; appearance factors and ease of use — including display features, man/machine interface and accompanying literature and manuals; performance and reliability in service; market success and value for money.

Applicants will be expected to prove the successful and efficient application of the software in actual use for a period of not less than six months.

For further information and application forms please contact the awards office, The Design Council, 28 Haymarket, London SW1Y 4SU, or telephone 01-839 8000.

OLIVETTI TAKE 49% SHARE IN ACORN

Office equipment giant, Olivetti, has acquired a 49% holding in Acorn, following announcements of huge losses and the subsequent suspension of Acorn shares in February of this year.

This alliance with Olivetti has resulted in considerable restructuring of the company, with founders Chris Curry (ex-Chairman) and Herman Hauser (ex-Managing Director) sliding one or two rungs down the organisational ladder, taking positions as Deputy Chairmen.

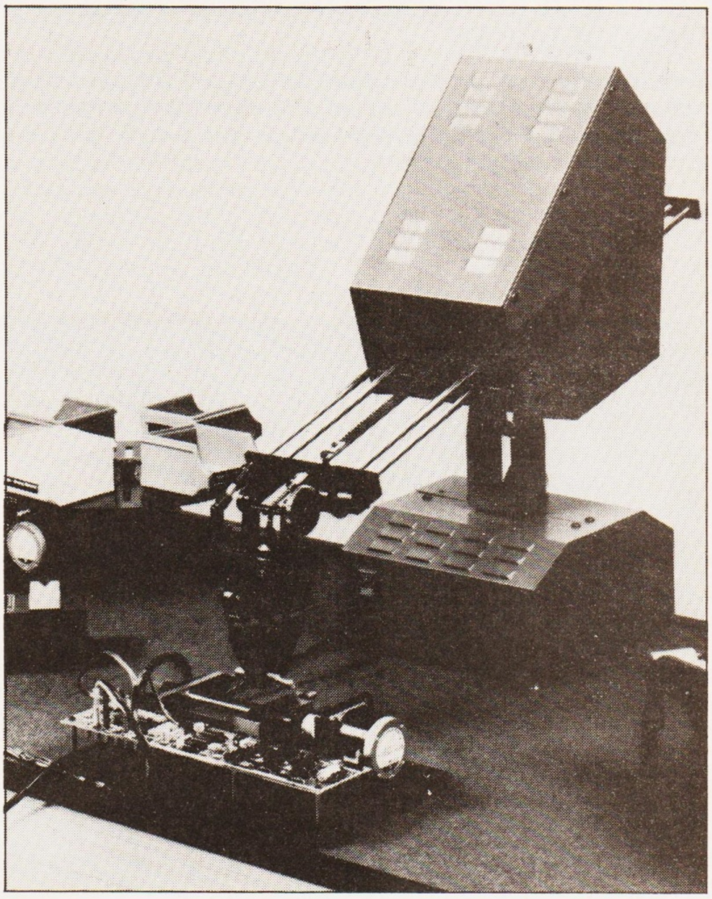
Acorn's leader is now Olivetti's Dr Alexander Reid, who will oversee work in the four divisions established as part of the reorganisation, to wit: Education and Training; Scientific and Industrial; Business; and Consumer.

The alliance with Olivetti is expected to have a significant effect upon Acorn's new ABC machines, which were to be marketed as 'business' micros. However, Olivetti already sell an impressive range of machines for this market, and rumour has it that the ABC's are to be pitched at the Scientific and Technical community.

Interestingly enough, 'spares' for the BBC micro have been in desperately short supply since Christmas and phrases like 'black-market' and 'under the counter' are enjoying a return to popularity, reminiscent of the days when Intel's 8271 disk-controller — as used by the Beeb — was almost impossible to obtain from reputable dealers. Now it'll only cost you an arm and a leg...

ROBOTIC WORK-CELL SYSTEM

Designed for use with their ATLAS robot arm, the new work-cell system from LJ Electronics offers a cost effective means of investigating the working of a computer-controlled robotic work-cell situation



Representing a fully automated parts-selection system, the new work-cell is made up from four hardware modules and a TINA 6502 advanced microprocessor control system.

Intended specifically for education and training usage, the modules fit onto a purpose-made base board unit that ensures their accurate and repeatable location.

The hardware modules that make up the system are:

Parts Dispenser Module-a twin hopper, gravity fed parts dispenser with 'pin' and 'bush' components

Motor Calliper Module-a precision built motor calliper module for performing measurement of overall length, inside and outside dimensions of parts.

Load Cell Module-for selecting parts by material using the weight differentials.

Indexing Table Module-a four-bin rotating carousel for locating parts to specified bins, i.e. right-size, right material; wrong size, wrong material; etc.

The software that controls the system is written for use with the TINA, when used with this comprehensive microcomputer system it gives full computer control over all the elements of the work-cell including the ATLAS robot full colour graphics are used in the on-screen representations of the various work-cell operations.

As with all LJ products the new work-cell is a modular system, built with expansion in mind. The next stage in its development will be an automated assembly facility that will be added to existing parts-selection systems.

COMPUTER HELPS MISSION IN THAILAND

There cannot be many human activities that have not felt the impact of the computer — but an unusual order to supply two computer systems to a mission in Thailand has just been completed by CEMOC LTD., based in Cowes, Isle of Wight.

Robin Griffiths, with his wife Rosemary and their three children aged 14, 11 and 7, will be returning to Thailand next April to continue their missionary work with the Pwo Karen Group based in West Thailand, about 200 miles NW of Bangkok. Their main field of activity is a literacy program and they plan to use microcomputers to produce books and other teaching aids. It is also proposed to use computers for Bible translation.

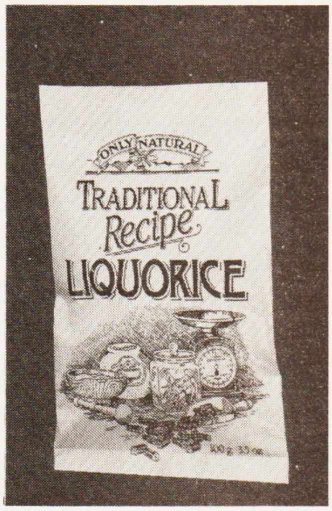
One of the problems with the Pwo Karen people is that their language is quite different to the national Thai language. They have had a written form of their language which is similar to Burmese script, but the knowledge of this is not widespread. In recent years the Thai government has promoted the use of the Thai language and writing system amongst tribal groups so that many of the younger generation are more familiar with this. Mrs Griffiths, over the past 3 years or so, has made a linguistic analysis of the Pwo Karen language and reduced it to written form using the Thai alphabet. Up to now, producing literacy materials to teach this has been a time consuming and laborious job, especially when revisions have had to be made. With a computer, information can be stored, revised and updated any number of times and this will simplify and speed up the work compared to manual methods.

Mr Griffiths selected an ACT Apricot as the microcomputer most likely to fulfill his requirements. The Thai script has 66 characters, plus tone markers. The Thai font was developed on a Sirius computer by a friend of Mr Griffiths working in the N.E. of Thailand. This Sirius disk had then to be transferred over to an Apricot disk and modifications had to be made to enable it to match the Apricot keyboard. The program was designed so that a particular character could be allocated to any key and Mr Griffiths was given a short training course so that he could make changes easily as he finds necessary when he returns to Thailand. A special print (Daisy) wheel was also obtained so that the translations could be printed out using an Olympia printer.

It will be another four years before the Griffiths family returns to this country, but they will keep in touch with CEMOC and report progress. Adrian Medley, Managing Director of CEMOC said, "This was a very interesting project and underlines and adaptability of modern microcomputers, which are very reliable once a system has been set-up. And it is an excellent example of how a computer supplier can work closely with a customer to solve a unique problem. We wish Mr Griffiths every success in his work."

THE EFFECTS OF LIQUORICE ON THE COMPUTER ENTHUSIAST SHOCK

QUESTION: what have liquorice and computers got in common?
ANSWER: not much!



However, we at the Computing Today offices received a bag of liquorice (and very nice it was) courtesy of Only Natural Products Limited. Why they chose to send it to us is anybody's guess — it could be connected with rumours that computerists suffer from certain digestive disorders which liquorice is said to relieve. Only time will tell. Must dash...

NEW PROBLEM SOLVING SOFTWARE FOR COLLEGES

Burning the midnight oil may become a thing of the past as a new presentation of the software program, ***TK!Solver the equation processor,*** is made available to universities and colleges.

Designed to ease the drudgery of number crunching, the TK!Solver College Edition has the ability to back-solve equations; solve iteratively for multiple unknown variables; convert units of mtively for multiple unknown variables; convert units of measurement automatically and produce tables and plots of results.

This program will tackle problems requiring solutions to sets of simultaneous linear and non-linear equations, which have a particular relevance to students and lecturers in science and engineering.

Suitable for both the IBM-PC and the Apple IIe, TK!Solver from Software Arts comes complete with program disks, reference card, instruction and TK!Solver Book with its expanded index.

Price is £59 available direct from Software Arts International, 43 Buttermarket, Ipswich, Suffolk, Tel: 0473 221551.

HRH OPENS ITeC

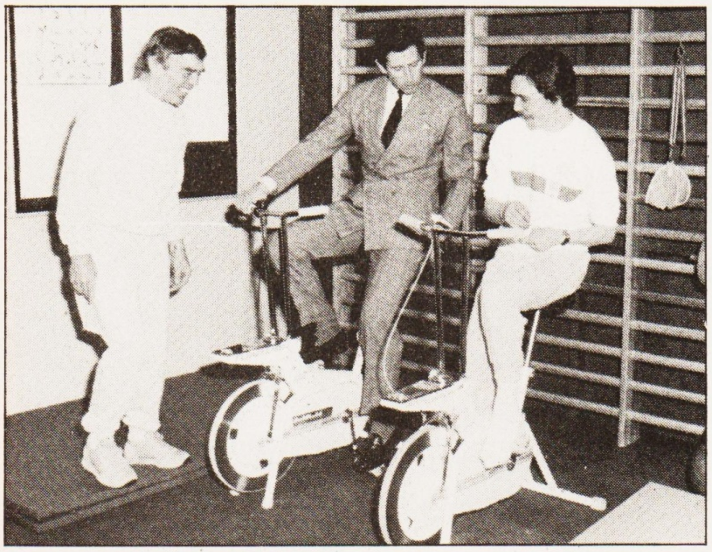
H.R.H. Prince of Wales officially opened the Rank Xerox Information Technology Centre (Slough) in March 1985.

The Rank Xerox Information Technology Centre (Slough) provides a year's training for school leavers, and in addition, concentrates on a special initiative of a flexible programme for disabled adults. These programmes offer to industry and commerce a new pool of youngsters and adults who have been professionaly trained in the basics of information technology. It is a non-profit making collaborative venture backed by Rank Xerox and by Central Government. Local Government, in the form of Royal Berkshire County Council and Slough Borough Council also provide significant support as part of their employment initiative.

The ITeC is staffed with people experienced in the various skills associated with information technology and provides a teacher/trainee ratio better than 1 in 7.

The ITeC has ten training places for disabled adults. Information Technology is bringing to these people new hope of obtaining suitable, useful and satisfying employment. This type of work, with its emphasis on mental rather than physical effort, is often well within the ability of physically disabled people. Being confined to a wheelchair, for example, does not prevent the person from being a skilled operator of a computer or word processor.

Each disabled person is given a tailor-made training programme suited to his or her capabilities, with a specific type of job in view.



NCR FUND MICROELECTRONICS CHAIR

NCR DUNDEE — a world leader in the manufacture of automatic telling machines and other computerised systems in the service of industry and commerce — is to fund a new Chair in Microelectronics at Heriot-Watt University.

The NCR Chair in Microelectronics will be a senior post in the University's Department of Electrical and Electronic Engineering and, to complement the establishment of the professorship, NCR is also donating to the Department computing equipment worth over £50,000 to aid the University's teaching and research activities.

The funding of the professorship and the gift of equipment was announced in Edinburgh by Mr Jim Adamson, General Manager and Managing Director of NCR's engineering and manufacturing company at Dundee.

"The continued success of the microelectronics industry in Scotland is critically dependent upon research and development (R&D)," said Mr Adamson. "Without local R&D our manufacturing base is not self-sustaining and we will get none of the benefit of the spin-off growth characteristic of 'Silicon Valley' in the USA.

"Industry looks to the Universities to provide skilled professionals to fuel this success — but this alone will not be enough for survival, let alone growth, in the next decade! We need the evolution of genuine partnership between industry and Universities if we are to compete effectively with the best the world has to offer.

NCR's support was warmly welcomed by Professor Colin W. Davidson, Dean of the Faculty of Engineering at Heriot-Watt University and Head of the Department of Electrical and Electronic Engineering, who said it was encouraging to receive real help, from a leading company in this field, to aid the teaching and research programmes.

NATURE CONSERVANCY COUNCIL AWARD £40K CONTRACT

G F Marshall Computer Services have been awarded a contract by the Nature Conservancy Council worth over £40,000 for software development for the National Countryside Monitoring Scheme (NCMS).

NCMS will accurately measure land use changes by means of computer comparisons of Aerial Photographs. The earliest photographs date back to the end of the second World War, and subsequent sets bring the data up to the present day. The scheme is intended to run on well into the 21st century and will provide precise answers to such questions as: "How many miles of hedgerow have been lost since 1945?", and "How much coniferous woodland has been planted on previously open moorland?"

G F Marshall Computer Services have completed and installed a similar set of digital mapping software (known as MAPLE) for the Forestry Commission in Scotland. The new software for the NCMS also runs on micro computers and will capture map information photogrammetrically from aerial photographs, or by digitising existing maps. The captured data undergoes automatic computer editing before being displayed on the micro VDU screen in map form. This screen map may be edited further by the computer operator. Individual land parcels are identified by the computer and their land use determined. Using data captured for the same area but 'photographed' at different times, (e.g. for 1945 and 1985), the new software will determine land use changes for selected sites throughout the United Kingdom. Equally important is the identification and measurement of areas where no change has occured. The software will also identify linear features (i.e. Hedgerows, treelines and streams), measure their length, and then compare individual feature changes over the time span. It will be possible to determine the length of hedgerow (for example) which has been totally lost, that length which has survived, and the length of any newly planted hedgerow for the selected site.

The initial results from the NCMS project are due to be published later in 1985. The Nature Conservancy Council intend to use the software developed for the NCMS for a wide range of other projects which require mapping techniques. The NCMS project has aroused a great deal of interest amongst the various local authorities in Great Britain.

Computing Today *will be taking an in-depth look at the MAPLE package in the July edition, so watch this space — Ed.*

MARCONI TRACKER BALL

The Marconi Tracker Ball is a new graphic input device which provides precise x-y cursor control by simple fingertip operation of the central ball. It is more stable and accurate than a joystick, needs less space than a mouse and can also be hand held.

At present Tracker Balls are available for direct connection and operation with BBC, Commodore and MSX micros, priced at £59.50 (BBC and Commodore), £79.50 (MSX).

The RB2 Tracker Ball was developed by Marconi for serious hobbyists, educational establishments, users of graphics programs and incorporates many features of Marconi's Tracker Balls for military and air traffic control applications.

The output from the Tracker Ball is a train of pulses with directional information for each of the x and y axes. Each pulse, generated by an optical encoder, provides an increment of cursor movement enabling precise positional control with zero drift.

The Tracker Ball comes complete with a user guide and sophisticated graphics software as well as software to facilitate linking with RB2 to users own programs. Other software packages are also available from dealers.

BLACK BOX CRYPTON DEFIES THE HACKERS

A data link scrambler known as the Crypton is announced by BLACK BOX aimed at securing the dial-up mini and micro system from illicit access.

BLACK BOX is confident that this hardwired device, priced at £275 per unit, will prove impenetrable to any unauthorised users, even those who know the system passwords. Hardwired into the data link in pairs with one unit situated on the computer side and the other at the terminal end, the device works like an electronic lock, each pair with its own unique code. Only the user who has an identically wired scrambler can receive intelligible data when dialling into the computer.

An additional advantage is the ability to restrict access to certain parts of the database and give users different levels of privilege according to the code of their descrambler.

Comments Doug Prewer, General Manager of BLACK BOX U.K. "With the recent public interest in 'hackers' dialling into installations that had been considered secure, people are more aware of the potential damage to data bases and financial losses. The Crypton provides a unique type of safeguard at a quarter of the cost of typical security "methods."

Using an asychronous interface, the Crypton data link scrambler will protect databases on all types of microcomputers and minicomputers including Hewlett-Packard, Digital, Data General and Prime. Codes are easily set up by the customer, although this can be done by BLACK BOX if required.

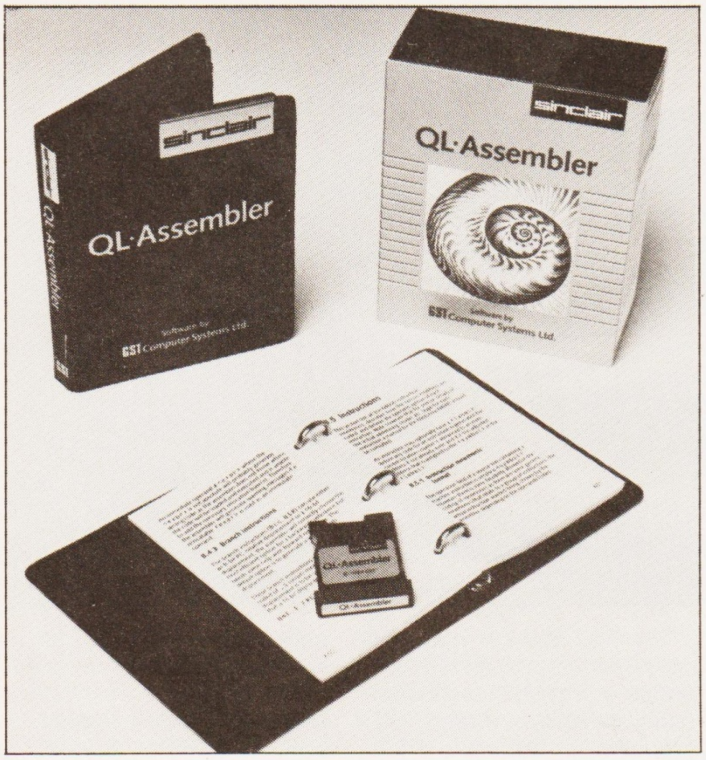
QL ASSEMBLER

Sinclair Research has announced the retail availability of QL Assembler, the machine code utility program written by GST Computer Systems, with an RRP of £39.95 including VAT.

As a non-relocatable, Motorola format-compatible 68000 assembler, the product converts source files written in M68000 assembly language into QL-compatible binary files.

Included with QL Assembler is Sinclair's standard, multi-tasking, full screen editor, (written by Metacomco). Both the editor and the assembler can be run simultaneously with SuperBASIC, allowing users to switch instantanteously between all three.

Commenting upon this latest QL software release, software manager Alison Maguire notes, "QL Assembler will help serious technical users and software houses to extend the range of high-quality applications programs available for the QL."



FLIGHT ANNOUNCE LOW-COST 8088 MDS

FLIGHT ELECTRONICS LTD is launching the MPF 1/88, the latest addition to the company's series of microprocessor educational and development systems.

The MPF 1/88 is a single board computer educational system based on the 16-bit Intel 8088 Microprocessor.

It is designed to teach the fundamentals of 8088-based hardware, or to be used as a low cost development system. Says Max Soffe, Managing Director, Flight Electronics "We believe we have a 'first' with a system at £325 and with these facilities".

In concept, it is analogous to a 'Meccano' set, based around the Intel 8088 — the chip at the heart of the IBM-PC and many other leading 16-bit machines. It allows designers to prototype control devices for robot systems and servo mechanisms, for example: the traditional 'packaged' microcomputers provide no access to the chip for optimal control.

The unique design of the MPF 1/88 allows the user to open the case and look at the system board components. By viewing the single-board CPU, the student is able to see and understand the computer operations at all levels.

The MPF 1/88 is the third generation in a series of microprocessor educational and development systems from Multitech Industrial Corporation, which has already sold more than 70,000 units worldwide in three years; 5000 of them in the UK. The first system developed by Multitech taught the fundamentals of the Z80 and 6502 Microprocessor.

The MPF 1/88's standard software teaches a user how to write and debug his own programs. The system can also be used to learn Assembly language, as well as the fundamentals of I/O interfacing.

Standard hardware includes an Intel 8088 CPU, 48K of RAM and 16K of ROM on a single chip. RAM is expandable to 24K, and ROM to 48K.

The MPF 1/88's LCD display shows any two lines of a 20 character x line logical screen. The last 22 lines are automatically scrolled off the screen and saved in a screen buffer. The LCD display supports 192 different character patterns, each within a 5 x 7 dot matrix. A Centronics parallel interface with a 16-pin connector is supplied as standard. One of the most important features is a 64 pin card-edge connector with a 62-pin expansion bus allowing the MPF 1/88 to interface with many expansion cards designed for use with the IBM-PC. This facility gives RS232C, memory expansion, video colour output and many other facilities common with the hardware of the IBM-PC. In addition, there are routines that support interface to a standard ASCII terminal, a printer driver to support Centronics printers, and asynchronous communications routines. By interfacing with the routines, or integrating them into self-written programs, the user has a strong command of the power of the 1/88.

Other standard features include a cassette tape recorder interface, a 9 volt AC adapter (although an integral switch-mode power supply will also be available) a built-in speaker, and red and green LED status lights.

THE DEMON MODEM

The Demon Modem (£49.95) features auto dial, auto answer, full and half duplex, full baud rates 300/300, 1200/75, 75/1200 and (1200/1200 half duplex only). The unit comes complete with all leads, power supply, operating and reference manuals and fits all RS 232/423 interfaces.

Software is available for the BBC in ROM and features auto dialling for Prestel, Micronet, Telecom Gold, Easylink etc, printer routines, and a remote facility. The ROM costs £20 plus VAT.

Software is expected for Electron, Amstrad, Commodore and Sinclair in the next month.

The Modem is not yet approved, but all components are. Approval is expected in a month.

COURSES AND TRAINING

Education around the U.K.

LATEST FROM OPEN TECH

The Open Tech Programme is a major training initiative that is allowing thousands of key personnel across a wide range of industries to update their skills or learn new ones.

Launched by the Manpower Services Commission in 1982, Open Tech provides start-up funding for other organisations like colleges or professional institutions, to mount projects that help people at supervisory, management or technician levels to learn new skills without disrupting their work.

This involves the production of home study texts, videos, audio tapes and small kits of hardware that learners can use when and where they choose. These are backed by facilities providing access to expert tutors, either on the telephone or in person.

The Programme has a budget of £41m to the end of March 1987, by which time an estimated 50,000 people will be benefiting.

Open Tech is part of the MSC's Adult Training Strategy, aimed at providing industry and individuals with the skills they need for economic success.

As part of this strategy, MSC are urging employers to accept that training is essential if they are to compete with foreing rivals and cope with the many changes facing industry.

A campaign to increase awareness of this important subject is being run by MSC, and at the same time they are updating their own training provision with the aim of more than doubling the number of adults helped to 250,000 in 1986/87.

NEWS NEWS **Open Tech Programme** NEWS NEWS

Spring 1985 — MSC — No 7

One of the many rewarding aspects of working within the Open Tech Programme is to observe the energy, imagination and expertise being brought to bear by many of the Local Education Authorities across the country.

The publication of this issue of *Open Tech News* coincides with a major LEA conference and exhibition on the theme of Open Tech.

During the two days 12/13 March at the Birmingham Metropole Hotel, senior education officers from every LEA in Great Britain will be hearing from Open Tech projects how they are tackling material development, setting up delivery and open learning support systems, and marketing their open learning goods and services to industry and commerce.

For those LEAs receiving MSC Open Tech funding it will be an opportunity to view their own projects in the context of how other projects have approached the issues. For those not yet involved, it will be a chance to assess what Open Tech and open learning have to offer them. Because, whilst those LEAs in receipt of Open Tech funding may be at an advantage, the field is open to everyone with the will to become involved and the expertise to make a contribution.

Inside this issue of *Open Tech News* we take a look at LEA involvement in the Open Tech Programme. *LEA Update* (page 3) emphasises that LEA delivery projects will be providing a service both to material providers and to the end users — the learner and the employer. *Meeting the Client's Needs* (page 2) discusses some of the innovatory methods being adopted by LEA projects in their efforts to provide support at a distance as well as opening up access to college resources and facilities.

Finally (page 4), there is a profile of TAYTEC, an exciting project developing on the banks of the Tay Estuary in Dundee involving collaboration between a diverse range of organisations over a wide geographical area.

Coventry Open Tech Unit – Broadgate Learning Centre

A former shop has turned into a city centre access point for Open Tech. A network of BBC micros, an IBM PC, tape-slide, overhead projector and video are the facilities currently available along with study and tutorial space, an exhibition of open learning materials and, most importantly, a learning centre administrator to welcome and assist customers.

On the first floor above the City Library's Information Centre, it may attract some passing trade, though for practical purposes it will be preferable if users book beforehand.

Evening opening is being considered in case there is demand for study time after work.

Open Tech Goes Mobile

After a few unforeseen difficulties Coventry Open Tech Unit (COTU) has piloted its mobile centre, a vehicle containing a network of microcomputers and

— continued on page 2

COMPUTER BASED TRAINING FOR ENGINEERS

The National Computing Centre (NCC, and the Engineering Industry Training Board (EITB) today signed a contract for the development of a range of video-disc based training materials.

The project is a joint venture between EITB and NCC and a group of companies that are members of EITB. One of the companies, Ferranti Computer Systems, will be loaning Ferranti Advance 86 computers to be used as the basis of this teaching equipment.

Interactive video-disc is one of the most promising new technologies for computer based training. It combines the flexibility of computer-based instructions with the highest quality of sound and vision to assist the learning process. The NCC produced its first interactive video-disc last year to demonstrate the training potential by using this new technique.

This marks the start of a pioneering project that will establish interactive video-disc based training for engineering. The first training modules will be 'Design for Profit', 'Quality Control' and 'Information Processing', they will be designed to improve the competence of engineering in these important non-technical subjects.

Financial support for the project will be provided by the Deparment of Trade and Industry and the training modules will be marketed by EITB and NCC.

NEW COMPUTER COURSES AT HUMBERSIDE

Humberside College of Higher Education is now able to offer a special retraining course for the unemployed: an HND in Computer Studies.

The course will last one year, is being funded by the Manpower Services Commission, and has been approved by the Business and Technical Education Council. It is designed for those who have at least two years work experience, but who are presently unemployed.

Students will come from either an Engineering and Business background, and normally hold an appropriate HNC or similar qualification.

Students will work with a variety of super-micro computers in the College's new UNIX computer laboratory.

The course leader is Alex Ainsworth, School of Information Studies, and you should get in touch with him for further details at Humerside College of Higher Education, Cottingham Road, Hull, HU6 7RT, (Tel: 41451 ext 342/387) or in his absence Janet Martin, Information Centre (ext 397/347).

16-BIT HANDS-ON WORKSHOPS

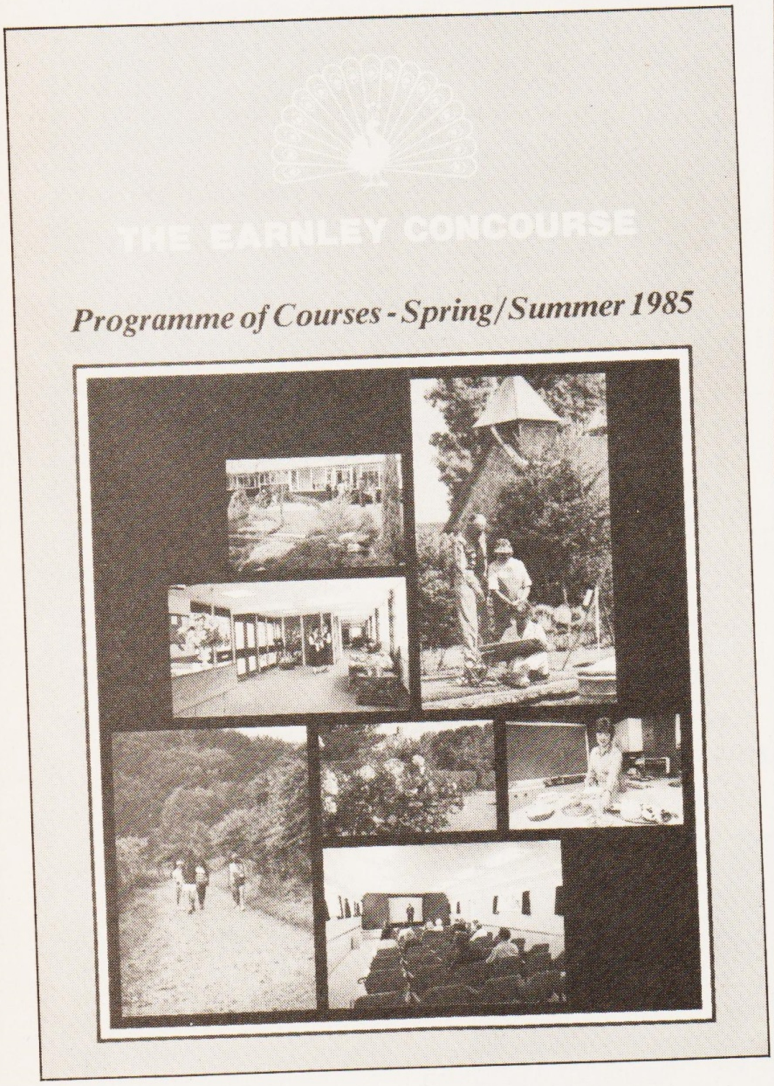
Two very popular workshops run by the Microcomputer Unit in Central London cover the front edge of technology demands.

There is tremendous interest in the 68000 and 8086 families of microprocessors. For these hands-on courses, the Unit has designed and developed special interface cards to investigate the input/output capability of the processor, as there is no other suitable equipment on the market. No sharing of facilities is involved and ineffective demonstratiosn are avoided. Not surprisingly, additional courses have been organised to cope with the number of participants.

This is the approach taken by the Microcomputer Unit. Each participant has access to his/her own single board computer (SBC) and a line to the Unit's multiuser computer. Software is developed on the computer and then downloaded to the SBC. On all courses run by the Unit, priority is always given to practical experience as this is the best way to learn. Consequently, numbers on both these workshops are limited to twelve, which enables trainees to have their own equipment and ensures sufficient personal tuition.

The workshops also cover the newer members of both familes — the 80286 from Intel (used in the new IBM AT) and the 68010 and 68020 from Motorola. Companies which have benefitted from these courses include British Aerospace, British Telecome, Thorn EMI and Ferranti.

In addition to workshops dealing with specific processors, it is well worth noting that the Unit runs a two-day hands-on introductory course on 16-bit processors. For those who have not yet decided which processor would best suit their purpose, this is an excellent opportunity to gain first-hand knoweldge and an unbiased view of what is available as the Unit is independent of any manufacturers. For further details on any workshop or courses run by the Microcomputer Unit, telephone 01-405 3020.



EARNLEY SHORT COURSES

The Earnley Concourse in Chichester, Sussex, is to offer a range of Computer Studies short courses as part of its Spring/Summer curriculum this year.

The courses, aimed mainly at the computing novice, are described in a brochure (obtainable from The Earnley Trust). However, also available is a two-day course entitled 'Microcomputer Applications', and another called 'Further Steps in BASIC'.

A complete list of leisure interest courses is available by writing to:

Owain Roberts,
The Earnley Concourse,
Earnley,
Chichester,
Sussex PO20 7JL.

TALKING SHOP

Don Thomasson

Past colleagues of mine are inclined to believe that I enjoy sticking my neck out, and I was certainly doing that when I took on the task of editing ***Computing Today.*** Now, with four issues published, my tightly-crossed fingers are beginning to relax. I think we may be winning.

What are the portents that suggest this? For one thing, our readers are beginning to write to us again, their letters a welcome sight amid the deluge of routine mail. Not all letters expressed approval, but there seems to have been a reasonable balance between criticism and praise, which is as it should be.

Bill Horne, in particular, has come in for some stick, but he always expects that. His role is to preach heresy, so encouraging people to defend their beliefs. (But he shouldn't have slipped my name in at the heading of one of his articles. I have my own troubles, without adding his.)

Bludners have not been avoided. Some have arisen because of a change in production arrangements which has taken time to settle down, but for others we must take more direct responsibility.

The Benchmark 8 business (PRINTOUT MAY 85) was spotted by a number of people, and has brought to light a major misunderstanding. It seems that some reviewers, impatient to get the tests finished, have run this last Benchmark for only 100 iterations, instead of the usual 1000, and have failed to multiply by ten. Some contend that this is standard in other magazines, but we - and Microchoice - have always declared a 1000-iteration standard. At least we have brought the discrepancy to light, and shown up its implications.

□□□

THE MOST gratifying portent is that a number of stalwart contributors of the past have renewed contact with us. Checking back, we found that about 240 authors and programmers have provided material for us since December 1980. Most have only made one appearance, or perhaps two, but some thirty-odd have given more regular support, really keeping the pot a'boiling. Some of these have looked a little askance at our new policy, but most seem to approve.

Among newcomers, we have had offerings from Germany, Spain, Egypt, India and Hungary, showing the wide distribbution of ***Computing Today*** in clear-cut terms. One item in preparation is an analysis of a program written by an Australian - The story of how we got in touch with him is quite a saga in itself!

One interesting point is that some submissions have been produced by full-size line-printers suggesting that the authors enjoy full professional facilities. And why not? Some of our stalwarts work in computing on a full-time basis and write for us in their spare moments. This bridges the nominal gap between mainframe work and microcomputers, to the benefit of both.

□□□

OUR TARGET area in terms of readership is continually under review. We began by cutting away three areas of the computer world which we judged to be outside our scope: the big-machine people and academic theorists who hang on to their coat-tails; the pure business scene; and the 'toy computers'and their programs. We would like to edge nearer to the business computer world, but there are problems. Just as it has been said that Shakespeare's plays embrace all possible plots, one set of business software may cover all the broad possibilities for that kind of work.The next set does much the same thing though perhaps in a slightly different way.

The other aspect of the problem is that business software is often quite simple in form. To find interesting programming examples, it is necessary to look at games!

□□□

I NOTICE that no-one has taken me to task - as yet - for my comments regarding the 'nervous' state of the computer market. The fact that Acorn ran into problems a couple of days before that issue appeared was pure coincidence. Their difficulties appear to have stemmed from an over-estimate of the size of their market, a mistake that is only too easy to make. Now that Olivetti have stepped in, there may be more margin for error.

The pundits are saying that the computer 'boom' is over. They expect the 'toy computers' (in which they include everything costing less than £1500 or so!) to fade away, leaving the field to 'real' computers. There's a taste of wishful thinking about that view, but it mainly shows a certain ignorance.

Forget the size and shape of the containing box, and the constituents of a computer will be found to conform fairly closely to predictble standard. The difference between a £400 model and a £1500 model is likely to lie in the peripheral area. For the cheaper type, discs are an add-on, whereas they are incorporated in the more expensive version as part of the basic system. The price differential is not as wide as it seems, while the performance differential may be simply a question of available software.

This needs a measure of qualification, perhaps. The limited memory space of the BBC computer puts it at an immediate disadvantage, despite its virtues in other directions. The more expensive machines tend to use '16-bit' processors, which cuts both ways. It gives access to larger memory areas, but ties the user to the available systems, which are not yet too numerous.

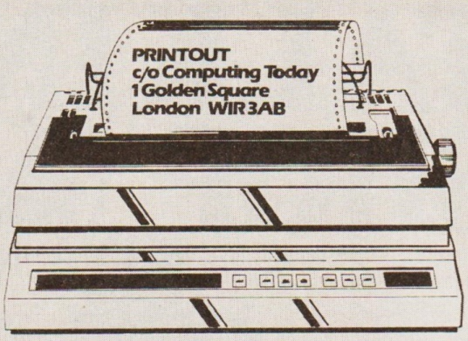
Oddly enogh, the only lower-cost micro to join the 16-bit revolution - the QL - uses the 68000 family of processors, for which there is less system material than there is for the older 8086 series, but that may prove to be a wise choice.

The QL is the first hint of a closure of the gap between 16-bit business machines and 'popular' computers. It may represent a valid trend, but there is an equal possiblity that the mainstream of action in that area will take a different course.

Meanwhile, it is noticeable that the flow of new products in the small computer field is drying up somewhat. Noises are being made, but not very positive ones, and that is about it. At one time, we were reviewing two or more new machines a month, but the jpresent rate is nearer one machine in three months. The sad aspect of the matter is that only a few of the devices which we reviewed at the peak of the rush are still with us. Some have fallen by the wayside in a definite way, others have simply sunk gently out of sight. In most cases, the reasons are not hard to discover, and can mainly be traced to inadequate initial capital, which led to lack of publicity, the marketing of machines before they were ready, and regrettably obscure manuals.

Without doubt, we are not the only people who have faced up to a new order of things in 1985. The microcomputer world will go on, whatever the pundits say, but it sems likely to move in new directions. Our job will be to identify the trends as quickly as possible and let you know what they are. □

PRINTOUT



Your opportunity to ask questions, put us straight, seek advice.

THE PRIORITY DECISION SYSTEM

Dear Sir,
Your article on the use of the Priority Decision System (PDS) to help resolve the coal dispute raises two important points. As with other versatile systems, PDS is used for many different purposes and applications, being a general decision-advice or problem-solving system. In fact, my staff who use PDS on problems of selection (of equipment, of personal, and of products) found it somewhat confusing to suddenly learn from *Computing Today* that it has been used son contentiously on industrial disputes. This suggests that more focused versions of PDS and other such systems are needed, each version tailored to a particular type of problem.

Your article also stressed the use of PDS as a decision-*making* system. May we point out that in our experience of PDS (as with many others in the PDS User Club), its best use is as a decision *insight* system. It helps managters sort out their thinking about problems. This is more important than coming up with decisions, even if the decisions are based on managers' views. This is why we have found its *main* use has been for training and development, rather than as a day-to-day instant decision aid.

Yours faithfully
Peter Ranken
Planning Director
ESCAPADE

The micro-computer industry has thrived on producing from the same basic system, sets of programs which look different and have various applications. Mr. Peter Ranken suggests the same development for the Priority Decision System (PDS).

But surely it's more cost effective for users to have access to one "versatile general problem-solving system", rather than several versions of the same program problem-by-problem. Further, an important benefit of PDS has precisely been to help managers using it analogise from one kind of problem to another. Mr. Ranken's firm Escapade have found this themselves, in using PDS for several different problems of selection — "of equipment, of personnel, and of products". In industrial disputes, PDS is analogously used to "select" the appropriate solution package — except that it's used by groups rather than by individuals.

Generally we have found that PDS users fall into two groups — those who mainly use it for daily decision-making, and those who (like Escapade) mainly use it for training managers in decision skills. We are at a loss to explain why 'PDS trainers' rarely progress to using the system for real operating decisions, when 'PDS decision-makers' often progress to using the system for training too.

J. Algie
Co-Director of Work Sciences Associates & Brunel Management Decision Programme

MORE COMPUTER INTELLIGENCE

Dear Sir,
After the battering Mr Weybridge gave me (CT March '85) I took a little time to recover, so this response is a bit belated.

I am accused of taking a negative view of Artificial Intelligence, but I would plead that my stance is realistic, rather than pessimistic. For example, take Mr Weybridge's point about colour. The experiments of Dr Land, the Polaroid man, have shown that our appreciation of colour is relative, whereas a spectroscope identifies colour in absolute terms. It may identify two colours that appear to us as being quite different. Such colours are called metametric pairs, and were much discussed in the early days of colour television.

This means that the intelligent computer might be able to classify the colours on a firm absolute basis, but might be unable to communicate with us on the subject in an intelligible manner. This is the difference between man and machine.

Then there was my reference to 'Syntax Error', which is a pre-coded phrase output in given logical circumstances. There is very little intelligence involved in that. If the machine offers you the erroneous line ready for correction, that is a little more intelligent, but the intelligence is that of the programmer. A human observer would probably lean over your shoulder and point out the error in a thoroughly irritating way. He doesn't have to do that. He just likes to annoy you. A programmed response, in Pavlovian terms? In that case, why are some people more tactful?

In the end, the key question still stands: What is intelligence? At least Mr Weybridge and I agree on that point...

Bill Horne

LADDER LOGIC

Dear Sir,
I hope you don't mind me mentioning it, but there seems to be a slight glitch in your Ladder Logic program (CT March '85). When using timers or counters, any output increments the counter or timer, due to entry into PROCcountstatus. this can be corrected by adding:

```
4135 IF yptr >7:ENDPROC
```

Yours faithfully,
John Hall

P.S. It's a great program!

ART AND THE AMSTRAD

Dear Sir,
I have been buying your magazine since December 1984 when you began printing material for the Amstrad CPC464.

I'm sorry that my first letter to you is in a negative vein, but in the December issue you published the first part of Art and the Amstrad. We were promised part two in January issue, but in that edition we were told that it had been postponed until the February issue.

I have now bouth the March issue and, lo-and-behold, no Art and Amstrad Part 2.

Could you please tell me when you intend to complete the program, or has it been cancelled completely?

W. Bell
Bradford.

Part 2 of Art and the Amstrad has been unavoidably delayed for reasons which providence forbids us to describe in any great detail. However, Kevin Smith - the author of the article - is well aware that part 2 is *required, and that we are very impatient for his copy. We apologise to all readers who were expecting this article. Please bear with us - the article will be published in either the June or July issue.*

HISOFT PASCAL

Dear Sir,

I should like to bring to your attention some factual inaccuracies in David Scott's review of HiSoft Pascal in the April edition of *Computing Today.*

Firstly, the product that was reviewed is out of date, not currently sold by our company and must have been acquired some time ago. HiSoft Pascal for CP/M-80 systems is now sold with a full screen, WordStar-like editor which is supplied with its own extensive manual and tutorial and is configurable to the user's system; both terminal codes and command sequences may be installed by the user and this process produces a help file for use with the friendly help screens available from the editor. Also supplied on the Pascal disc are two programs written in HiSoft Pascal: a prime number solution and a powerful Z80 disassembler which is a valuable tool in itself.

In addition to the above points of fact, I feel that one other point should have been made in the articl: HiSoft Pascal is a 100% British product, originated, manufactured and fully supported by HiSoft, here, in Dunstable. We are continually improving and extending our products and already have HiSoft CP/M Pascal available for the Amstrad CPC464, Tatung Einstein, MSX under MSXDOS, and other popular CP/M-80 micros as well as our cassette version of Pascal which runs on most of the Z80 home micros.

Apart from the above, one or two other errors crept into the review:

The Price: HiSoft Pascal plus editor plus examples = £39.95 fully inclusive.

Our Address: 180 High Street North, Dunstable, Beds LU6 1AT.

Our Telephone No: Dunstable 696421

I realise that is diffucult to ensure accuracy within reviews and I would therefore ask you to publish this letter in full in order to set the record straight on HiSoft Pascal.

May I take this opportunity to congratulate you on the new style format of Computing Today, a thoroughly good and informative read.

Yours faithfully
David Link

IN SEARCH OF SPACE

In the February issue of Computing Today you ask readers to give their views on choice of computers and here is my attempt to do this.

My main criticism of virtually all reviews is that they give very little for the person like myself who uses a computer for computing. I use my computer for Astronomical and lens design problems i.e. ephemerids of comets and asteroids from orbital elements; reduction of position from coordinate measurements of large scale photographs; ray tracing for new phtographic telescope designs. I write all my own programs and use a humble ZX81 which I find superior to many other computers for the following reasons (reasons which are never mentioned in reviews):

- Variables are retained when correcting or extending a program. This is extremely valuable when debugging a program, or for expample of calculations. One can simply GOTO a line and continue the work. BBC make one cry with frustration when debugging a machine like program with much data entry. A subroutine for data is the only way out, but there is no simple way for the other problem except perhaps disk storage.
- Variable names greater than 2 characters are recognised and there are no reserved words to worry about. Newer designs are beginning to get round these problems.
- The ability to display a full screen of program without guessing the last line number, and the built in syntax check.
- Despite Bill Horne's rather sarcastic comment in the March Computing Today the functions Arc, Sine, and Arc Cos are in my opinion a must for any self respecting computer. It should not be necessary to have to remember or look up such equations for these functions. The ZX81 has them.
- The simple command for "copy" is extremely useful for recording the result of calculation or plots.

Benchmarks are of interest to me particularly BM8, but I am puzzled by your figures in Computing Today. I get 207 secs. for Spectrum and find the Amstrad more than twice as fast as the Spectrum for a similar test. Precision is also important for my applications.

Physical dimensions are also important in my overcrowded workshop. I use a 5 inch television with the ZX81 and printer, which is light and takes up minimum room. I just could not find room for an Amstrad complete with Monitor, printer and disc drive, or afford it, even though it has some good features.

Finally, I have an I/O Port on the ZX81 and have written a machine code program for reading a homemade binary coded angular encoder and made my own 8K RAM, therefore I do have some interest in the workings of computers, but basically it is a very useful tool, and I am not interested in computing for its own sake. I realise many people are, and you have to cater for them, but there are quite a few amateur astronomers etc., like myself who I am sure would appreciate information on the aspects I have mentioned.

Yours faithfully,
B. Manning
Worcestershire

Here is the answer to those who say the computer is of little use to man or beast. Mr Manning's down-to-earth approach, using about the smallest computer available, is refreshing.

The virtues of the ZX81 which he mentions are, of course, shared by other computers, but this point is taken: Why go up-market if the bottom item pleases you? We know a radio station in Florida that uses a pair of ZX81s to control a bank of sound tape machines. They look a little odd perched in the middle of a row of six-foot racks, but they clearly do the job.

As to ARCSIN and ARCCOS, the point Bill Horne made was that lack of these functions need not be catastrophic. Some people would like to have hyperbolic functions, too, and a limit must be placed somewhere.

Finally, this Benchmark 8 business. We innocently took the figures from Microchoice, not realising that impatient reviewers had chosen to run 100 iterations instead of 1000, and had failed to multiply by ten to bring the result in line with the declared standard. Oh, dear!

LESSONS OF HISTORY

Bill Horne

During the early, post-war years, alternatives to the thermionic valve were being sought. However, general agreement had been reached upon a standard architecture for the CPU, and this is where our story continues...

ENIAC, the ACE, and the EDSAC all used the same essential elements. There was a central processing system, which fetched instructions from store and interpreted them, executing specified functions and defining the relevant data by setting up its address in store. There were provisions for input of data and instructions, and for output of results. This, in essence, is still a valid description of the functions required for a basic computer.

The central processor was essentially a logic system which could be set up in various configurations to perform particular operations. For example, if the contents of store address A were to be added to the contents of store address B, and the result put in store address C, the actions might be:

Copy the contents of location A to the accumulator.
Add the contents of location B to the accumulator.
Copy the contents of the accumulator to location C.

The accumulator was a special storage location or register which worked in association with an adding system to 'accumulate totals'. It was not regarded as essential in some concepts, though something of the sort had to be provided.

The process of adding two binary digits together, taking into account a possible carry from the addition of the next less significant digits, is a relatively simple logic function. Three inputs and two outputs are involved. If either one or three of the inputs is in a 'true' state, the sum output is also true. If either two or three of the inputs are 'true', then the carry output to the next digit pair is true.

The required logic function can be implemented in a number of different ways, but one will serve for the purpose of illustration. The two digits to be added will be denoted by A and B, and the input carry by CI. The Sum output will be denoted by Z and the Carry output by CO.

The carry output is calculated first, using the Boolean expression:

$$CO = (A^*B) + (A^*CI) + (B^*CI)$$

The * sign stands for an AND function, the + sign for an OR function. The expression as a whole means that CO is true if A and B are both true, or A and CI are both true, or B and CI are both true. In other words, if any two of the inputs, or all three of them, are in a true state, then CO will be true.

The Sum output is then calculated, using the expression:

$$Z = (A^*\overline{CO}) + (B^*\overline{CO}) + (C^*\overline{CO}) + (A^*B^*CI)$$

The overscore above CO indicates that the negation of CO is required. Z is true if A is true and CO is not, or if B is true and CO is not, or if CI is true and CO is not, or if A, B and CI are all true.

This needs a little consideration. If CO is not true, then no more than one input can be true, since CO is true if two or more inputs are true. Hence if A or B or C is true and CO is not, only one input is true, and Z should be true. The fourth term makes Z true if all three inputs are true.

Each term in the above expressions must be implemented by a 'gate', and each expression needs an overall gate. In the early days, that meant that at least nine valves were required, as many as would be found in a fairly complex radio receiver. The power required to drive the system might be as much as thirty Watts.

Since all that was needed to add two decimal digits together, it is understandable that the early systems worked on a basis of adding two digits at a time, carrying out this process thirty-two times to add two binary numbers together. In a modern system, the additions would proceed almost simultaneously, using 288 gate elements, but that would have entailed the dissipation of 870 Watts of power, which would have been, at least, inconvenient.

COMPLICATIONS

In practice, the matter went further, for subtraction was also required, together with logic AND and OR functions. For subtraction, an ingenious method was used. It was seen that inversion of every digit in a binary number was equivalent to subtracting the number from $2^N - 1$, where there were N digits in the number. This followed from the fact that the number $2^N - 1$ is represented in binary by a 1 in each digit position. For example taking N as 6, and adding the binary representation of 25 to its complement:

25	011001	
Complement	100110	
	111111	$2^N - 1 = 63$.

If the complemented form is added to another number, X, the result is $X + 2^N - 1 - 25 = X + 2^N - 26$. But 2^N is too large to be represented in an N digit number, and is represented by a carry from the most significant digit, so the result appears as $X - 26$. Addition of unit will give $X - 25$. The calculation of $30 - 25$ in six-bit binary thus becomes:

$\overline{25}$	100110	
30	011110	
Sum	000100	000101 = 5
	Add 1	

This kind of trick mathematics was essential in the production of practical calculating systems, and is now taken for granted.

All the internal workings of a computer system being in binary form, special problems arise in conversion from decimal input data and to decimal output data. Systems have been built which evade this problem by using decimal (binary coded) representation inside the computer. The two single streams of data fed to the adder were expanded to two sets of four streams each,

Dr A. D. Booth

through which the adder was presented with pairs of deimal numbers coded in four-bit binary. The adder became more complex, but there was an overall saving through the simplification of input and output.

One such system was derided by established designers, because it failed to accord with inbred ideas stemming from earlier work, but the method is available in most present-day microprocessors, though it is not used a great deal.

This illustrates a feature of the early years of computer design. Techniques had developed rapidly, and there was no time to discuss them in absolute detail. They were accepted, and became fixed parts of the computer firmament. Questions left unasked were never asked again, though in some cases a review might have produced useful illumination.

STORAGE

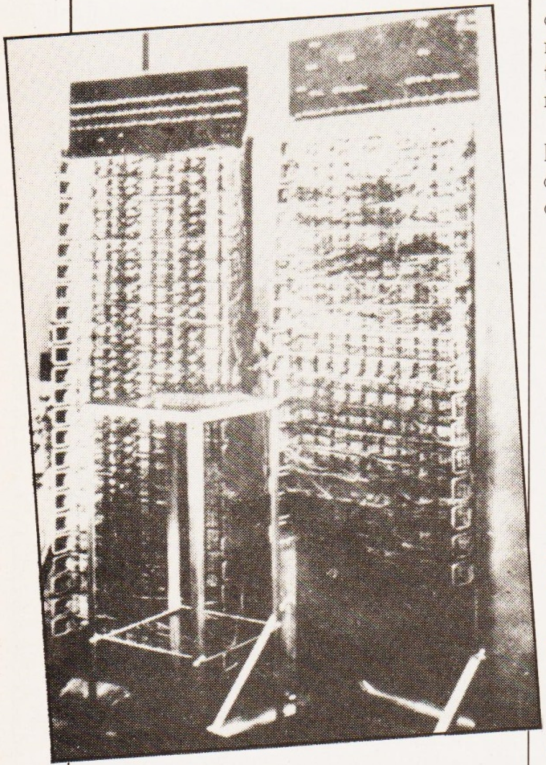
Apart from performing the necessary calculations, the central processor had to locate and collect the required data, and pass results back to store. With the early forms of store, this was not entirely straightforward.

It has been noted that Cambridge and N.P.L. favoured the mercury delay line store, while Manchester preferred the cathode ray tube. Both systems relied on comparatively short-term storage media, with arrangements to refresh the stored data at frequent intervals.

The mercury delay line used a tube filled with mercury. At one end there was a driving transducer, which sent a train of pulses into the mercury column. These travelled down the tube to a receiving transducer, which converted the pulses back into electrical form. If the system timing was properly synchronised to the time taken for a pulse to pass down the tube, the output of the receiving transducer could be used to generate a fresh train of pulses, any losses in the system being made up for by electronic amplification.

The cathode ray tube system was broadly similar in principle, the mercury being replaced by charges on the face of the tube.

The Royal Society Computing Machine Laboratory at the University of Manchester in 1949 (courtesy *The Radio and Electronic Engineer*)



Relay arithmetic and control unit of A.R.C., now in the Science Museum, London. (courtesy ***The Radio and Electronics Engineer*****)**

It had the advantage that the necessary parts were readily available, whereas the mercury store entailed very accurate machining and other production problems. Both shared the disadvantage of lengthy access time, in other words the need to wait until the required data became available, and both required the presence of power, without which the data they held was lost.

A rather different storage system was devised by Dr A.D. Booth in 1947. He was then attached to Birkbeck College, after spending six months with the Von Neuman team at Princeton, but a shortage of laboratory space made it necessary to use facilities at the British Rubber Producers Research Association. After trying an oxide-coated paper disc without success, he experimented with a nickel-plated drum, and this worked early in 1948. The drum was two inches in diameter, and gave a mean access time of 10 milliseconds.
tainly better than the more pessimistic forecasts.

It was also discovered that failures were much higher with brand new valves, and a graph of failure rate against time fell steeply to a satisfactorily low level, continued at that level for perhaps a year of continuous use, and then rose sharply again. The shape of the curve led to the expression 'bathtub' effect. On the basis of these findings, new valves were subjected to a 'burn-in' period, which weeded out the cases of 'infant mortality'.

Against these encouraging factors, there was the view expressed by Professor D. Hartree, who was said to know as much about computers as any man alive. He expressed the view that the Manchester, Cambridge and N.P.L. machines would suffice for all the calculations that the country would ever need, and there was no point in considering the manufacture of further models on a commercial basis.

Nevertheless, several companies were beginning to show interest in the production of computing equipment. Ferranti had already been involved with manufacture

Apart from its use in the A.R.C. computer, which Dr Booth designed, the magnetic drum was taken up by the Manchester team. Dr Booth also introduced the shift register as a secondary storage medium. The repertoire of useful techniques was growing.

There still remained the problem of dependence on valves. The first transistors had been made, but they were expensive and even more unreliable than valves. Someone close to the work expressed the view that they would never be mass-produced, and would never replace valves.

So, for the time being, valves were essential. It was found that British valves were more reliable than those of American origin, and that caused a sigh of relief among those who had predicted an excessive failure rate, since they had used American figures as a basis for their calculations. Practical results were cer-
for Manchester University, and English Electric had been involved in the work at N.P.L. There was a close link between Cambridge and J. Lyons, who wanted a machine to do clerical work. Elliott Brothers were showing interest, mainly in relation to their work in the military fire-control field.

GROWING INTEREST

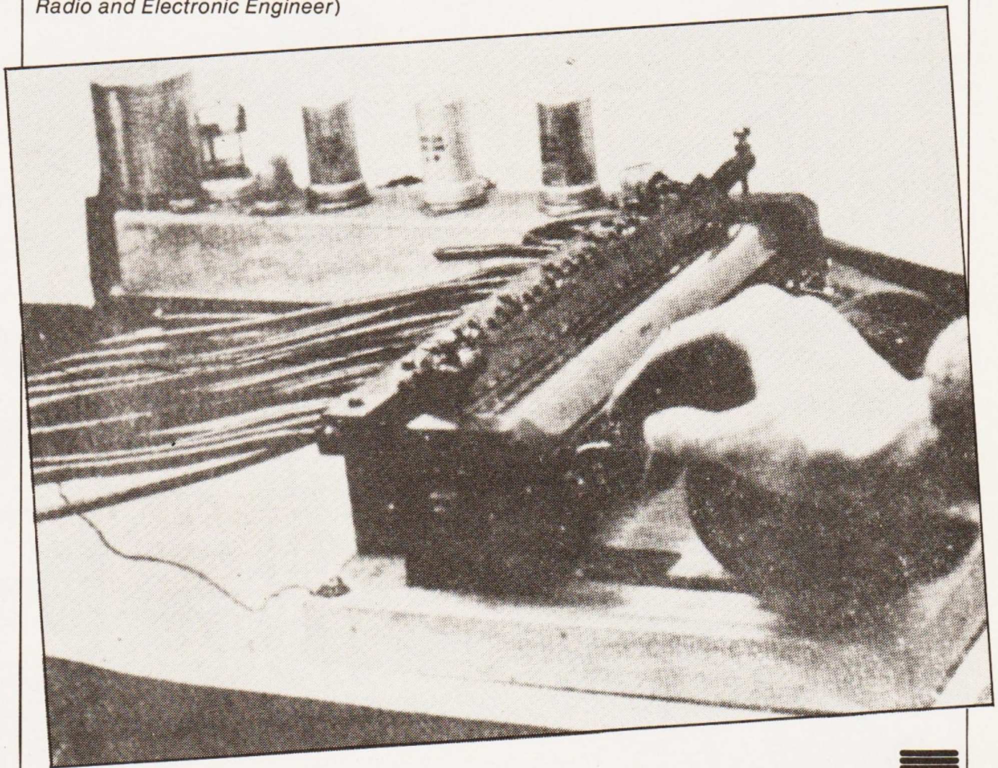
Quite apart from the firms who were dealing with actual computers, there was a growing interest among manufacturers of allied equipment. The relatively primitive input/output systems of the early days were being replaced by adaptations of standard punched card equipment, and both Powers Samas and the British Tabulating Machine Company became interested, among others.

Even so, the total effort involved was relatively small. A survey taken many years later concluded that there might be as many as two hundred genuine computer engineers in Britain. In 1950 that figure might perhaps have been divided by ten. There were many on the periphery of computer design, dealing with power supplies and routine matters, but the real know-how was shared by a tiny circle of experienced people who had been involved from the start.

One problem was that fear of the unknown which was mentioned in the introduction. Many potentially useful engineers were reluctant to consider involvement in computers. Not only did they doubt their ability to adapt to the relevant technologies, they also wondered whether the computer industry would ever become a viable proposition. At that time the market was extremely limited. Some specialist firms showed keen interest, but the majority were not even aware that computers existed. This must suggest a lack of adequate public relations, but it should be remembered that computer manufacturers faced formidable problems during the nineteen-fifties, and were perhaps not too keen to see the market expand before they were ready.

It had been proved that computers were practicable. It was still not certain whether they would be economically viable.

Booth's first working drum store, now in the Science Museum, London. (courtesy ***The Radio and Electronic Engineer*****)**



HISOFT

High Quality Microcomputer Software

CP/M PASCAL

When choosing a compiler, support is crucially important. It is vital to ensure that you will receive technical back-up for the package as quickly as possible. This may prove difficult, if not impossible, when the software was designed in another country and you have no access to the authors. **HiSoft Pascal** is a British product, created and manufactured by ourselves and fully supported by our technical team here in Dunstable. We offer inexpensive upgrades and we are continually extending and improving the compiler.

HiSoft Pascal is available in a wide variety of Z80 CP/M disc formats, is very close to Standard Pascal and comes supplied with a comprehensive and sophisticated full screen editor (ED80) which is specially designed for program development; the editor may be easily configured to suit all displays and also for special keyboards and individual keystroke sequences.

All this for a fully inclusive price of **£39.95!**

High Quality Software at a Fair Price.

HiSoft was founded over four years ago and since then has built up a reputation for the quality of its products and the strength of its support. Our utility and language software is available for most of the home computers currently on the market and we are always developing new products: our latest is **Devpac 80**, a powerful assembly language development package for all Z80 CP/M computers. **Devpac 80** is a macro assembler (GEN80), a configurable full screen editor (ED80) and a super front panel debugger (MON80) all in one integrated package.

Devpac 80 makes Z80 assembly language development a joy to do and is incredible value at only **£39.95** inclusive!

Please feel free to write to us or telephone for immediate despatch of full technical details of all our products – there is so much more than we can say in this small space. Our next product for CP/M-80? Wait and **C**!

HISOFT

180 High St. North
Dunstable, Beds
LU6 1AT
Tel. (0582) 696421

EPROM SERVICES

QL SPECTRUM and ZX81 HARDWARE

QL SERIAL TO PARALLEL PRINTER ADAPTOR – £38.95

★ Supplied with all cables ★ Integral mains power supply — reduced strains on the QL's PSU and high power printer output ★ Power-on LED indicator ★ True RS232 levels — for reliability ★ Fully enclosed black box ★ Crystal controlled BAUD rate generator — default 9600 baud, adjustable down to 300 baud ★ Adjustable parity and bits per character input — for use with other computers ★ Double buffered input — no lost characters ★ SUPERBASIC and QDOS compatible

Also available for ZXSPECTRUM Interface One and other computers with a SERIAL PRINTER PORT.

QL EPROM CARTRIDGE — £14.95
Fits the EPROM Cartridge socket holds two 2K, 4K or 8K EPROMS. Allows SUPERBASIC extensions to be available at power on.

QL EPROM CARTRIDGE with Software — £24.95
A collection of useful utilities supplied on a plug in EPROM CARTRIDGE.
Procedures include:
CLOCK — on screen clock
TIMER — on screen elapsed time
XDIR — extended directory (with pause)
REPEAT — Set key repeate rate and duration

Other EPROM Software also available e.g. Graphics, Basic extensions, multi tasking, machine code monitor.

QL CABLES

QL SER with 1.5 metres of cable	**£3.95**
QL CTL with 1.5 metres of cable	**£3.95**
MICRODRIVE approx. 8"	**£7.95**
QL SERIAL CABLE	**£11.95**

ESBUS SYSTEM
For ZX81 and SPECTRUM enables interfacing to a wide range and combination of interface cards, e.g.

Motherboard	**£16.95**
24 line I/O Card	**£14.95**
Real Time Clock	**£24.95**
8 Input 8 Bit ADC	**£23.95**
4 Input 10 Bit ADC	**£28.95**
8 Input 12 Bit ADC	**£57.95**
Dual Centronics I/F	**£26.95**
Single RS232 I/F	**£25.95**
Dual RS232 I/F	**£34.95**
3 Output DAC	**£26.95**
EPROM Programmer	**£54.95**

ZX81 EPROMS

Eprom Board C/W Toolkit Eprom	**£19.95**
Graphics Board C/W 4K of Eprom	**£24.95**
Autostart Card	**£9.95**

Wide range of software on EPROM e.g. Toolkits rapid load/save, M/C Monitor disassembler, etc. etc.

SPECTRUM

ROM Board **£14.95**
Replaces the 16K ROM with two 8K EPROMS plus 4 user definable switches.
Keyboard Interface **£11.95**
Add a second keyboard **£19.95**
I/O Board plugs into the Spectrum, providing two 8 bit I/O ports plus Strobe line.

EPROM CARTRIDGE SYSTEM
An EPROM system which replaces Sinclairs microdrive storage system with 48K interchangeable EPROM cartridges.

Interface	**£34.95**
Cartridge 6 x 8K Eproms	**£7.95**
Programmer	**£19.95**

Auto loading of programs on power-on. The Z80 vectors may also be redefined by reprogramming the interface's 2K ROM.
★ No mods to the Spectrum needed.
★ Works on 16K and 48K versions.

EPROM SERVICES

3 Wedgewood Drive, Leeds LS8 1EF Tel: 0532 667183

Prices include p&p. Please send SAE for full details. Export and Trade enquiries welcome.

£17.95 INC VAT, POST & PACKING FOR A LIMITED PERIOD!

PACE!

WANT TO LOSE WEIGHT? ...PACE IS THE ANSWER!

PRACTICAL! RECIPES FOR YOUR LIFESTYLE

PERSONAL! PACE USES YOUR COMPUTER TO DESIGN A DIET SUITED EXACTLY TO YOU!

COMPLETE! 2 CASSETTES PLUS A 200-RECIPE BOOK WITH UP-TO-THE-MINUTE HEALTH & SLIMMING FACTS

Please send me ______ copies of The PACE program
at £17.95 each to run on Acorn BBC B ☐
Commodore 64 ☐
I enclose a cheque/PO for £ ______
payable to Practical Programs Ltd.
Name
Address
Telephone
Or order by telephoning 01-794 4033.
Money back guarantee. Allow 28 days for delivery.
Mail to: Practical Programs Ltd., 7 Downside Crescent, London NW3 2AN. CT1

FAST PACE WEIGHT LOSS

TWO Cs

David Scott

To complement our 'Learn C' series, we look at two versions of the language, from Software Toolworks and Quality Computer Systems.

C/80 is one of several cheaper C compilers that are available under CP/M. C/80 costs £45.00 + VAT while an optional Mathpak costs £30.00 + VAT. Both C/80 and the Mathpak can be purchased together for £70.00 + VAT and all versions can be upgraded for a small charge. For example, the upgrade from C/80 version 3.0 to 3.1 cost me £13.00 + VAT from Grey Matter.

When C/80 is purchased you receive a disk suitable for your computer and a 49-page loose-leaf manual (Grey matter can supply most formats). (C/80 is supplied with the program shown in table 1.

So, although the compiler is quite cheap, you do get a wide selection of programs and utilities supplied as standard. C/80 is a standard implementation of B Kernighan and D Ritch 'C' as implemented in **The C Programming Language** with the following omissions:

Float, double, entry and typedef keywords.
Long and float constants and arithmetic (implemented in Mathpak).
Declarations are only allowed at the start of a block.
arguments to fdefine.
Bit fields.
£line.

As can be seen from the list, no editor is included in the package and you are expected to use your own CP/M compatible editor such as Wordstar or Ed. I personally do not like this and feel that all companies supplying compilers should include some form of editor, preferably a full screen editor like Wordstar. However, C/80 is very cheap when compared to some of it's rivals, such as Whitesmiths C compiler priced at £470.00 + VAT, and for this reason alone I can understand them not including a CP/M editor.

C/80 DOCUMENTATION

C/80 comes with a 49 page 'manual' (loose sheets of paper stapled together). However, although the manual is quite small it contains enough information to get you started, and is aimed mainly at the experienced C programmer. But, it does recommend **The C Programming Language** on page three as a better reference manual, and states quite often through the manual that the C/80 compiler is ***based*** on this but with the limitations listed above.

The fact that 'The C Programming Language' is recommended is not new as many C compilers are based on this, and many of the more expensive compilers tend to include the book when you buy the compiler.

COMPILATION

Software Toolworks' compiler is a single pass compiler which generates absolute 8080 assembly code. This can then be assembled with AS, which is supplied as standard, this assembler has the ability to assemble separately compiled modules are called with an XTEXT directive in your assembly source code. This ability means that you could create your own compiled routines to be inserted in any program at a later date.

Optionally, assembly code can be output in format suitable for Microsoft's Macro-80 or Digital Research's RMAC relocatable assemblers, it is then possible to develop .REL files which can be linked using Link or Link-80 link loaders. Also, on page 34 of the manual, it suggests that a relocatable assembler should be used for large projects although AS could still be used for most purposes.

Included with C/80 is a configuration program, CCONFIG.COM, which can be used to change the defaults on C/80, as well as any other .COM file generated by C.COM and AS.COM. This is very handy, as it saves having to write out a long command every time you want to use specific compiler options.

Also available is an optional Mathpak which adds Long , Float and double keywords and constants which helps to bring C/80 nearer to the Unix standard for C as defined by Kernigan and Ritchie.

Listed below is the range that long , float and double keywords have:

STANDARD LIBRARY

C/80 comes with a standard library of just over 45 functions. These include sequential, formatted and random access. Also included are string and character functions and extensive operation system support. Most of the library is listed in C source-code, so it should be possible for the good C programmer to modify or improve these routines to suit his own needs.

In conclusion, I would recommend C/80 to anyone that is interested in learning C or even developing commercial software. However, you do have to provide your own editor, relocatable assembler and linker, although the assembler supplied should be suitable for most purposes. I would suggest that if anyone is going to develop commercial software or transfer C source code from one compiler machine to another, then purchase the optional Mathpak which should reduce most of your problems.

Q/C

Q/C is another of the cheaper C compilers available, and can be ontained in two forms: one for 8080 and another for the Z80, the main difference being that the Z80 version takes advantage of the extra Z80 opcodes, which in turn gives you faster and more efficient code. The reviewed version was for the Z80. As well as the Q/C compiler, I was provided with the Code Works Assembler (CWA) which consists of an assembler, In addition to these programs, I was also provided with two manuals. These were photocopies onto A4 size paper, the reason for this was that the latest manuals were not available but would be sent when they became available. The manual supplied for Q/C consisted of 173 pages of fairly detailed information, with six main chapters linker, and library manager, and is similar to the Microsoft M80 and L80 software packages. The price for CWA is £25.00 + VAT.

Q/C does not come as standard with an assembler and it is therefore important that you have a Z80 or 8080 assembler when Q/C is purchased, otherwise you will not be able to create executable programs of any sort.

When you receive your disk with the Q/C compiler (which includes the complete source code of the compiler) you receive the programs shown in table 2 as standard
Also supplied on the same disk was the CWA package, which consisted of the programs shown in table 3
and six appendices.

Q/C DOCUMENTATION

The Q/C manual consists of the following six main chapters: Getting Started... Fast!; Using the Q/C compiler; Running your Q/C program; Advanced Q/C topics; Q/C function library and Compiler internals. Each chapter has smaller chapters devoted to more specific subjects. Also included are six appendices entitled 'How Q/C differs from standard C'; 'Q/C error messages'; 'Sample compiler output'; 'Compiling the compiler'; 'Maintaining the function library' and 'Q/C on CP/M compatible systems'. In general, I found that more than adequate information was given, well and in a logical order. It's only fault was that no index is given and it means you're constan-

C.COM	— The C/80 compiler.
AS.COM	— An absolute 8080 assembler.
PRINTF.C	— C/80 formatted output routines.
HELLO.C	— Sample program from 'The C programming Language'.
TAB.C	— Sample program to replace blanks with tabs.
CMP.C	— Sample program to compare two files.
TREE.C	— Sample program showing use of structures.
TPRINTF.C	— Smaller version of PRINTF.C.
PRINTF.H	— Header file for PRINTF.C.
SCANF.C	— C/80 formatted input routines.
SCANF.H	— Header file for SCANF.C.
STDLIB.C	— Standard library in C and assembly language.
SEEK.C	— Routines for random access input/output.
EXEC.C	— Routines to chain another .COM program.
COMMAND.C	— Routine to expand file wildcards in command line.
CLIBRARY.REL	— Relocatable version of CLIBRARY.C.
CLIBIO.C	— System dependent portions of CLIBRARY.C
CPROF.C	— Runtime execution profile library.
CTRACE.ASM	— Alternate runtime execution profile library.

Table 1. C/80 program directory

CC.COM	— Q/C C compiler.
QRESET.COM	— Configuration program for Q/C.
EXPAND.COM	— Program to expand compressed text files.
HELLO.C	— Sample program from 'The C programming Language'
COMPARE.C	— Sample program to compare to files.
CRUNTIME.MX	— Runtime library macro file in compressed format.
QSTDIO.H	— Q/C standard library and functions.
SETJMP.H	—
QRESET.CX	— Source code for QRESET.COM in compressed format.
CASMLIB.CX	— Assembly library in compressed format.
CDISKLIB.CX	— Disk library in compressed format.
CRUNLIB.REL	— Relocatable runtime library.
CSTDDEF.HX	—)
CGLBDEF.CX	—) Source code for Q/C compiler
CGLBDECL.CX	—) in compressed format.
GC1.CX to CC9.CX	—)

Table 2. Q/C program directory

CWA.COM	— Code Works Z80 assembler (M8O compatible).
CWLNK.COM	— Code Works linker (L80 compatible).
CWLIB.COM	— Object code manager.
CWPATCH.COM	— Configuration program for CWA,CWLNK and CWLIB.

Table 3. Codeworks assembler directory

FACTSHEET	C/80
Price	£45.00 + VAT
Available from	Grey Matter 4 Prigg Meadow Ashburton Devon TQ13 7DF Tel. (0364) 53499

FACTSHEET	Q/C
Price	£80 + VAT
Available from	Grey Matter 4 Prigg Meadow Ashburton Devon TQ13 7DF Tel. (0364) 53499

tly flicking through the manual looking for specific items. However, the Contents page is quite detailed, reducing the amount of searching required.

The manual supplied with CWA consisted of 17 pages explaining how the CWA package works. The CWA manual does not have chapters like the Q/C manual, but as separate sections consisting of an introduction and explanation of the programs supplied. The CWA manual does not have a contents or index page, so you will just have to get used to it. Also the CWA manual is not a tutorial on Z80 assembly language, so if you are considering a lot of Z80 programming I suggest you consider getting a tutorial or reference book on the Z80. Overall the Q/C and CWA should be more than adequate to get you using the programs supplied, but, although well laid out and presented, they are not tutorials.

Quality Computer Systems C compiler is a single pass compiler similar to C/80 which generates Z80 or 8080 object code, depending on the version purchased. Q/C creates either an .ASM or a .MAC file depending on whether you are using Digital Research's RMAC or Microsoft's Macro-80, the default is set for .MAC files but this can be changed easily to .ASM by entering the compiler option -A or changing the default with the configuration program QRESET. This option is not available on the Z80 version.

When using Q/C, I generally found that it compiles faster than C/80. Also, when creating C programs there are less £include statements required. This is due to C/80 having its library in more than one file. For example C/80 requires two files to be included plus the standard library in any program that involves printing to the screen and random access, where as Q/C only requires the standard library to be included, the main advantage of this being increased compatibility with the Kernigan & Ritchie standard.

The library which Q/C uses presently contains 80 functions, unlike C/80 which only contains 45. The functions included are mostly compatible with the Kernigan & Ritchie standard but with the following omissions:

- Variable types Long,Float and Double.
- Parameterized £define commands.
- Initialisation of auto or register variables.
- Local declarations in compound statements.
- Bit fields.

I would also like to let readers know that Quality Computer Systems is in the process of releasing version 3.3 of Q/C which should rectify some of the above omissions. As far as I know, version 3.3 is going to include Long integers and floats, as well as a larger library. Also to be released soon is an MS-DOS Q/C. The prices for version 3.3 should be the same as they are now.

In conclusion, and like C/80, I would recommend Q/C to anyone interested in learning C at low cost. However, for people wishing to get greater compatibility and access to more of the C language, then they should definitely give Q/C greater consideration over C/80. This will most definitely be the case when version 3.3 is released. However, Q/C does not provide an assembler or full screen editor as standard, so the extra cost must be considered when purchasing any of these compilers. Anyone that requires greater compatibility than C/80 or Q/C can offer, should consider some of the more expensive compilers available such as the Whitesmiths C compiler. But, Q/C or C/80 should suit home and most commercial users.

INSIDE LCDs

Are Liquid Crystal Display panels a realistic alternative to the Cathode Ray Tube?
Peter Gee **of Epson (UK) describes the fortes and the foibles of the flat screen.**

Industry observers will have noticed that the LCD appears to have won the battle between differing display technologies in the truly portable computer market. Epson, Hewlett-Packard, Data-General and Texas Instruments have all chosen the LCD as the display medium for their portable machines. The reason why is perhaps not so obvious.

Since 1968, when the RCA Corporation introduced the first liquid crystal display (LCD) device, the LCD has been considered a promising display medium. Progress in developing the technology has resulted in the use of LCD panels in several consumer products, but more work is needed before the same kind of stability can be achieved in colour LCD displays.

The rapid progress in electronics, and semiconductor technology in particular, has added to the scope of application of LCDs, beyond the use in calculators and games and into potential products like the long dreamed-of TV set that would hang on a wall. It is already being used in compact, flat-faced office computer terminals and composite, TV-based terminals, and is drawing considerable interest from many manufacturers for more development work.

This article discusses the state of LCD development and future possibilities. Liquid crystal, although an organic compound, is unique in that it is midway between being a liquid and a solid, thus its applications are not rigidly defined, and are expected to be more and more diversified. The possibilities range from optoelectronics to biotechnology. The description in this article will concentrate primarily on LCD display technology.

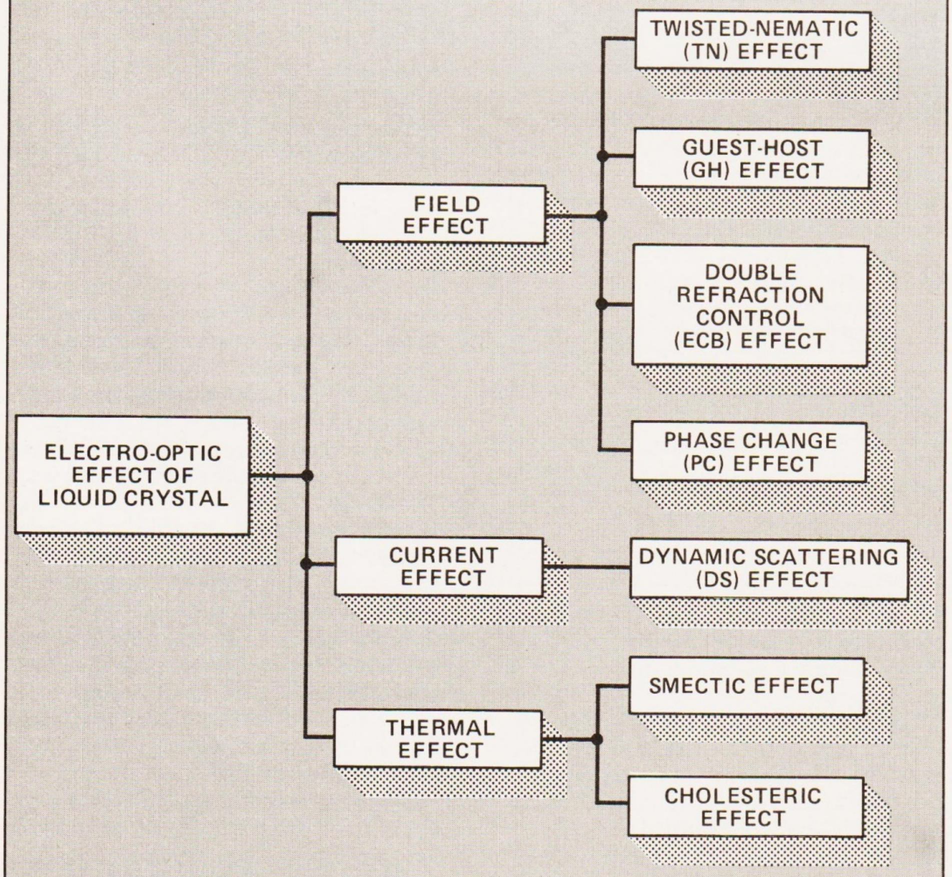


Figure 1: the LCD family tree

WHAT IS LIQUID CRYSTAL?

Liquid crystal is a unique material having properties of both a liquid and a crystal. Liquid crystal has a long history; it was discovered by F. Reinitzer, an Austrian botanist, in 1888. Since then about 2,000 kinds of liquid crystal materials have been discovered.

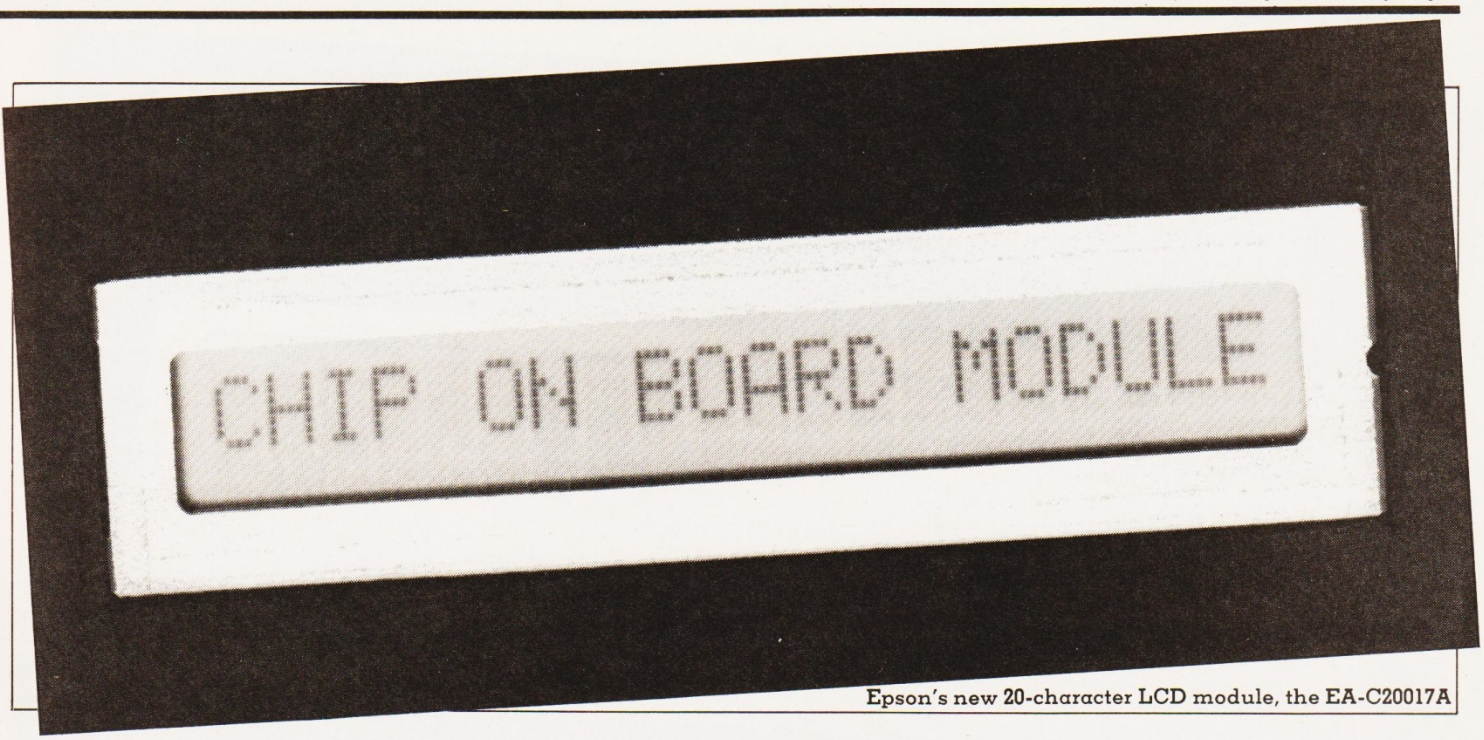
But it was in the late 1960s that the use of liquid crystal as various functional materials was started. Now, thanks to the remarkable advancement of its application technologies, liquid crystal is being employed in a great variety of products: digital watches, pocket calculators, office automation equipment such as personal computers and word processors, sensors and optoelectronic elements.

LIQUID CRYSTAL: MIDWAY BETWEEN LIQUID AND SOLID

The term liquid crystal represents no particular material, it is used to denote any material having both properties of liquid and solid. There are three material conditions — gaseous, liquid and solid — but some material has the unique property of not being bound to any of these three conditions.

Since liquid has an irregular molecular arrangement and also has fluidity, incident light upon liquid is reflected. Solid crystal, however, has the regular molecular arrangement and, depending upon the direction of the arrangement of incident light, is either transmitted or reflected. Which is called anisotropy.

Liquid crystal possesses both properties



Epson's new 20-character LCD module, the EA-C20017A

of liquid and solid within a certain temperature range. It is liquid in appearance with fluidity, but has molecular arrangement and optical property similar to those of solid crystals. For this reason, liquid crystal is sometimes called the "fourth material condition" falling between liquid and solid.

APPLICATION TECHNOLOGY

Molecules that constitute liquid crystals each have the shape of either slender rods or flat plates and they are regularly arranged so that the longitudinal axis of each molecule is aligned in parallel with each other.

Liquid crystal has the property of fluidity, thus its molecular arrangement is not rigid as is that of solid crystal. Liquid crystal molecular arrangement can be varied simply by subjecting it to external excitation such as electric or magnetic field, temperature and stress. Most contemporary liquid crystal application technologies depend upon this unique molecular structure along with its optical property; in other words, it is the "weak" molecular structure that opens up the present wide application of liquid crystals.

In LCDs, liquid crystal is sandwiched between two glass plates. When voltage is applied to the liquid crystal, the affected molecular portion will vary its arrangement, creating dark areas. This is because the varied portion does not pass light, but refracts it.

With LCDs, information can be displayed by transmitted or reflecting light from the outside; while electric lamps and Light-Emitting Diodes (LEDs) display by self-emitting. The actual LCD structure is not as simple as it is described. When voltage is applied to the liquid crystal molecules, their arrangement, and hence their optical property, will vary. This phenomenon is called the Electro-Optic Effect, which is the basis for liquid crystal application technology. This effect can be classified as shown in figure 1.

Of these three classifications shown, the field effect is the most advanced at present in terms of application technology. Especially widely used as the display material is the Twisted-Nematic (TN) liquid crystal. Depending upon the molecular arrangement, the liquid crystal can be grouped into three types as shown, that is, Nematic, Smectic and Cholesteric.

Other than these, there are a number of new liquid crystals having other properties, recently discovered or synthesised, with great possibilities for wider application.

LIQUID CRYSTAL ADVANTAGES

Liquid crystal is regarded as the leading technology for displays found in various products. LCDs were first introduced in watches and pocket calculators, then, in the 1970s, they were used in instrumentation, home electronics and audio equipment,

THE EPSON LCD RANGE

Large screen LCDs: As well as the EG7001A-AR 100 charater by 25 line device, Epson is also able to supply large screen LCDs built to customer's specifications. Typical requests include specifying a different glass or pixel size or, in some cases, a change of aspect ratio to enable graphics software written for a CRT batch machine to be converted to run on an LCD with no alteration. A 640 x 400 dot 100 character x 50 line LCD is in development: ideal for word processing applications.

Backlighting: To improve the contrast of LCDs in poor lighting conditions, an EL backlight is available as an option. By use of an inverter, this backlight can run from a 5V supply and typically draws 220mA — a lot for a battery powered portable machine but not for a desktop machine.

Touch key overlay: A transparent touch key overlay can be fitted over Epson LCDs enabling the panel to act as both an information display and an input device. This opens the way for easy-to-use software packages where the user just touches the command he requires.

Up to 2000 contacts can be placed on a 200 x 640 dot LCD. The touch-panel works on an X-Y matrix method using a flexible transparent membrane which carries the contact wires.

Plastic film panel: As used in the Casio credit-card calculator*. Just 0.8mm thick, same principle as a normal LCD but uses plastic instead of glass, enabling LCDs of any shape to be made — f curves, circles etc. Maximum panel size is better than 300 x 200mm. Cost is 2.5 times price of normal LCD.

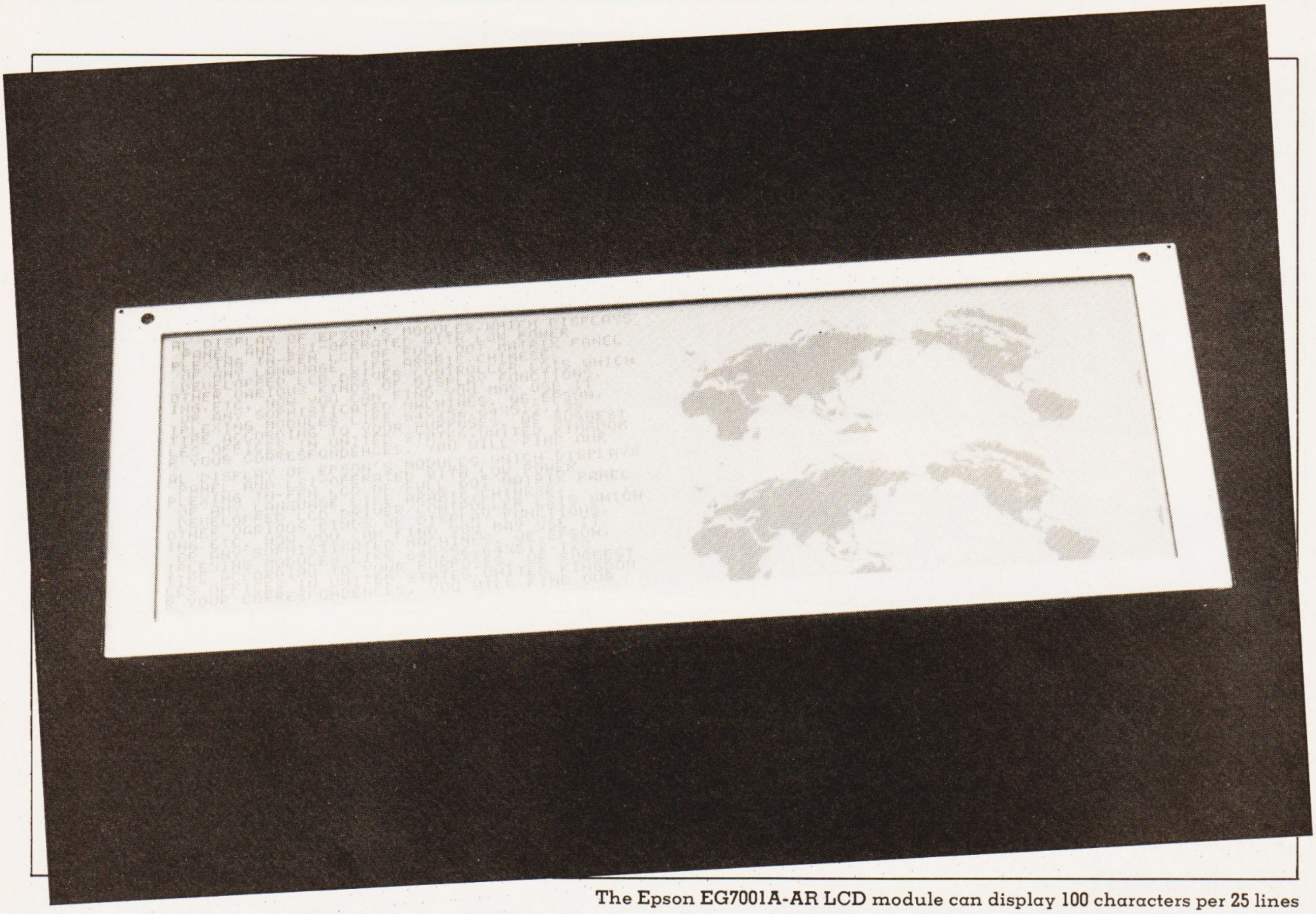
Black shutter — Epson first: To be released by Epson in March. Consists of a normally black opaque LCD panel which, when activated, turns transparent to light. By placing a LIGHT SOURCE BEHIND, THE LCD gives a very high** contrast display suitable for use in consumer goods (video, clocks) and information boards at railway stations and airports.

The limit at which blackshutter can be multiplexed currently limits this technology to small simple displays.

EA-20017A: A 20-character dot matrix LCD featuring a chip-on-board driver and memory, to give a compact design and very low cost.

*Epson is the world's largest supplier of LCD panels for calculator and watch applications. Current production is 7 million panels per month.

**Could be used as a camera or projector shutter and in anti-dazzle car mirrors.



The Epson EG7001A-AR LCD module can display 100 characters per 25 lines

office and learning equipment and hand-held games equipment. Large-capacity character displays, graphic displays and colour displays are bringing LCDs into the fields where cathode ray tubes are difficult to use.

LCDs have become so popular in such a short period because of their advantages over other display elements. Among them are these:

- Lower drive voltage (could be driven with solar cells)
- Lower power consumption (longer battery operation possible)
- Flat panel display possible
- Better electrical compatibility with display-driving ICs and simpler driver Circuit, thus permitting easier interface with a computer
- Easier on the eyes, with no emissions (no "VDT problem" encountered)

For instance, features such as lower drive voltage, lower power consumption and lower cost make LCDs suitable for watches and calculators, while better compatibility with ICs, flat panel construction and less eye fatigue are features for hand-held computers or personal word processors.

FROM SIMPLE PATTERNS TO COMPLEX PICTURES

The liquid crystal application technology has advanced greatly in the last decade, from simple pattern display to complex picture display, from low-contrast display to distinct display and from monochrome to colour. The difference is obvious when comparing an LCD digital watch of 10 years ago with the recently marketed LCD colour TV receiver.

The early LCD is a segment display, in which the liquid crystal is sandwiched between glass plates having rod electrodes which control the activation of the liquid crystal. The presently used LCD is a dot matrix device with numerous band electrodes arranged perpendicular to each other to produce a matrix of dots for display.

The LCD capability can be increased dramatically to such an extent that more complex and minute patterns can be displayed. Even with this dot matrix method, however, increasing the number of scanning and display electrodes will tend to reduce the liquid crystal contrast, thereby degrading the quality of the display. This contrast problem can be overcome with the Active Matrix display, which is gathering more attention.

This method employs a matrix of switching elements corresponding to picture elements with each picture element being switched directly for display. The receive innovative LSI manufacturing technique makes this extremely distinct and minute picture display possible.

TELEVIAN — FULL-COLOUR LCD TV

The development of LCDs has been concentrated upon large displays, expansion of the quantity of displayed information, easier reading and colour display. Most of these requirements have been fulfilled because of recent, rapid technological progress. Notably, the LCD with full-colour capability is a new breakthrough, which is comparable with the progress from the vacuum tube to solid-state technology.

The colour television picture is a detailed moving picture in subtle halftones. To display this picture with an LCD, the LCD must have high resolution, high-speed response, multiple tone wedge display capability, high contrast and capability of producing any colour on its panel.

Although theoretically possible, the creation of a full-colour LCD TV set has been considered difficult. In August 1984, however, Televian, an LCD pocket colour TV set, was introduced, jointly developed by Epson and Suwa Seikosha. Its two-inch picture screen has 52,800 picture elements and can produce a colour picture with high contrast and resolution.

The liquid crystal panel of "Televian" consists of a glass plate with a matrix of polysilicon Thin-Film Transistors (TFTs), a transmissive TN liquid crystal and a glass plate with tiny, regularly arranged sets of three primary colour (red, blue and green) filters.

Here, the liquid crystal elements function as optical shutters, controlling the light passing through the colour filters. As an example, the picture screen looks blue with the red and green shutters closed and the blue shutter open. Since about 53,000 liquid crystal shutters with 53,000 corresponding three-dot primary colour filters arranged on a tiny 2-inch space, very sophisticated technology is required.

Epson has no plans to market the colour LCD TV in the UK

MEDIA FOR THE NEXT DECADE

As the advent of television creates the image culture of the present, new technology makes possible media improvements which in turn can create new culture. This seems to be the case in the history of technological advancement.

By the end of this century, completely new information media, originating from liquid crystal technology, could become available. This may appear as a portable composite terminal with the combined capabilities of TV, telephone, copier, facsimile and computer. This would be a go-anywhere type with integrated LCD and keyboard powered by solar cells or small batteries, networked with other terminals or computer systems via radio or optical communication function.

With this communication with the host computer, various data bases can be easily made for immediately obtaining necessary information. By varying the combination of available capabilities, it can be used for business or in the home. It can be carried to school for learning purposes or playing games, or it can be used for housekeeping or used as a controller for electric utilities and gas appliances, all with the functions of a personal TV.

This information equipment could be produced with the progress of liquid crystal technology. Even now, reduction in size is being achieved with computers and word processors which use LCDs for their displays. The LCD advantages, of low power consumption and flat display, can be quite attractive in size reduction and multi-function capability.

The Epson PX-8 incorporates an 80 character by 8 line LCD screen

DEC BENCHMARKS

Rich Billig and Randy Cronk, Digital Equipment Corporation

This article surveys the more common microprocessor benchmarking techniques as applied in industry. Benchmarking is defined as a measure of total system performance as opposed to simple hardware performance. The various areas of system performance are described. A method for measuring performance in these areas is proposed with a brief example of its application.

Before evaluating various microcomputers, it is important to establish relevant performance criteria. As engineers we always try to select the best tool or component for our particular design. When that component is a microcomputer, the relevant question to ask is: Which microcomputer runs my application the "best". The word best here usually means the fastest, in the smallest package, for the lowest cost.

The only way to answer this question with complete certainty is to try the application with each microcomputer under consideration. Because of time and cost, however, it is rarely possible to build a real-life application around several different microcomputers just to test which one actually performs best in a final design situation.

One reason is that real-life applications tend to involve special relationships between the microcomputer and particular hardware devices. These relationships are difficult to duplicate in a benchmark situation without going through the trouble of actually building the special hardware environment to be used.

A second reason is that real-life applications are usually too large to program cost effectively to use solely to compare microcomputers.

A common alternative to expensive real-life benchmarks is to run no benchmarks at all. Instead of benchmarks, the application engineer relies on product specifications, design descriptions, and perhaps past experience to guide him in his microcomputer selection. These indicators, however, do not accurately predict how a microcomputer will perform in a new and complex application.

A second alternative frequently used is to run what are called hardware benchmarks. These benchmarks are small segments of assembly code designed to test the pure execution time of the basic processor. The problem with hardware benchmarks is that they do not measure microcomputer performance from the high-level application point of view. To increase programmer productivity, most applications are written in vendor-supplied high-level software, not assembly code.

A third alternative is to run applications that consist of high-level code. These programs draw upon all resources of the microcomputer, not just the hardware. The challenge with this approach is to test total system performance without getting bogged down in special hardware or extensive programming.

TOTAL SYSTEM PERFORMANCE — A DEFINITION

Total microcomputer performance depends on performance in three areas:

1. Instruction set and system architecture
2. Hardware implementation of the architecture
3. Software implementation of the architecture.

All computers implement a general scheme of organization, functions, and behaviour known as an architecture. The architecture is the way the machine appears to the machine level programmer and can be characterized by the instruction set processor (ISP). Better architecture makes it possible to perform more functions with fewer instructions. Hence, even though a processor may execute its instructions more slowly, it may execute an application fast because of better architecture.

Three important factors that contribute to architectural performance are bit efficiency, orthogonality, and addressing capability. A bit-efficient architecture allows the computer to execute an algorithm with fewer instruction bits. Bit efficiency is a function of the number of bits in the instruction word and the number of operations performed for each instruction. A computer with a large instruction word may be more bit efficient that a computer with a small instruction word if the computer can do an equal number of operations with far fewer instructions.

The benefits of bit efficiency are small program size and high execution speed. With fewer bits needed for operations, programs can be smaller. Therefore they are more likely to fit into high-speed on-board memory. Also fewer memory references are required to fetch program instructions.

Orthogonality measures the ability of a computer to address different data types the same way, independent of the data type it references. A possible problem with non-orthogonal architecture is that some data types are harder for the computer to handle than others. Operations that use a more difficult data type require more memory and execute more slowly.

The programmer's choice of data type should be a function of the application rather than the microcomputer. If an application calls for a problem data type, the programmer is faced with a hard choice. He can force the computer to work with the data type, inefficient though it may be. Or he can use a data type that matches the preferenes of the computer at the expense of the application. Neither choice is optimal.

An architecture with a good addressing

Figure 1 Three Major Factors Controlling Microcomputer Performance

AREA	WHAT IS IT?	HOW TO MEASURE IT?
SYSTEM ARCHITECTURE (INSTRUCTION SET PROCESSOR)	THE ORIGINAL 'GLOBAL DESIGN' FOR A COMPUTER	• BIT EFFICIENCY • ORTHOGONALITY • COMPLETENESS
HARDWARE IMPLEMENTATION	THE ACTUAL REALISATION OF THE DESIGN IN HARDWARE	• SPEED OF INSTRUCTION EXECUTION PER UNIT COST • POWER • PHYSICAL SIZE
SOFTWARE IMPLEMENTATION	THE ACTUAL SOFTWARE THAT IMPLEMENTS APPLICATION ON THE ARCHITECTURE	• SIZE AND SPEED OF RUNTIME EXECUTIVE • SIZE AND SPEED OF LANGUAGE SUPPORT CODE • SIZE AND SPEED OF COMPILED PROGRAMS

capability uses the same instruction to address a processor register, main memory, or an I/O device. Moreover, no distinction is made between data and address locations anywhere within the system, including the processor registers.

This can be helpful when manipulating arrays, for example. In a system with dedicated data and address registers, an array subscript must often be created in data registers before it can be copied to address registers to access the operand. This transfer from data to address requires additional program code that can reduce system performance.

Hardware performance determines how fast the machine executes particular features of the architecture, such as a MOV instruction. This is a function of the basic technology, circuit layout, component selection, design ingenuity, and other factors. Good hardware design also reduces package size, cost, and electrical power requirements.

Software is the code executed by the hardware. It consists of compiled source code, the runtime executive, and high-level language support code. A high-level application typically consists of all three types of software.

An application's performance depends greatly on the compiler that translates the source lines written by the programmer into executable machine object code. A better compiler produces fewer bytes of object code per line of source code. This implies that the hardware needs to execute fewer instructions per application so that applications execute faster from smaller

Figure 2 Estimated Application ROM Memory Requirements as a Function of Program Length

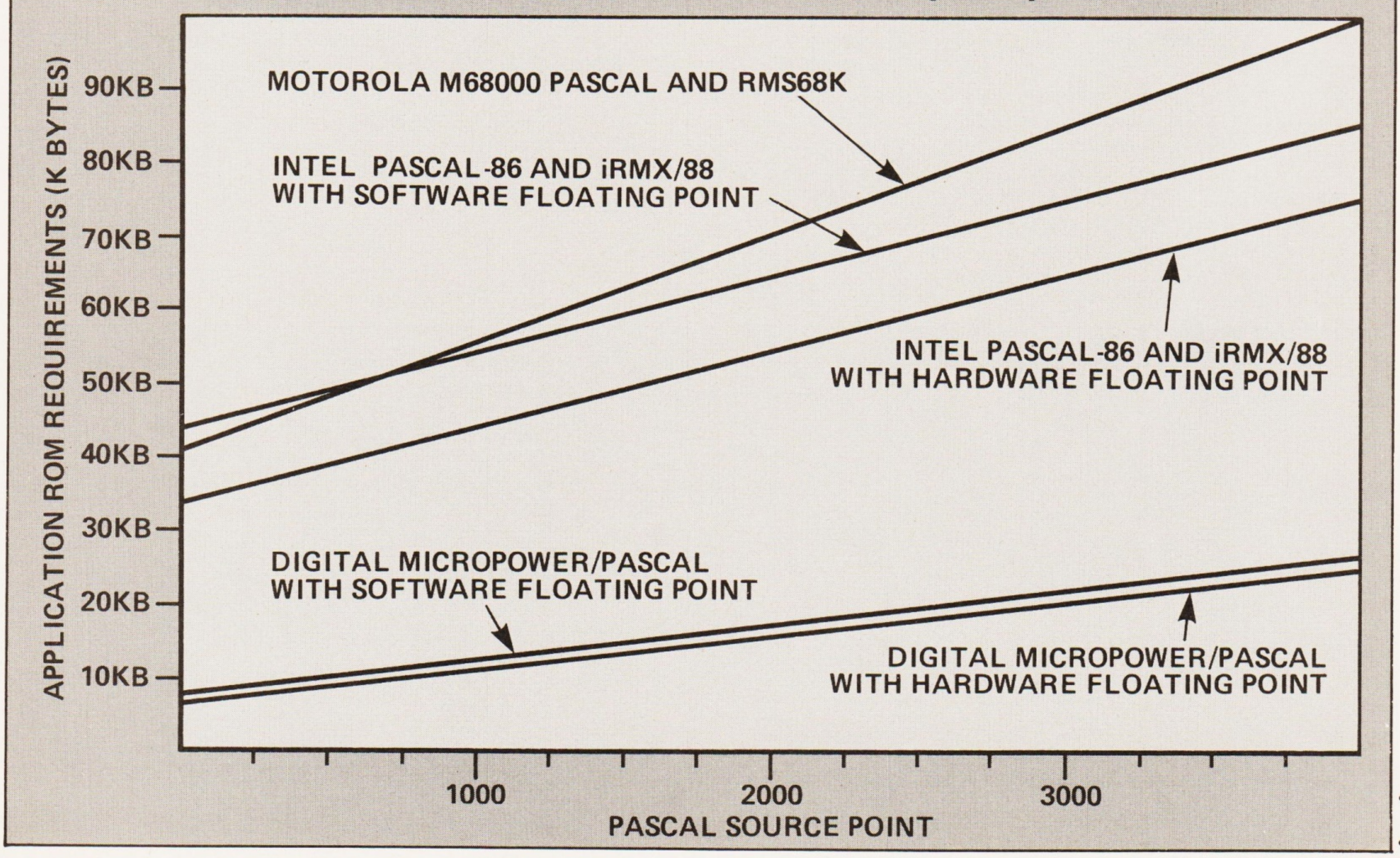


TABLE 1
Pascal Benchmarks — Execution Speed in Seconds
Benchmarks

	1	2	3	4	5	6
DIGITAL FALCON SBC-11/21 with on-board memory	*	19.9	*	58.2	61.8	72.3
DIGITAL FALCON SBC-11/21 QBUS memory	37.8	33.4	84.9	94.2	104.3	118.9
DIGITAL LSI-11/2 with KEV11	31.0	31.0	34.1	43.5	34.7	22.4
DIGITAL LSI-11/23 wqithout floating point hardware	16.7	15.0	17.9	41.4	48.1	53.8
DIGITAL LSI-11/23 with KEF11	16.7	15.0	17.9	9.0	19.5	14.8
DIGITAL LSI-11/23 with FPF11	16.7	15.0	17.9	3.3	10.7	5.9
INTEL iSBC-86/12A without floating point hardware (5 MHz)	53.5	15.0	335.5	!	124.4	!
INTEL iSBC-86/12A with iSBC-337 (5 MHz)	53.5	15.0	335.5	3.9	17.6	8.6
MOTOROLA M68KMPU	!!	34.2	15.0	190.0	5700.0	305.0

*Required more RAM than available on-board; timing could be estimated from QBUS memory time (1.6 to 1.7 times faster than on-board memory)
! Time not taken with E8087.LIB (software floating point support)
!! Program would not execute; NEW () and DISPOSE () functions not implemented.

memory.

Sharing the target system's memory with the compiled source code are the microcomputer vendor's executive and language support services. The executive provides the routines needed to schedule and synchronize processes, drive external devices, and manage other software resources. The language support services are functions especially implemented for the high-level language. Some examples might input/output data formatting, special math functions, and dynamic allocation o storage space.

Runtime executive and language support software should occupy a minimum of computer memory. This allows more space for the compiled source code. Finally, fast executive and support software allow faster and more reliable realtime applications.

A BENCHMARK METHODOLOGY

The purpose of benchmarking is to measure performance in all areas that could affect the user's real'life application. Usually more than one benchmark program needs to be run to measure the various application components. It bears repeating, however, that no benchmark or set of benchmarks can take the place of the real-life application as a completely accurate test of how that application will run on a microcomputer.

The following would therefore be an appropriate list of microcomputer benchmarking guidelines.

- Benchmarks should represent a scaled-down version of the algorithms to be used in the actual applications.
- Benchmarks should be written in the high-level language intended for the application.
- Benchmarks should be compiled under the existing vendor-supplied compiler.
- Benchmarks should be run in the actual runtime environment to be used (i.e., with the vendor-supplied support software).
- Benchmarks should measure memory usage, in addition to execution speed.
- Benchmarks should always be measured, not estimated, in the intended execution configuration.

Such benchmarks, run on different microcomputers, would provide a reasonable basis to compare systems for high-level real-life applications.

MICROCOMPUTER BENCHMARKS — AN EXAMPLE

When DIGITAL recently ran microcomputer benchmarks against Motorola and Intel it selected Pascal as a high-level language.

The subset of Pascal defined by Jensen and Wirth was selected because it is the Pascal standard most universally accepted.

Six benchmark programs were run. Each required compilation as well as various levels of system software support. Each program also was complex enough to enjoy the

TABLE 2
Compiler-Generated Program Size

	DIGITAL MicroPower/Pascal V1.06		INTEL PASCAL-86 (Series III) V1.1		MOTOROLA M68000 Pascal V1.20	
Benchmark	**Size**	**Ratio**	**Size**	**Ratio**	**Size**	**Ratio**
1	600	1.00	997	1.66	994	1.66
2	372	1.00	563	1.51	648	1.74
3	764	1.00	1036	1.34	1320	1.73
4	148	1.00	260	1.76	272	1.83
5	338	1.00	571	1.69	846	2.50
6	1470	1.00	2012	1.37	3872	2.63
Average Size Ratio:		1.00		1.60		2.00

Note: Sizes listed are in decimal bytes, and take into account all instructions generated, as well as any constant data (strings).

advantages (or suffer the disadvantages) of each vendor's architecture. Of the six, three used integer-only computation and three used floating point computation. The goal in selecting the benchmarks was to find programs that demonstrated various performance aspects in order to arrive at a general comparison of the total systems offered by the three vendors.

The six benchmarks were as follows:

- LIST—This function uses the "NEW ()" function of Pascal to dynamically allocate 1000 records of the type that might be used by a compiler generating a symbol table. These records are then formed into link lists that are searched and manipulated. This program is specifically aimed at testing Pascal for system implementations.

- QUEENS—This classical Pascal benchmark calculates the number of possible ways to place eight queens on an 8X8 chessboard so that no queen is attacked by any other queen.

- SALE97—This program tests some of the more difficult to implement Pascal features such as set manipulation.

- SMALL1—This program tests real number statement execution speed. The program calculates a short expression 30,000 times.

- MATRIX—This program tests operations on three 4X4 real number matrices. The main loop is repeated 1000 times.

- WHET2—This is one of the Whetstone benchmarks translated from FORTRAN into Pascal. It performs a variety of complex floating point operations repeatedly.

Again, the first three benchmarks are integer-only while the second three are floating point.

As shown in Table 1, the DIGITAL machines tested were a FALCON, LSI-11/2, and LSI-11/23. The FALCON was tested with the application running completely from on-board memory (its normal configuration as a single board computer) and also from a RAM card located on the QBUS. The LSI-11/2 was tested with the KEV11 hardware floating point option. The LSI-11/23 was tested without floating point hardware support and with both the KEF11 and FPF11 floating point hardware options.

The Intel microcomputer tested was the iSBC-86/12A. It was tested both with and without the iSBC-337 hardware floating point option.

The Motorola machine was the M68000-based M68KMPU.

As the data in Table 1 shows, DIGITAL demonstrated total system performance generally equal to or better than Intel or Motorola. (It is interesting to note that the application run times show no relationship whatever to processor clock frequencies.) Table 2 shows the amount of object code generated by each of the three compilers involved. The DIGITAL compiler consistently produced code that was 30 to 60 percent the size of the others.

PERFORMANCE = SPEED + MEMORY EFFICIENCY

This data illustrates the importance of memory efficiency as a microcomputer performance parameter. Most benchmark data does not show the amount of memory required by the benchmark on various mahines. This may be a serious oversight.

The software's ability to fit into a small area may determine whether or not a given application can even be run on a microcomputer in the first place. Another consideration is cost. At several hundred dollars per board, the cost of added memory modules can substantially impact the cost of an application. Physical size and weight are also factors. An extra memory card or two may severely restrict where an application can go.

Suppose, for example, that a particular application absolutely must fit into 64K bytes of ROM. The question then becomes: How many lines of high-level code can be executed from 64K bytes? The greater the number of lines, the more functions that can be put on a single board computer or single memory card. The answer can be estimated in two steps.

1. Subtract from 64K bytes the amount of memory required for vendor-supplied system software. This yields the

TABLE 3
Sample Calculation of Number of Pascal Source Lines That Fit in 64 KB ROM

	DIGITAL MicroPower/Pascal		Intel Pascal-86+IRMX/88		Motorola M68000 Pascal+RMS68K
Floating point Hardware?	Yes	No	Yes	No	No
Initial ROM	64 KB	64 KB	64 KB	64 KB	64 KB
Minus size of Minimum OS support*	-5 KB	-5 KB	-21 KB	-21 KB	-17 KLB
Net ROM after OS	59 KB	59 KB	43 KB	43 KB	47 KB
Minus Size of Pascal Support Code	-2 KB	-3 KB	-13 KB	-23 KB	-24 KB
Net ROM for Pascal prog.	57 KB	56 KB	30 KB	20 KB	23 KB
Divided by byes/l;ine compiler efficiency	/7.1	/7.1	/10.4	/10.4	/15.3
Estimated number of source lines in 64 KB ROM	8.0 K	7.9 K	2.9 K	1.9 K	1.5 K

* OS size includes nucleus and terminal handler plus I/O system. Sources: ***Intel Systems Data Catalog*** #210299-001 (January, 1982); ***M68000 Real-Time Multitasking Software User's Guide*** #M68KRMS68K(D1) (December, 1980).

amount of memory available after loading the software required to handle system runtime requirements such as process scheduling and driving external devices.

2, Divide into this available space the number of bytes typically generated for a line of Pascal by the vendor's Pascal compiler.

Data for both these steps is available from the benchmarks. To get data for step one, average the amount of runtime support required in each benchmark. To get data for step two, divide the number of Pascal lines into the number of bytes of instruction and constant data generated for each compiled program. Average this result across all the benchmarks. Table 3 shows the results of these steps for the six benchmarks.

As Table 3 shows, system software from different computer makers typically leave an application with different amounts of memory. Even though DIGITAL, Intel, and Motorola may start off with 64K bytes of ROM, DIGITAL leaves 57K bytes for the compiled application, while Intel and Motorola may only leave 30K bytes and 23K bytes, respectively.

The next question is: Which computer uses this space most efficiently? As Table 3 shows, a line fo Pascal code typically compiles into 7.1 bytes of machine code on a DIGITAL system. This compares with 10.4 bytes for Intel and 15.3 bytes for Motorola. The available application space is next divided by the number of bytes generated, to arrive at the number of lines of Pascal that can be compiled and loaded into 64K bytes of ROM with room left over for support software.

The results indicate that 8.0K Pascal lines can be loaded into the DIGITAL system as opposed to Intel's 2.9K lines and Motorola's 1.5K lines. Figure 2 compares how the ROM requirements for different vendors change as applications increase in size.

KEY PERFORMANCE QUESTIONS

The purpose of this article has been to illustrate some of the many factors that should be considered when measuring microcomputer performance with benchmarks. When correctly used and interpreted, benchmarks are a valuable tool with which to compare microcomputers. Some specific questions that are appropriate to ask when evaluating benchmark data are suggested below.

Do the benchmarks measure the ***total*** computer product — hardware, architecture, and system software?

Besides speed, what other factors are important to this application — such as cost, size, expandability, support, and so on?

How much memory do the runtime executive and language support software leave for your application?

How many lines of high-level code will fit into on-board memory?

ERATOSTHENES REVISITED

An article in ***Byte Magazine*** (September, 1981) provides an example of how to evaluate microcomputers based on execution of high-level instructions. The article written by Jim Gilbreath describes the Eratosthenes Sieves Prime Number Benchmark. This benchmark computes all prime numbers from 3 to 16,000. The benchmark avoids division and uses prior knowledge about numbers that cannot be prime (such as even numbers and multiples of primes). Multiplication is not used. Gilbreath ran the benchmark on several microcomputers using various compilers. The article provides data on both execution speeds and compiled sizes of the target application. No data on the sizes of the target operating systems or language support software, however, is provided.

In response to Gilbreath's article, Intel ran the identical benchmark on its System 86/330 using FORTRAN, C, PASCAL, and MicroSoft BASIC. (System 86/330 was not included in the original article.) Intel listed the results in its publication, ***Benchmark Series: Eratosthenes Sieves Prime Number Benchmark on System 86/330,*** Number 5, April 1982, (order number: 210441-001). As Intel notes, "This benchmark demonstrates compiler efficiency in terms of object code size. CPU efficiency is also demonstrated as the benchmark is CPU-bound."

Although no LSI-11 microcomputers were included in either evaluation, Intel states that the benchmark also demonstrates that, "The System 86/330 is clearly superior to the LSI-11 for this benchmark." On the contrary, when DIGITAL ran the benchmark we found just the opposite to be true. Both the LSI-11/2 and the LSI-11/23 executed the benchmark faster and required less memory than the System 83/330. Both LSI-11s ran the same Pascal source published by Intel and Gilbreath. The source was compiled by the MicroPower/Pascal compiler.

The following table combines the Pascal performance data from Gilbreath, Intel, and DIGITAL.

Article reproduced courtesy of Rapid Recall Ltd.

	Language and Machine	Compiled Size (Bytes)	Times Larger Than LSI-11/23	Execute (Seconds)	Times Slower Than LSI-11/23
1.	NBS Pascal, PDP-11/70	333	2.49	2.6	0.60
2.	MicroPower/Pascal, LSI-11/23	134	1.00	4.33	1.00
3.	NBS Pascal, PDP-11/60	333	2.49	4.5	1.04
4.	MicroPower/Pascal, LSI-11/2	134	1.00	8.83	2.04
5.	MT Mirosystems Pascal MT, 4 MHz 68000	410	3.06	9.0	2.08
6.	Intel Pascal, 5 MHz System 86/330	240	1.79	9.20	2.12
7.	RSI Pascal, 4 MHz 68000	318	2.37	10.2	2.36
8.	Motorola Pascal, 68000	387	2.89	14.0	3.32
9.	Microsystems Pascal MT+,Z80	308	2.30	19.0	4.39
10.	Pascal, HP3000	—	—	20.0	4.62
11.	UCSD Pascal, Pascal 100	298	2.22	54.0	12.5
12.	UCSD Pascal, Pascal Microengine	298	2.22	63.0	14.5
13.	Ithaca Intersystems Pascal/Z,Z80	761	5.68	109	25.2
14.	Atari Pascal, Atari 800	—	—	190	43.9
15.	UCSD Pascal, Z80	282	2.10	239	55.2
16.	UCSD Pascal, TRS-80 Model II	282	2.10	274	63.2
18.	Pascal/M, Z80	301	2.25	450	104
19.	JRT Pascal, Z80	232	1.73	470	109
20.	UCSD Pascal, Apple II (6502)	287	2.14	516	119

THE INDUSTRY SPEAKS

We introduce here an innovation which we hope will become a regular feature: a slot which is made available to representatives of the computer industry so that they can talk about matters which they feel need airing.
The first speaker is Michael Thom, Chairman of System Designers Scientific, who raises a matter of considerable importance to the well-being of British software houses, but which does not appear to have been publicised elsewhere.

I am grateful for this timely opportunity to use *Computing Today* as a platform for airing an issue. Right at this moment (early March) a vitally important matter to the software industry is being resolved by the Department of Trade and Industry. This is going to have a tremendous impact on many of us but — with a few notable exceptions — I do not see much sign that the industry has realised what is about to happen. I refer to the new regulations on the control of software exports which are likely to be brought into force very shortly as a result of the decision of the COCOM meeting in Paris last autumn.

The COCOM export control committee is a forum of the NATO nations and Japan which rules on the limits to be imposed on the export of Western technology to the Eastern-bloc countries. To ensure the rules are obeyed member governments operate a very positive control on all relevant exports to *all* countries East *and* West. For many years COCOM has embargoed the export of a considerable range of computer hardware. In support of this there have been two recent court cases in which Company Directors have been convicted of attempting to evade the embargo. At the Paris meeting late last year the decision was taken to tighten up the embargo and, for the first time, to extend it to include computer software.

STRATEGIC EMBARGO

Let me at this stage declare that, as Chairman of a company which primarily services the UK Government and Aerospace Industry, I fully support the principle of a strategic embargo on the export to the Warsaw Pact of NATO high technology products. I am only too well aware of the grave loss of hard won technological advantage over the last 40 years — from the early jet engines through to mainframe computers. However my particular concern in the present situation is that there is a grave danger that the software embargo, whilst banning our non existent sales to the East, will inhibit our considerable business with our allies.

Let me explain.

The already published United States' regulations on the scope of the software embargo give a very good indication of what is likely to happen here. Firstly all software for proscribed hardware is embargoed (with a few commercial exceptions). Then software in the following categories is embargoed:

- High level language development systems e.g. Ada APSE
- Cross-hosted compilers
- Operating Systems providing on-line transaction processing and on-line data base up-dating.
- Application Software in a wide range of defence categories and all Artificial Intelligence and KBS.

From the above it should be obvious, that for software companies specialising in real time complex systems just about every activity from applications development through to specialised product sales will be constrined by the embargo. In the case of Systems Designers' virtually the whole of our product range would appear to be covered eg.:

- HORIZON, PERSPECTIVE and PERSPECTIVE Ada — Development Systems
- CONTEXT and MAGIC — Host-Target cross compilers
- POPLOG and ENVISAGE — Artificial Intelligence products

For any *free world* export sale of such products or for related applications software it will be necessary to obtain a validated licence from the Government licensing office. This is likely to be a time consuming process. Furthermore there will be obvious difficulties in obtaining export licenses to cover the arrangements to supply through distributors — and for volume export sales the local distributor is a key element in the sales strategy.

In our own case we have well-establish distributors in three NATO countries and in Sweden. The nature of our products is such that the typical customers in those countries are the local defence and aerospace companies. Thus it seems probable that a sale of one of our off the shelf products by our Italian subsidiary to Augusta Helicopters would first have to be approved by the UK licensing office. Alternatively if we wished to make a direct sale from the UK to Messerschmitt-Bolkow-Blohm of a product relevant to the NATO Tornado programme we would again require a licence. Further problems would then occur when we wished to issue an update as part of our routine product support activities.

FURTHER COMPLICATIONS

When the application software situation is examined the position becomes even more difficult. Off-shore systems projects are often best undertaken jointly with a local hardware company. The application software may be developed on their site by our team, or on our site in the UK and then shipped out for integration, or a bit of both. The potential for bureaucratic delay in such situations is horrifying.

I hope by now that I have made my point. I have researched the situation fairly carefully and I am not exaggerating the likely impact.

□□□

IF THIS MESSAGE has struck chords of alarm then it may not yet be too late to act. The Computing Services Association has been negotiating the details of the British regulations with DTI. I am sure the CSA Director General, Dr Douglas Eyeions, would appreciate further support. What we need is a set of regulations which, while enforcing an embargo on transfer of software technology to the Eastern bloc, does not hinder us doing normal business with our long standing customers in the Western World — particularly those in other NATO nations.

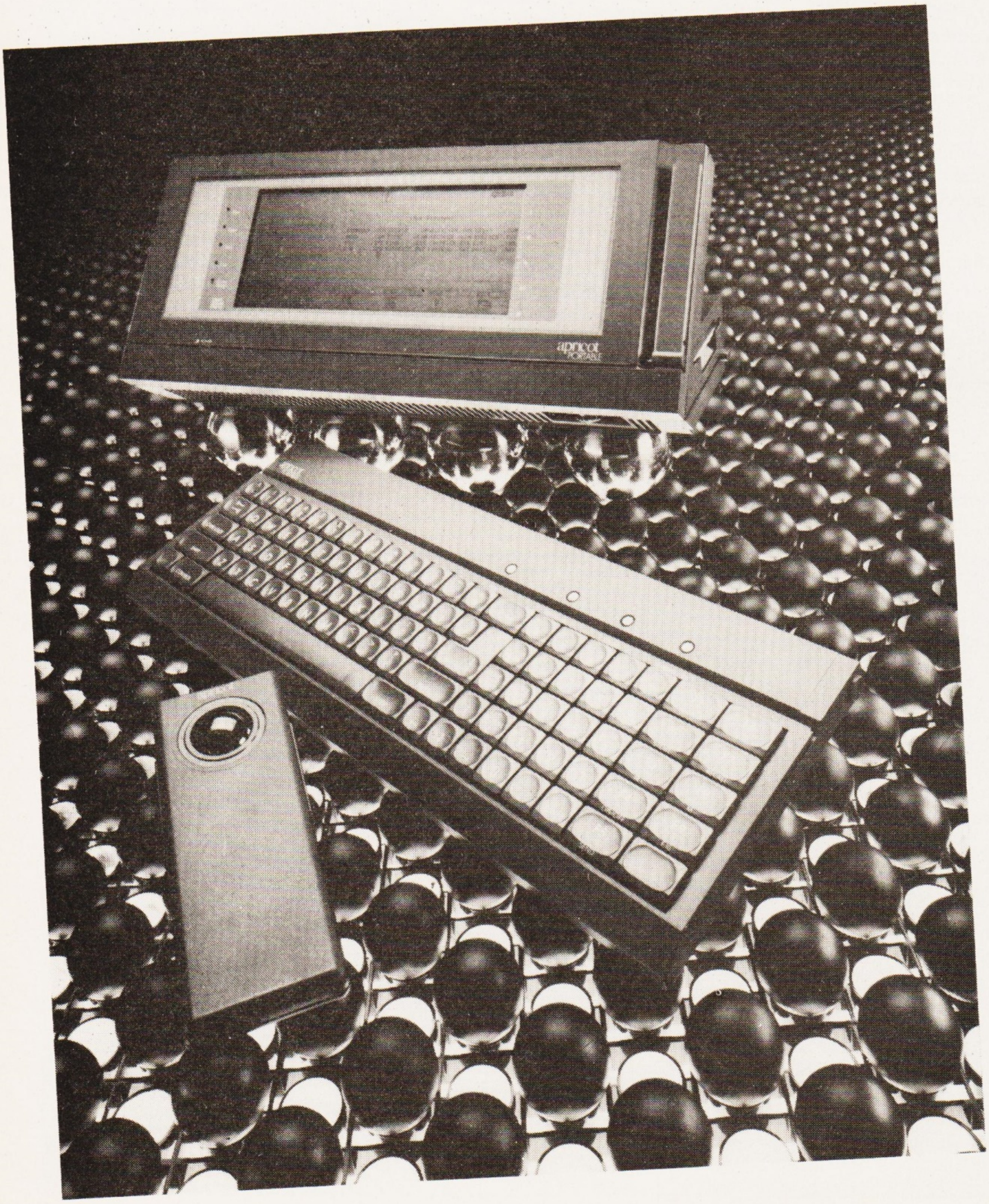
There is a great danger that the COCOM software embargo could result in the UK industry shooting itself in the foot. Don't let this happen.

Michael I. Thom
Chairman, Systems Designers Scientific
March 1985

We invite contributions from other members of the industry to raise issues of a similar nature.

THE APRICOT PORTABLE

Bill Horne **reviews the sleek, slim, portable micro from ACT.**



My weekly visit to the local library takes me past a computer showroom, and for some weeks past I had been intrigued by a machine which I saw on display. In the end, I asked if I could look it over, and what I found is reported here. It is by no means a full review, but it does bring out some interesting points.

What had particularly attracted my attention was the display, a compact rectangle perhaps an inch thick, but there were other surprises in store.

SETTING UP

With the mains cable plugged in, I looked round for the connection between the keyboard and the main unit. There wasn't one. For normal use, an infrared link serves, the keyboard being battery-driven. If several similar machines are being used close together, creating the possibility of cross-talk, an optical fibre link can be used, but that would rarely be necessary. The absence of cable straggling across the bench made everything look neat and tidy.

The disc drive was at the right-hand end of the main unit, and took 3½" double-sided discs giving 720Kb capacity. A slight craning of the neck to see where to put the disc was the only problem here.

At first, the display looked rather dim and gloomy, but once the initial load had been performed it was possible to adjust the contrast by keyboard action, and all became clear and bright. For text, there was room for 25 lines, each of

80 characters, and the shape of the screen allowed a much better aspect ratio for the individual character areas than is possible on a typical VDU screen. The bit resolution was 640 x 200, and this allowed the display of fine lines and detailed icons, crisp and clear. For those who need colour, there are separate display units giving 8 or 16 colours, but these need an optional colour RAM in the main unit.

The only reservation I had about the display was that there was sometimes a loss of image during rapid movement. A scrolling text display used in one demonstration was difficult to read, though it instantly became clear when the scrolling was halted.

Apart from this, the display was very satisfactory, especially in that it avoided the cramped feeling induced by some 80-column formats.

The keyboard did not prove to be as pleasing. Measuring some 18" x 7" in plan view, and a little more than an inch thick, it presented a solid phalanx of keys that was a little confusing. Each key had a concave top surface, which should have made touch-location easier, but in practice did not.

The left-hand two-thirds of the key area was occupied by a fairly standard QWERTY layout, but the nearest row of keys held the space bar, ESCAPE, CONTROL, STOP (halt action), the SHIFT keys being in the second row. Some keys bore cryptic graphic symbols, including an L-shaped Return key. This was suitably large, but closely surrounded by other keys.

To the right of this came ten miscellaneous keys, including four arrow keys, then a 15-key numeric pad, and finally ten function keys. Finally, there were four miniature buttons beyond the keyboard proper, for Reset, Repeat Rate, Set Time and Keyboard Lock.

The result may be a stylist's delight, but a typist might take a different view. No doubt familiarity would make the layout more acceptable, but that would not solve the problem for casual and occasional users. Personally, I would have preferred some separation of the keyboard sections.

TALK TO ME

Of course, the hope is that we will eventually be able to do without the keyboard altogether, relying entirely on voice input. The Portable takes a step in that direction, the microphone being on a level arm that stows neatly on the right hand end of the display, but can be hinged forward for use.

Since this facility takes a while to set up, I was unable to put it through its paces, but I have heard that it can be temperamental if you have a cold. Perhaps Tunes would remedy that...

Table 1

Tutorial	GW BASIC
Activity	Personal BASIC
User Interface	Dr Logo
ACT Diary	CBASIC-86
ACT Sketch	CBASIC-86 Compiler
Superwriter	C Compiler
Supercalc	Pascal/MT + 86
Superplanner	PL/1
MS-DOS 2.11	Level 11 COBOL
GSX	Assembler plus Tools
Concurrent DOS	Display Manager
CP/M 86	MBASIC Compiler
	FORTRAN
	etc

Software for the Apricot portable

THE TIMOROUS BEASTIE

For an extra £300, you can have a mouse as a pet. This one works in two modes. Place it flat on the bench, and you can mnaipulate the top of the ball with your fingers. Tilt it, and the ball rests on the bench, ready to register movements of the mouse as a whole.

Like the keyboard, the mouse is infra-red coupled to the main unit, which makes it very convenient to use. It was certainly more convenient as a control of cursor movement than the alternative keys — which are not the arrow keys, but the numbers of the keypad, omitting 5.

The mouse is not essential, though it is very useful in some kinds of work. Whether the extra cost is justified is for individual users to decide.

OTHER HARDWARE

The 8086 central processor is not a favourite of mine, though it enjoys the support of more software than some other 16-bit types. However, it does its job, allowing the use of 256K, 512K or 1M of memory, the last two being manufacturing options. Other memory sizes can be obtained by using 125K, 256K or 512K expansion boards. One option here is the provision of a 'RAM Disc' facility, which is claimed to be equivalent to the addition of a second disc drive.

In addition to the main RAM, there is a separate 16K display RAM, and 32K of Boot ROM, which incorporates the BIOS. With MS-DOS loaded, 211K out of 256K are available for applications programs, unless the speech driver is in use, claiming a further 55K. There should be no need to complain of inadequate RAM space.

A printer may be driven by a Centronics port or via RS232C facilities, the latter also being usable for external communications. There is a 'pc/xi compatible' expansion slot, and a tone/noise generator for those who need waking up now and then.

PORTABILITY

A recent Sunday Magazine cartoon showed a portable computer being set up in the middle of the desert — only there was nowhere to put the mains plug! When we talk of a portable radio, we expect it to use internal battery power. That may even be true of a portable television set. So how do we define a portable computer?

For a start, it must not be so heavy that only a weight-lifter can carry it round. The Apricot Portable passes that test. It comes in a light-weight case, and with accessories should turn the scale at no more than 15lb.

Secondly, to be strictly correct, it should use internal power supplies. Otherwise it should be classed as 'Transportable'. That is not necessarily a slur. Anyone who needs to take a computer with him on his travels will usually be able to find a mains supply, unless his work is done on windy moors or by the seaside. He may even be thankful that there is no need to worry about the batteries running down at a critical moment — though he must remember to renew the keyboard cells from time to time.

So the transportable computer is a viable class, especially if it does not depend on degradation of performance to ahieve its status. By using the flat display, the Apricot version keeps performance high.

SOFTWARE

Compatibility with other Apricot machines means that there is ample software available. (See Table 1) There was no time to explore these items, however, exploration being based mainly on a tuition program, which raised an interesting point.

In accordance with recent practice, the menu selection procedures involved pointing to icons displayed on the screen and then pressing Enter. Now, this is perhaps an advantage for the newcomer, who may find the procedure intriguing and easier than selecting an item from a text list, but it could cause problems in the long run, when the user finds himself having to use standard procedures.

One essential is that the icons should be clear, and the Portable satisfies that need, though one or two patterns were difficult to identify.

CONCLUSIONS

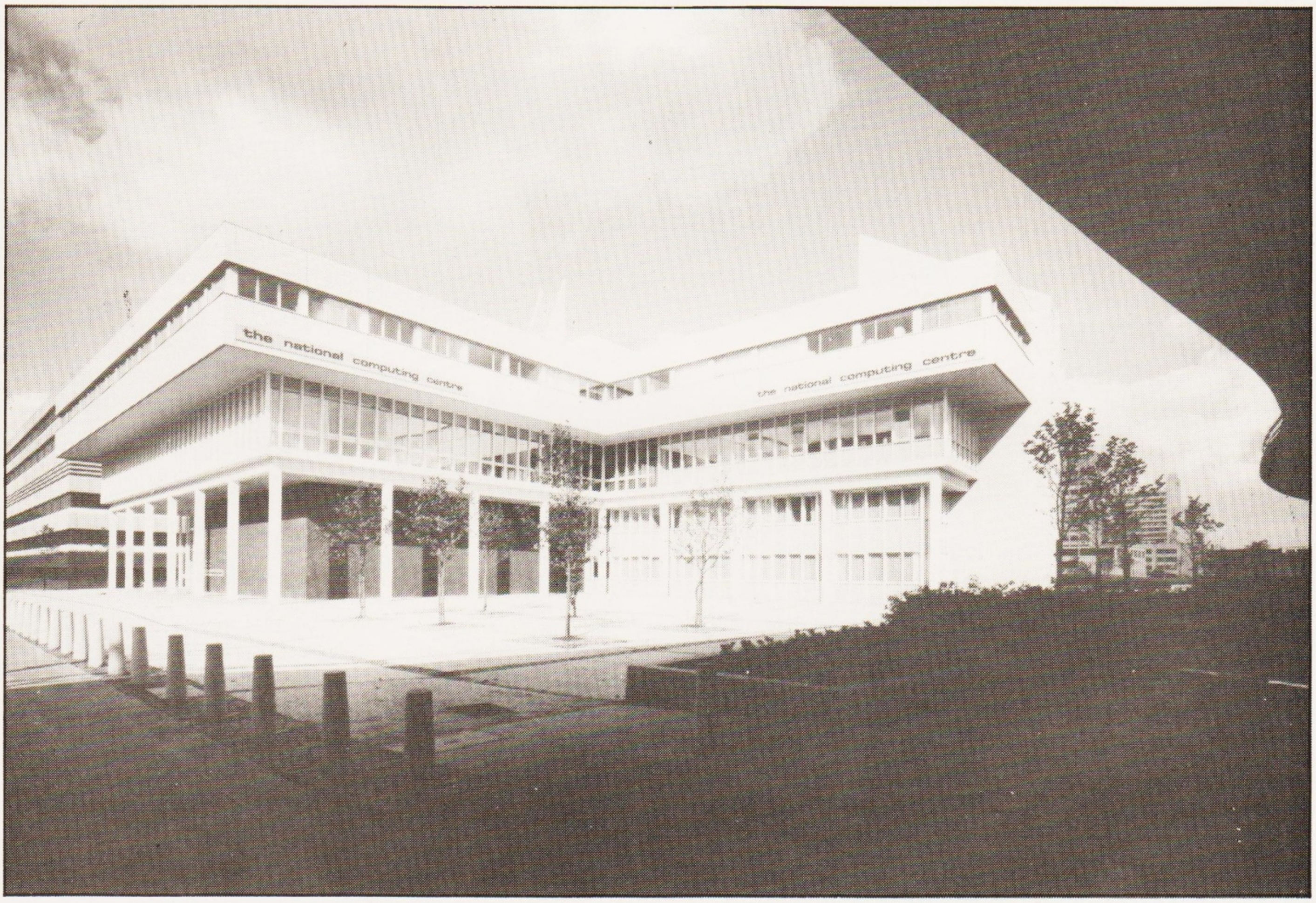
This quick look at a complex machine brought out a number of interesting points. It was concluded that the system was a good representative of the 'transportable' class, and a great deal better than some competing models. At £1695, it is not unduly expensive, and should be a steady seller.

ACKNOWLEDGEMENT

Thanks are due to **Ozwise Computers** for giving me the chance to investigate this interesting machine, and for helping me to learn about it quickly.

THE NATIONAL COMPUTING CENTRE

Jamie Clary



Try saying 'The National Computing Centre' to whoever happens to be nearby. Chances are the reaction will be 'What?' or 'Don't they vet software or something'. Computing Today had many questions about the NCC, so we went to Manchester to find out what they do and why they do it...

Situated off Manchester's Oxford Road, and just a stone's throw away from Manchester Polytechnic, the bright, white, National Computing Centre is impressive — and curiously inviting. Established by the Wilson Government in 1966, with the primary objective of "Promoting an increased and more effective use of computers in every field of national and commercial activity", the NCC has remained a touch enigmatic.

"What's that?" or "Well. I've heard of it" is the reply most will give when questioned about its existence. It was for this reason that *Computing Today* paid a visit to the Manchester headquarters. William (Bill) McCool, the centre's head of corporate publicity, very kindly showed us around the buildings — answering our many awkward questions as he went...

CT — The National Computing Centre was founded in 1966, to 'Promote an increased and more effective use of computers in every field of National and commercial activity'. Why did the government feel the time was right to establish such a centre?

Bill McCool — "The mid-60's was a time when phrases like 'white-hot technology' were being used to describe the output from the then burgeoning British computer industry. Harold Wilson, who was leading the Labour government at the time, realised that the new technology was going to mean an awful lot to us, and that the time was right to set-up an institute of some sort, to collect and disseminate information on computing. Hence, we have the National Computing Centre."

CT — But why was Manchester chosen as the site for the NCC? Wouldn't pressure have been put on the government to situate it elsewhere — in London, for example?

Bill McCool — "About the time that discussions were in progress to decide the site for the Centre, many Labour people were putting a lot of pressure on Harold, saying 'I think you should get out of London', and 'we think a alot of the things you're doing should be sited elsewhere'. Also, Harold's father went to UMIST across the road, and, of course, much of the developmental work into computing was done at Manchester University, so the city was chosen as a base for the Centre's activities."

CT — The NCC is officially listed as 'The National Computing Centre Limited'. When and why was there a parting of ways between yourselves and the Government?

Bill McCool — "In the early days, we were completed funded by Government, and at the end of the year any money we made was taken back by the Government. We subsequently became a membership organisation, but a membership organisation consisting of other organisations and not individuals. So, we have a membership of over 2000 organisations, like ICI, Shell, and so forth. This means that we probably have about 9000 listed names. With the passage of time, we became what was probably the biggest name for training within the sphere of Information Technology. We then started to make some money of our own in training, advisory services, and books, and in 1977 somebody decided it was time to break away from the Government, become self-funding, and 'stand on our own two feet' — hence 'NCC Limited'."

CT — How involved is the Government now in the everyday functioning of the NCC?

Bill McCool — "We still have Government representatives on the Board, and we still do projects for the Government, because there is national work which cannot be done by anybody else — the development and maintenance of Standards, for example."

NCC press officer, Bill McCool.

CT — How much Government, i.e. public, money do you receive for the services you provide?

Bill McCool — "We earn approximately ten million pounds from our publishing services, and we get about two million pounds from running schemes for the Government. We also receive about three million from the Government for project work, but they only provide fifty percent funding which leaves us to provide the remainder."

CT — What happens to the profits?

Bill McCool — "Everything that is bought from the NCC helps to fund the National role, beause our profit is ploughed back into our projects. We don't have any shareholders, so there are no dividends to be payed."

CT — Does it worry you that very few people are aware of the NCCs existence, and that those who know of you are uncertain about the extent, and perhaps the value, of your activities?

Bill McCool — "The problem we have is in identifying who our 'public' really are. We see our public as being the computer user, the computer manfuacturers, people in the Information Technology industry, and also the smallish firms — if we can help them. However, it costs a lot of money to make ourselves known to the general public, and we really only have enough money to service people within our own industry. So, if you are within the industry and you don't know about the NCC — then I have fears for us! Also, whenever we do so on television or radio, the switchboard is jammed for a week with callers wanting answers to their own, very specific, problems."

CT — One of the NCC's better-known roles in 'validating' software, ensuring it meets certain standards. How can you guarantee impartiality, given that the NCC itself generates software to compete in the same market as those products submitted for validation?

Bill McCool — "The validation service is part of an EEC project we are working on. This is tied in with the Federal Bureau of Standards in America, because all American government departments that buy-in software must ensure that the product has a validation license proving it satisfies certain criteria. As to the question of impartiality: our problem at the NCC is that we are too impartial. We've grown up with this impartiality, and in some ways it's our 'Achilles Heel'."

CT — How much of the NCC's work, if any, is sub-contracted?

Bill McCool — "We sub-contract some jobs, but when we began we had a much better idea. We used to get people in to do specific work, and we payed them for as long as the contract lasted. Once the contract finished they would go off and get a job elsewhere. This had two advantages: it meant that people would leave us on a regular basis and move around the industry; it also meant that we had a regular throughput of people — new people to do new things. But it eventually became difficult finding replacements of the right calibre for those finishing contracts, and so we began to employ people full time. This is not such a good thing, as we find that once a particular project is concluded, personnel are moved onto other projects which often becomes like trying to fit round pegs into square holes."

Major departments within the National Computing Centre

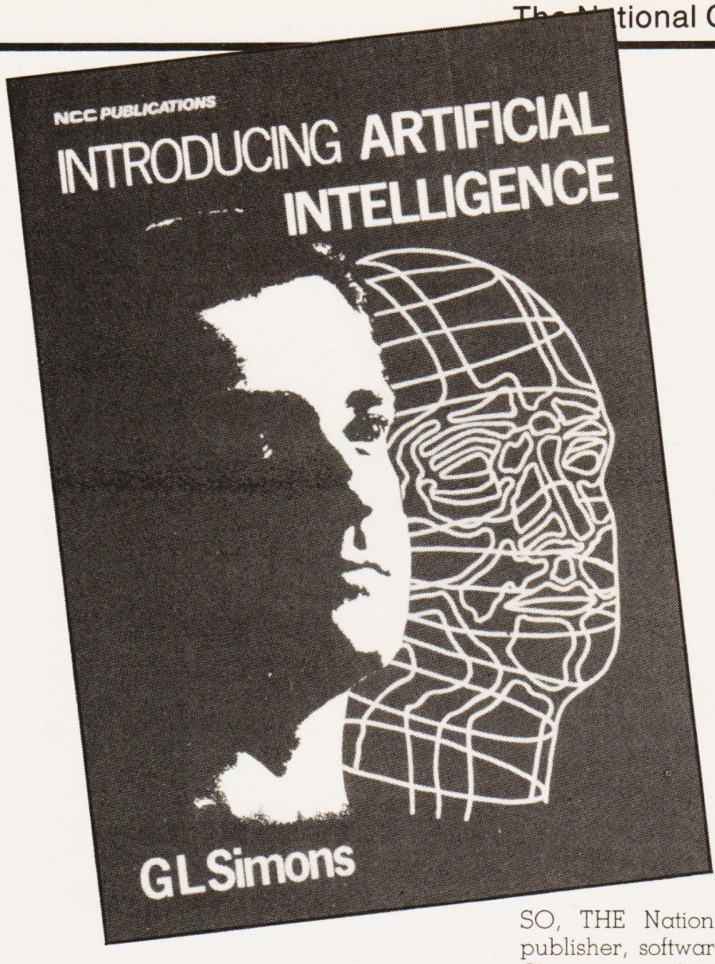
Government Schemes	End User Systems
Telecommunications	Sofdtware Engineering and Methods
Knowledge Management Systems	Training
Publications	Information Services
Seminars and Conferences	Mobile Exhibitions
Computer Based Training	

CT — Do you find it difficult to maintain your independence, given that NCC projects may sometimes coincide with work already being done by member counties represented on your Board of Directors?

Bill McCool — "Yes, there are difficulties. We have six regions for membership in Britain, and each region elects a chairman to sit on our Board as a representative for the users. We then have chairmen to represent the authors of the software, and finally we have representatives of the manufacturers. Sometimes objections are raised against projects simply because a representative's company is already involved in that area, so we are occasionally restricted in what we do."

CT — In conclusion, how would you like the general public to view the NCC?

Bill McCool — "We would like to be known as the Users' Champion. We defend the user more than anybody else, even if it means antagonising our manufacturers; even if it means antagonising our class C members (the software writers). If there's something wrong, we will shout about it, because we feel the user is the most important person."



NCC publications: from AI to IT.

SO, THE National Computing Centre: publisher, software factor, arbiter, Users' Champion and more; impartial, independent and democratic. A British invention that's doing well — and will be doing well, we hope, for some years to come...

Inside the NCC Microsystems Centre, London.

EVERY PICTURE

Seamus Dunn, University of Ulster

. . . tells a story, or so the saying goes. Representing information in pictorial form is enjoying a new era of popularity, particularly in areas like mathematics and statistics. This is where we come in...

The major contribution of the new technology to Education may come from a renewed emphasis on pictorial or graphic forms of communication. It has been argued that there was a time in our social past when the ability to 'read' pictures, and to communicate by means of non-verbal symbolic representations, was an important part of human culture. The effect of any cultural reversion of this sort on our approach to the teaching and learning of mathematics would have profound consequences for how we perceive the subject. Certainly some of its fundamental ideas are essentially pictorial, and the easiest and most natural way to 'know' them is to 'draw' them, or to 'look' at them, rather than to talk about them.

This is also related to the way in which the subject is organised for teaching. One of the great controversies associated with mathematics in schools relates to its structure and sequencing. The subject is normally presented as being completely hierarchical, so that when a new idea is to be presented it must rest heavily on a great many other ideas already assumed to be known and understood. This version of mathematics underlies the way in which most text-books are written and the way most courses are constructed, and so it is in a sense the official or generally accepted version. Indeed most of us were taught mathematics in this way for all of our school years, and this may be one of the reasons why we didn't actually learn very much of it.

INTUITIVE KNOWLEDGE

A little reflection might suggest that this emphasis on structure and sequence is not a very good description of the process of learning mathematics that most of us experience. Certainly many people seem capable of learning (or at least using) the subject in other less structured ways, and very often it is the learning acquired in this way that is remembered and made use of and valued. Another part of the problem is that, although what is learned (and so known) in this way is clearly mathematical, it is not always perceived as such. For example the very wide-ranging knowledge of practical geometry achieved by carpenters or by graphic artists; or the extraordinary number-skills shown by darts players; or the intuitive knowledge of number-bases and their inter-relationships shown by otherwise non-mathematical hackers: all of this knowledge is in some way excluded as exceptional or peripheral, and so not 'real' mathematics.

Modern forms of communication place great emphasis on the spoken and the written word. Despite television, and all the other forms of graphical media, the world still runs mainly on the language of words. In the case of mathematics this language has become so symbolic and highly compressed that communication is in the form of a series of combinations of a very few highly charged symbols. A proper understanding of these is available only to the (comparatively) few high priests.

So it may be that the arrival of the computer allows us to begin to change this. For perhaps the first time ever it is possible for everyone, using simple software and hardware tools, to try to develop and present ideas in graphic form and so to try to communication in this way. It also allows for the presentation of quite complex mathematical notions in a form which makes clear their own internal logic.

THE SOFTWARE

The piece of software that is now described

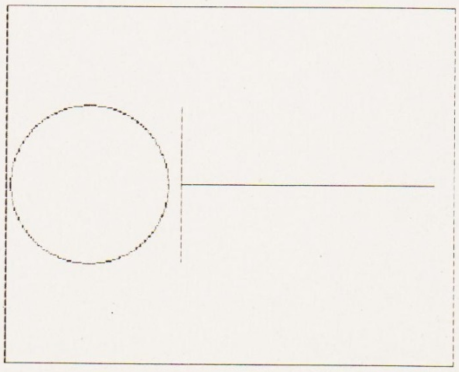


Figure 1

illustrates many of these issues and problems. It is made up of two programs, the first of which is called SINE1. It concerns the notion of a circular function in its simplest form, as a SINE function. It simply presents on the screen the following sequence:

1. A circle of unit radius is drawn very quickly on the left.
2. The positive X axis and a general Y axis are presented on the right, with the X axis and the centre of the circle on the same horizontal level. (See figure 1).
3. A right-angled triangle then appears within the circle on the left, made up of a radius as the hypotenuse, and its horizontal and vertical projections as the other two sides. The vertical projection is the SINE of the central angle, which is 10 degrees to start with. This projection is coloured blue to distinguish it from the other lines. At the same time this blue line is also drawn as a value of Y on the X axis. (See figure 2).
4. A box then appears at the top of the screen with words as shown below:

   Angle in degrees is 10
   Sine of angle is 0.173

5. At the bottom of the screen a message appears as follows:

   CIRCULAR FUNCTION: SINE CURVE
   To stop or start the movement press the space bar.

6. Immediately the triangle disappears and a new one appears with a centre angle of 20 degrees. Again the blue line projection also appears as a value of Y on the X axis, and the message at the top of the screen changes to show that the angle is now 20 degrees.
7. This process continues. Each time a new triangle appears the centre angle is 10 degrees larger, and this repeats until 360 degrees is reached. At the same time a SINE curve is produced on the X axis by the blue lines.

Figure 3 shows the picture when the angle is 230 degrees.

8. It is possible to freeze the screen at any time by pressing the spacebar, and the opportunity to see it all again is presented at the end.

The idea is that, by watching this dynamic creation of a SINE curve, it is possible to develop an intuitive feel for its meanings and properties. It could be argued that this kind of presentation should represent a student's first experience of these notions, and that the normal approach to trigonometric ratios should come later. The idea that each angle size has associated with it a corresponding number also changes in an orderly fashion, is sensible and intuitively satisfying. The possibility of isolating one of these results and looking at it in more detail as an example of the rest then follows, and this then leads into the normal approaches to trigonometric ratios.

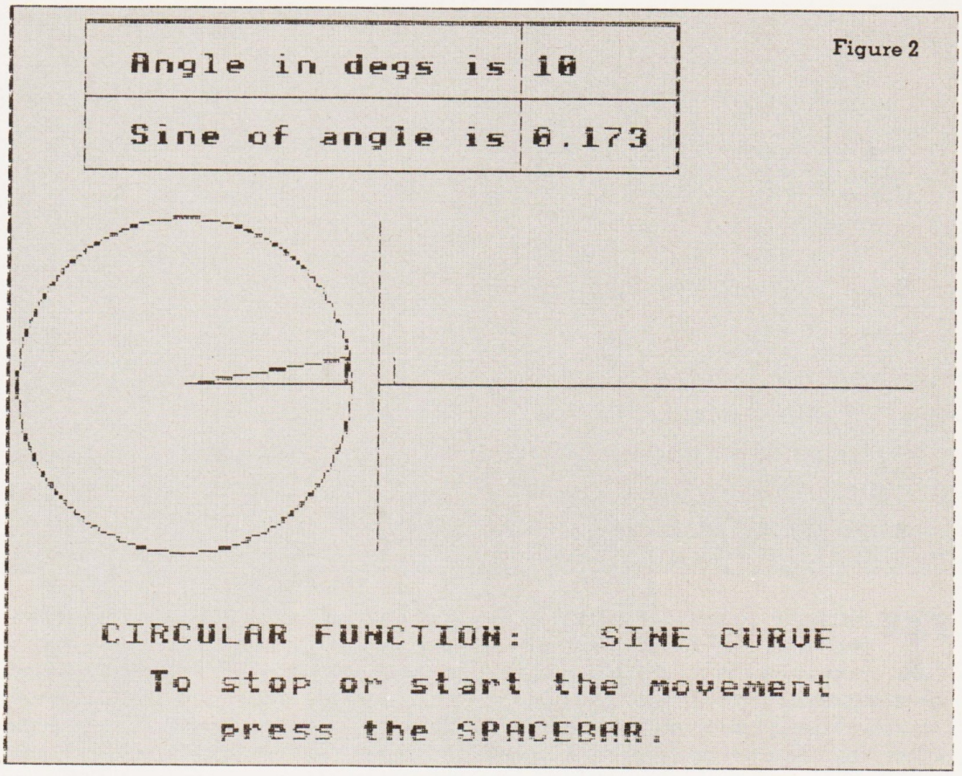


Figure 2

Perhaps the most confusing and difficult part of the experience is the contrast in the two forms of representation being used for the measurement of the angle. On the one hand, an angle of size 30 degrees (for example) is represented as a measure of turning within a circle (on the left in the picture): on the other hand it is represented as a measure of the length of a displacement along the X axis (on the right in the picture). Experience suggests that the attempt to be clear about this with words rarely succeeds, until the experience of watching it has been achieved and repeated. The concept is a visual one with its own internal logic.

The whole range of possible further developments of the mathematics of circular functions, and their associated curves, is not shown here, but there is clearly enormous potential for this kind of graphical approach. As one example, this program then goes on to demonstrate the interesting notion (perhaps being experienced for the first time) that some functions have more than one 'solution'. So if we know the numerical value of a SINE then, between 0 and 360 degrees, there are usually four possible associated angles.

The second program (SINE2) demonstrates this by randomly choosing an angle between 0 and 90 degrees. This is then presented in the circle on the left as before, that is as the centre angle of a right-angles triangle. It is also presented on the right as point on the sine curve. The three other possible positions on the curve associated with this number are also presented. The user is then asked to find the numbers associated with the three angles. Written symbolically the exercise should lead to the idea that:

$$\begin{aligned} SIN(A) &= SIN(180 - A) \\ &= -SIN(180+A) \\ &= -SIN(360 - A) \end{aligned}$$

But this final representation in symbolic form is only one way of 'knowing' this, and for many pupils it is not the most accessible or the most useful or the most satisfactory. As we suggested at the beginning, modern society has grown to value the ability to present 'what is known' in algebraic form (and therefore in a sense in verbal form) and there are obviously good reasons for this, and there is no suggestion that it can be avoided. But one of the consequences of the new technology may be that we will learn (or re-learn) how to communicate in visual and graphic terms.

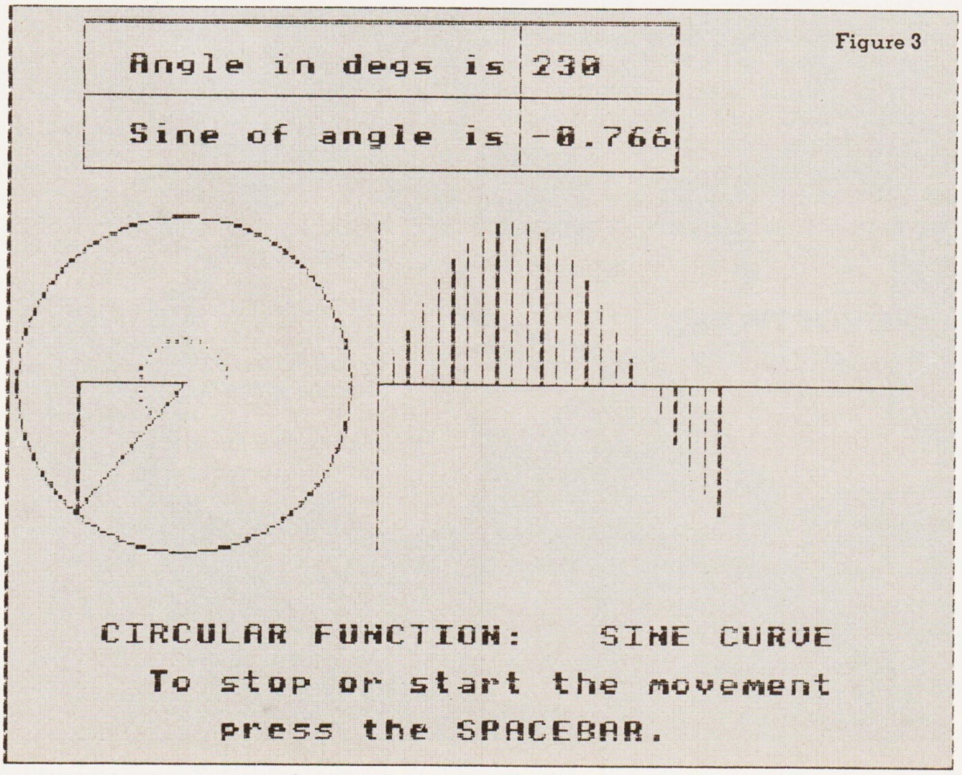


Figure 3

The program is in three parts, one of which is very short and is used once only. It is called LOADER and is shown as Listing 1. Because the program involves repeatedly drawing circles it was thought best to make this process as fast as possible, so this program calculates the appropriate set of coordinates once only and stores these in a data file. This file is then called by the other programs when necessary, and so enables circles to be produced with great speed. The data-file (called VALUES) is generated by this first short program, which can then be abandoned.

There are two other programs, labelled SINE1 and SINE2, and these are shown as Listing 2 and Listing 3. These have been separated because of memory limitations, and the second is chained automatically by the first when needed. They have both been written in a carefully structured fashion, and an attempt has been made within the limitations of memory to make them as easy to read and understand as possible.

Listing 1: loader

```
10 REM LOADER
20 REM TEMP PROGRAM
30 REM TO CREATE DATA FILE "VALUES"
40 REM Seamus Dunn (C), 1984
50 *DISK
60 DIM s(100),c(100)
70 X=OPENOUT"VALUES"
80 FOR count%=0 TO 90
90   s(count%)=SIN(RAD(count%))
100   c(count%)=COS(RAD(count%))
110   PRINT#X,s(count%)
120   PRINT#X,c(count%)
130   NEXT count%
140 CLOSE#X
```

Listing 2: SINE1

```
10 REM SINE1
20 REM SEAMUS DUNN (C),1984
30 ON ERROR GOTO 1400
40 *DISK
50 DIM s(180),c(180)
60 MODE 7:PROCintro
70 MODE1:PROCcolour
80 PROCvars
90 PROCcircle
100 PROCmessage1
110 PROCaxes
120 GCOL0,3
130 PROCsine
140 PROCmessage2
150 IF flag=1 THEN 70
160 IF flag<>2 THEN 140
170 IF flag1=1 THEN 180 ELSE 190
180 CHAIN"SINE2"
190 MODE 7:END
200 REM********************************
210 DEF PROCintro
220 PRINT CHR$128;CHR$157;CHR$129;CHR$141;"        S I N E   C U R V E"
230 PRINT CHR$128;CHR$157;CHR$129;CHR$141;"        S I N E   C U R V E"
240 PRINT CHR$130;CHR$157
250 PRINT CHR$130;CHR$157
260 PRINT CHR$128;CHR$157;CHR$132;"This program allows you to look "
270 PRINT CHR$128;CHR$157;CHR$132;"carefully at the following:"
280 PRINT' CHR$128;CHR$157;CHR$132;"     1.  A circle of radius one."
290 PRINT CHR$128;CHR$157;CHR$132;"     2.  A moving radius."
300 PRINT CHR$128;CHR$157;CHR$132;"     3.  A sine curve."
310 PRINT' CHR$128;CHR$157;CHR$132;"As the radius rotates, the angle at"
320 PRINT CHR$128;CHR$157;CHR$132;"the centre slowly increases in size."
330 PRINT CHR$128;CHR$157;CHR$132;"The blue line is the SINE of the"
340 PRINT CHR$128;CHR$157;CHR$132;"angle, and this is reproduced on the"
350 PRINT CHR$128;CHR$157;CHR$132;"curve on the right, which slowly"
360 PRINT CHR$128;CHR$157;CHR$132;"turns into a SINE curve."
370 PRINT TAB(6,21);"Press the spacebar to watch this."
380 PRINT TAB(7,22);"There may be a slight delay."
390 PROCcalc
400 Z=GET
410 ENDPROC
420 DEF PROCcalc
430 X=OPENUP"VALUES"
440 FOR count%=0 TO 90
450   INPUT#X,s(count%)
460   INPUT#X,c(count%)
470   NEXT count%
480 CLOSE#X
490 FOR count%=90 TO 180
500   s(count%)=s(180-count%)
510   c(count%)=-c(180-count%)
520   NEXT
530 ENDPROC
540 DEF PROCcolour
550 VDU 19,0,1;0;
560 VDU 19,1,4;0;
570 VDU 19,3,7;0;
580 GCOL0,2
590 ENDPROC
600 DEF PROCvars
610 n=0:xcentre=240:ycentre=512
```

```
620 radius=220:xorigin=500:yorigin=512
630 ENDPROC
640 DEF PROCcircle
650 radius1=radius+5
660 MOVE xcentre+radius,ycentre
670 FOR count=1 TO 180
680   PLOT 5,xcentre+(radius1)*c(count),ycentre+(radius1)*s(count)
690   NEXT count
700 FOR count=180 TO 1 STEP -1
710   PLOT 5,xcentre+(radius1)*c(count),ycentre+(radius1)*(-s(count))
720   NEXT count
730 ENDPROC
740 DEF PROCmessage1
750 PRINT TAB(4,26)"CIRCULAR FUNCTION:"
760 PRINT TAB(25,26)"SINE CURVE"
770 PRINT TAB(6,28)"To stop or start the movement"
780 PRINT TAB(9,30)"press the SPACEBAR."
790 ENDPROC
800 DEF PROCaxes
810 MOVE xorigin,yorigin:DRAW xorigin+720,yorigin
820 MOVE xorigin,yorigin-radius:DRAW xorigin,yorigin+radius
830 ENDPROC
840 DEF PROCsine
850 n=10:m=20
860 FOR count=1 TO 36
870   col=5:col1=85
880   PROCradii
890   PROCcurve
900   PROCmessage
910   col=7:col1=87
920   PROCradii
930   NEXT count
940 ENDPROC
950 DEF PROCradii
960 yy=1
970 MOVE xcentre,ycentre
980 nc=n*count
990 IF nc>180 THEN nc=360-nc:yy=-1
1000 PLOT col,xcentre+radius*c(nc),ycentre
1010 GCOL0,1
1020 PLOT col,xcentre+radius*c(nc),ycentre+yy*radius*s(nc)
1030 GCOL0,3
1040 PLOTcol,xcentre,ycentre
1050 PLOT 69,xcentre+60*c(nc),ycentre+60*yy*s(nc)
1060 ENDPROC
1070 DEF PROCcurve
1080 yy=1
1090 nc=n*count
1100 IF nc>180 THEN nc=360-nc:yy=-1
1110 GCOL 0,1
1120 MOVE xorigin+m*count,yorigin
1130 DRAW xorigin+m*count,yorigin+yy*radius*s(nc)
1140 GCOL 0,3
1150 ENDPROC
1160 DEF PROCmessage
1170 aa=1
1180 ang=10*count
1190 IF ang>180 THEN ang=360-ang:aa=-1
1200 sin=INT(1000*s(ang))/1000
1210 PRINT TAB(5,2)"Angle in degs is ";10*count
1220 PRINT TAB(5,5)"                              "
1230 PRINT TAB(5,5)"Sine of angle is ";aa*sin
1240 PRINT TAB(8,27);"                              "
1250 MOVE100,800:DRAW100,1000:DRAW900,1000:DRAW900,800
1260 DRAW100,800:MOVE100,900:DRAW900,900:MOVE 690,800:DRAW 690,1000
1270 SS=INKEY(30)
1280 IF SS=32 THEN Z=GET
1290 ENDPROC
1300 DEF PROCmessage2
1310 PRINT TAB(4,28)"Do you want to see that again?          "
1320 PRINT TAB(4,30)"                                          "
1330 INPUT TAB(15,30)"Input Y or N "ans$
1340 IF ans$="Y" OR ans$="y" THEN flag=1 ELSE flag=2
1350 IF flag=1 THEN flag1=2:GOTO 1390
1360 PRINT TAB(4,30)"                                          "
1370 INPUT TAB(4,28)"Do you want to go on? Y or N ",ans$
1380 IF ans$="Y" OR ans$="y" THEN flag1=1 ELSE flag1=2
1390 ENDPROC
1400 MODE 7
1410 *.PRINT"Error Number ";ERR;" On line ";ERL
```

Listing 3: SINE2

```
10 REM SINE2
20 REM Seamus Dunn (c), 1984
30 ON ERROR GOTO 1500
40 *DISK
50 DIM s(180),c(180)
60 MODE 7:PROCintro
70 PROCvars
80 REPEAT
90   MODE1:PROCcolour
100   GCOL0,2
110   PROCcircle
120   PROCaxes
130   PROCsine
140   PROCproblem
150   UNTIL W
160 MODE 7:END
170 DEF PROCintro
180 PRINT CHR$128;CHR$157;CHR$129;CHR$141;"         S I N E   C U R V E"
190 PRINT CHR$128;CHR$157;CHR$129;CHR$141;"         S I N E   C U R V E"
200 PRINT CHR$130;CHR$157:PRINTCHR$130;CHR$157
210 PRINT CHR$128;CHR$157;CHR$132;"This program chooses at random"
220 PRINT CHR$128;CHR$157;CHR$132;"an angle between 0 and 90 degrees."
230 PRINT CHR$128;CHR$157;CHR$132;"It then draws the SINE curve between"
240 PRINT CHR$128;CHR$157;CHR$132;"0 and 360 degrees.  It then shows"
250 PRINT CHR$128;CHR$157;CHR$132;"the four points on the curve"
260 PRINT CHR$128;CHR$157;CHR$132;"where the sines are numerically"
270 PRINT CHR$128;CHR$157;CHR$132;"equal.  It then asks you to input"
280 PRINT CHR$128;CHR$157;CHR$132;"the values for these."
290 PRINT TAB(6,21);"Press the spacebar to watch this."
300 PRINT TAB(7,22);"There may be a slight delay."
310 PROCcalc
320 Z=GET
330 ENDPROC
340 DEF PROCcalc
350 X=OPENUP"VALUES"
360 FOR c%=0 TO 90
370   INPUT#X,s(c%)
380   INPUT#X,c(c%)
390   NEXT c%
400 CLOSE#X
410 FOR c%=90 TO 180
420   s(c%)=s(180-c%)
430   c(c%)=-c(180-c%)
440   NEXT c%
450 ENDPROC
460 DEF PROCvars
470 n=0:xcentre=240:ycentre=400
480 radius=220:xorigin=500:yorigin=400
490 ENDPROC
500 DEF PROCcolour
510 VDU 19,0,4;0;
520 VDU 19,2,0;0;
530 ENDPROC
540 DEF PROCcircle
550 radius1=radius+5
560 MOVE xcentre+radius,ycentre
570 FOR c%=1 TO 180
580   PLOT 5,xcentre+(radius1)*c(c%),ycentre+(radius1)*s(c%)
590   NEXT c%
600 FOR c%=180 TO 1 STEP -1
610   PLOT 5,xcentre+(radius1)*c(c%),ycentre+(radius1)*(-s(c%))
620   NEXT c%
630 ENDPROC
640 DEF PROCaxes
650 MOVE xorigin,yorigin:DRAW xorigin+720,yorigin
660 MOVE xorigin,yorigin-radius:DRAWxorigin,yorigin+radius
670 ENDPROC
680 DEF PROCsine
690 n=1:m=10
700 FOR c%=1 TO 72
710   col=5
720   PROCcurve1
730   NEXT c%
740 ENDPROC
750 DEF PROCcurve1
760 yy=1
770 nc=5*c%
```

```
780 IF nc>180 THEN nc=360-nc:yy=-1
790 GCOL 0,1
800 PLOT69,xorigin+m*c%,yorigin+yy*radius*s(nc)
810 GCOL 0,3
820 ENDPROC
830 DEF PROCproblem
840 c%=RND(60)+15
850 c0%=c%
860 ang%=c%
870 sin=INT(1000*s(ang%))/1000
880 PRINT TAB(1,1)"Ang at 1 in deg is ";c%
890 PRINT TAB(1,3)"Sine of angle is ";sin
900 PROCradii
910 num$="1"
920 PROCcurve2
930 PRINT TAB(1,5)"Press spacebar for"
940 PRINT TAB(1,6)"3 other angles with"
950 PRINT TAB(1,7)"sines that are equal"
960 PRINT TAB(1,8)"numerically to the"
970 PRINT TAB(4,10)"angle ";c%
980 Z=GET
990 c1%=180-c0%:c%=c1%:num$="2":PROCcurve2
1000 c2%=180+c0%:c%=c2%:num$="3":PROCcurve2
1010 c3%=360-c0%:c%=c3%:num$="4":PROCcurve2
1020 VDU5:MOVE 790,970: DRAW1220,970:DRAW1220,660:DRAW 790,660: DRAW
790,970:VDU4
1030 PRINT TAB(25,2)"Num Ang  Resp"
1040 PRINT TAB(25,4)" 1          "
1050 PRINT TAB(25,6)" 2          "
1060 PRINT TAB(25,8)" 3          "
1070 PRINT TAB(25,10)" 4          "
1080 PRINT TAB(29,4);ang%
1090 a=6:cc%=c1%
1100 PROCcheck
1110 a=8:cc%=c2%
1120 PROCcheck
1130 a=10:cc%=c3%
1140 PROCcheck
1150 INPUT TAB(3,30)"Repeat?  Y or N ",qq$
1160 IF qq$="Y" OR qq$="y" THEN W=0 ELSE W=-1
1170 ENDPROC
1180 DEF PROCradii
1190 yy=1
1200 MOVE xcentre,ycentre
1210 nc=c%
1220 IF nc>180 THEN nc=360-nc:yy=-1
1230 PLOT col,xcentre+radius*c(nc),ycentre
1240 GCOL0,1
1250 PLOT col,xcentre+radius*c(nc),ycentre+yy*radius*s(nc)
1260 GCOL0,3
1270 PLOT col,xcentre,ycentre
1280 ENDPROC
1290 DEF PROCcurve2
1300 GCOL 0,1
1310 yy=1
1320 nc=c%
1330 IF nc>180 THEN nc=360-nc:yy=-1
1340 MOVE xorigin+2*c%,yorigin
1350 DRAW xorigin+2*c%,yorigin+yy*radius*sin
1360 GCOL 0,3
1370 VDU5:MOVE xorigin+2*c%+10,yorigin-10:PRINT;num$:VDU4
1380 ENDPROC
1390 DEF PROCcheck
1400 PRINT TAB(10,30);"                 "
1410 PRINT TAB(28,a);"       "
1420 PRINT TAB(5,28)"Enter the size of angle ";(a-2)/2
1430 INPUT TAB(28,a),ans
1440 PRINT TAB(5,28)"                                "
1450 IF ans=cc% THEN PRINT TAB(34,a);"OK":GOTO 1490
1460 IF ans<>cc% THEN PRINT TAB(10,30);"NO.   TRY AGAIN"
1470 FOR NN=1 TO 1000:NEXT NN
1480 GOTO 1400
1490 ENDPROC
1500 MODE 7
1510 PRINT"Error number ";ERR;" on line ";ERL
```

Subscriptions

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

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

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

COMPUTING TODAY Subscriptions,

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

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

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1985 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :

COMPUTING TODAY Subscriptions

INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

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

SUBSCRIPTION RATES (tick ☐ as appropriate)

- £16.20 for 12 issues UK ☐
- £18.70 for 12 issues Overseas Surface ☐
- £51.20 for 12 issues Overseas Air Mail ☐

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

BARCLAYCARD VISA

Please use BLOCK CAPITALS and include postcodes.

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

ADDRESS ....

....

.... POSTCODE ....

Signature ....

Date ....

CT May '85

BUILD IT!

Geoff Parselle

Make your micro go places with this DIY computer trolley.

Computers:- fun machines, serious machines, teaching machines, useful or useless, but where do you keep and use them? Stacked away in cupboards, under the stairs, in boxes, on top of the television, or on the dining table! Certainly a nuisance, usually in the way, often difficult to use in comfort or peace and quiet.

Many people use their computer with a second T.V., be it an older secondhand set or a portable black and white model. If you do then maybe the D.I.Y. computer trolley shown here could be of use to you.

The trolley is so designed that it will carry a smallish television, a computer, cassette deck or disc drive, joystick and various books, cassettes, etc. It also has a lift up side flap to rest manuals, written programs, etc. whilst typing in. The whole assembly is on castors for ease of movement.

THE FRAME

The basic assembly is built around a framework of planed 2" x 2" timber, with laminated chipboard for the shelves.

Look at Photo 1 to see the basic frame, made, as stated, from planed 2" x 2" whitewood timber. The uprights (a) are 31½" long with cross pieces (b) of 20", and base legs (v) to which are fitted the castors, or 20" length. It is important to buy your wood as straight as possible, with as few knots as can be found and cleanly planed. When cutting the wood to length, measure carefully on all sides before marking, and use a set square. Cut using a fine tenon saw, slowly so as not to split the wood as you come out at the far side. Better in fact to cut part way through and then start again from the other side, finishing off the cut at the centre of the wood.

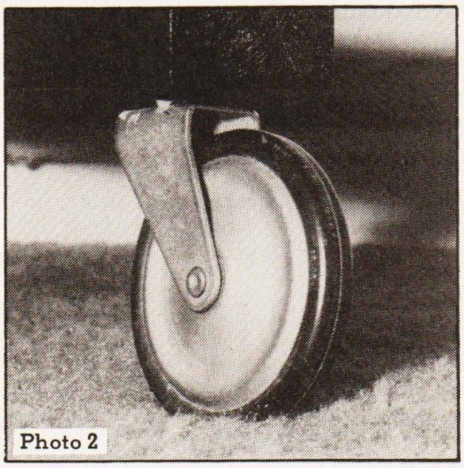


Photo 2

Once all the 2" x 2" wood has been cut to length correctly and cleaned up using medium sandpaper, the cross pieces can be fitted to the uprights. Lay the pieces on a flat base (the floor?) in the correct positions, i.e. the top cross piece (b) level with one end of the uprights (a) and the other cross piece 6" from the other end, again see photo 1. Mark the uprights and drill through from the outside, for four No. 8 countersunk screws — two per side per crosspiece. Countersink well below the wood surface. Crosspieces must have guide holes drilled to stop the wood splitting using an ⅛" or 2mm drill. Now screw together with 3" no. 8 steel countersunk screws (total 8) coating joints with good wood glue. Make sure the frame is not twisted using a set square before finally tightening the screws. Now leave for at least a couple of hours, lying on the flat floor until the wood glue has dried.

Whilst this assembly is drying, the castors can be fitted to the base legs (c) (see photo 2.) Whether you use sunken or flush fitting castors is up to you, but fit them centrally 2" in from the ends of the base legs. A total of four castors are required. I used sunken castors as I felt they would support the weight of the equipment better, but they do mean careful drilling of the wood. If you use flush fitting castors then pre-drill the wood to stop it splitting using a ¼" or 2mm drill — 4 screws per castor. If you use sunken castors, then once drilled to the correct depth and width,

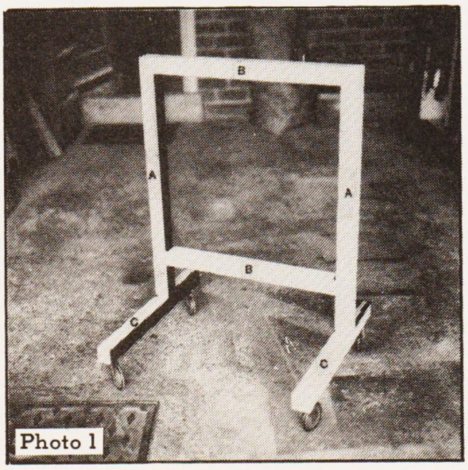


Photo 1

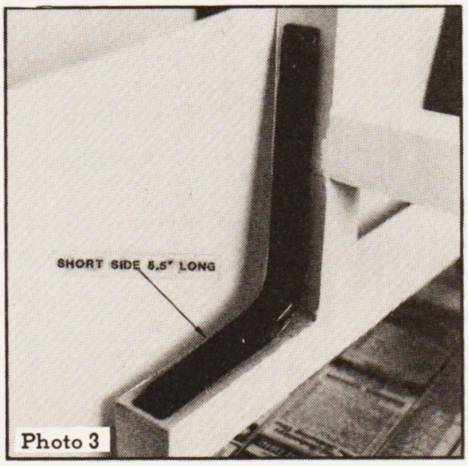


Photo 3

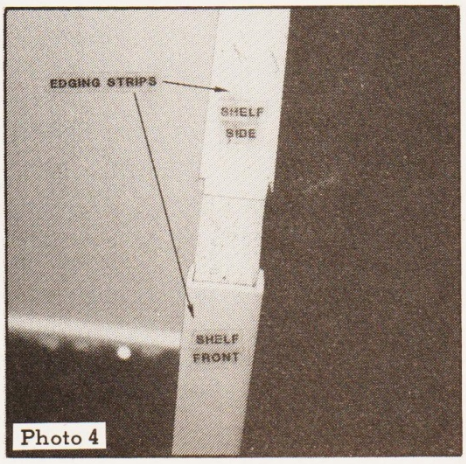


Photo 4

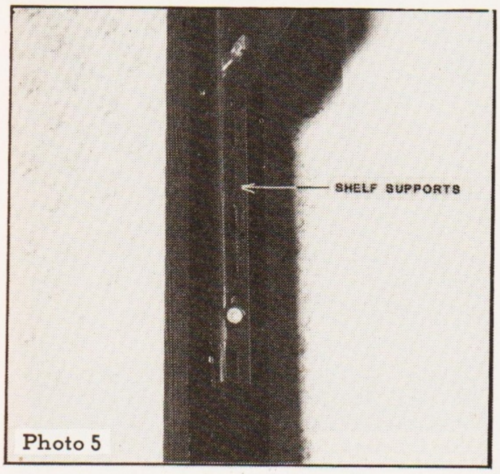


Photo 5

hammer in the swivel pieces, supporting the base legs carefully so as not to mark them.

Once the castors are fitted and the glue has dried on the frame assembly, the uprights can be fitted to the base legs, at a distance of 6" from the rear end of each base leg. As before, drill, countersink, and pre-drill, from underneath the base leg. Two 3" no. 8 steel countersunk screws for each upright. Again use good wood glue in the joint, but before this has dried, fit steel right angle brackets between the base pieces and the uprights, on the back and front of the uprights — photo 3. Brackets purchased (quantity 4) are approx. 5½" long on their short leg, and this short leg is fitted onto the base pieces with the longer leg against the uprights. Tighten all screws well, and leave this assembly to dry overnight.

When the glue has dried, fill all the screw holes with plastic wood, and when hard, rub down flush with the wood. Also rub down the wood with a medium sandpaper until smooth. Remove all dust with a cloth lightly dipped in white spirit and painton a good coat of wood primer, followed by undercoat and a top coat of your chosen colour. I used gloss black to give a high tech finish.

This whole assembly can now be put to one side to dry for a few days.

SHELVES

Whilst the framework is drying the shelves can be made. I used 15" plastic laminated chipboard (Contiboard) cut to 2ft. lengths. Mark and cut very carefully, scoring both sides of the shelf through the laminate to stop it chipping, using a tungsten laminate cutter. Cut gently using a sharp, fine saw and take your time. Contiboard is available in a number of finishes. I used magnolia which is a good deal less stark than white. The lower shelf also has to be cut down from 15" width to 12", to help in the fitting of the operator's knees when the trolley is in use.

The cut edges of the shelves have to be edged using either iron-on or push-on edging of the same colour. Iron-on edging has to be cut slightly oversize, then follow instructions on the packet. Use a hot iron with a single piece of cloth between the iron and the edging and press down quite hard, running the full length of the edging. When the builtin glue has set, trim off excess edging using a sharp knife (Stanley type with a new blade) and rub down with a fine sand paper.

I decided instead to use push on edging on all sides of the shelves (except between the centre shelf and the lift-up flap where iron-on edging was used). The push-on edging was cut as shown in photo 4 to save attempting to cut 45° angles.

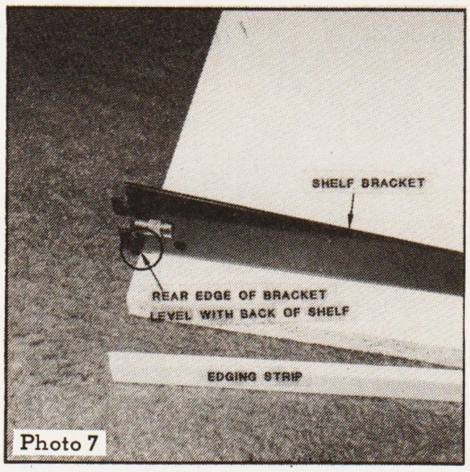


Photo 7

The lift up flap — d — photos 8 and 9, was also cut from Contiboard and is 12" x 12" in size, edged as explained above, but with iron-on edging on its hinged side only.

SHELF SUPPORTS

By now the main framework will be dry and the shelf supports can be fixed. I purchased adjustable steel brackets and supports — photos 5, 6 and 7, as I could then raise or lower the centre and lower shelves as I wished. One support for each upright is needed and they were purchased slightly longer than the height of the uprights, as, to fit the top shelf correctly, they need to be cut across half way through one of the slots so as to allow the support to fit at a position that enables the top shelf to be rested directly on, and screwed to, the upper crosspiece — photo 6. I used 1½" no. 6 countersunk crossheaded screws to fit the supports to the uprights, lined up centrally down the front of each upright with the cut through end just below the top of the frame assembly. Three pairs of shelving brackets were purchased

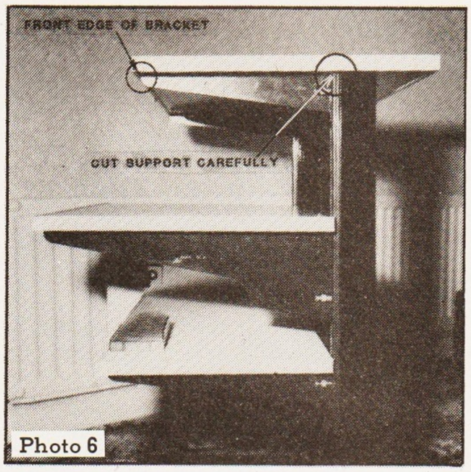


Photo 6

to fit the supports. The pair for the top shelf measure 9¾ inches length and are fitted so that the front of the bracket is positioned at the front edge of the top shelf — photo 6. The brackets are fitted with ½ inch no. 4 black round head screws.

Now, prior to fitting the brackets to the underneath of the top shelf, measure the distance between the centres of the uprights which should be 21⅞". Measure 12" from the end of the shelf and mark the point. Check that this point is also 12" from the other end of the shelf — i.e. the centre. Fit the brackets 10-15/16" from the marked point on each side, but confirm the brackets will fit the upright supports before fitting the screws, as slight adjustments may be needed. The top shelf is now fitted to the supports — and rested on the top crosspiece. Once fitted, drill through the top shelf into the top crosspiece at two points and using 1½" no. 8 countersunk steel screws, screw the top shelf to the crosspiece. Instead of countersinking using a countersunk bit, use a larger drill, suitable to enable the fitting of two plastic screw caps, the same colour as the shelving.

Brackets now need fitting to the other two shelves. The brackets for the centre shelf are 14¾" long and those for the bottom shelf 11¾" long. They are fitted the same distance apart as those of the top shelf, but with their near edges level with the back of the shelves this time — photo 7.

These shelves can now be fitted to the trolley at whatever heights you wish, essentially completing the assembly except for

Photo 8

Photo 9

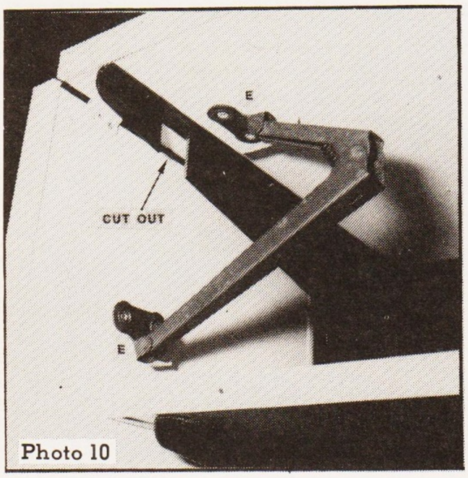


Photo 10

the fitting of the lift up flap. This obviously must hinge against one end of the centre shelf — d — photos 8 and 9. I fitted a couple of small 1'' plastic hinges screwed to the underneath of both the middle shelf and the flap, again pre-drill carefully and use ½'' no. 4 countersunk steel screws.

Fit the hinges 1'' from each end of the flap at the edge due to face the middle shelf. To do this, lay the shelf down on its top with the flap against it, correctly positioned, and fit the hinges.

SPRING SUPPORT

A spring support must now be fitted between the shelf and flap to support the flap when lifted. This type of spring support is normally used on dining table end flaps and there are a number of different types available, though all basically similar. The one I used had to be mounted on small wood blocks — e — photos 10 and 11, screwed under the shelf and flap. I also had to cut a piece out of the shelf bracket, using a hacksaw, to accommodate the spring support when the flap is in its upper position. The wood blocks were drilled-through so that the spring support fixing screws went right into the middle shelf. The wood blocks were also glued to the shelf using contact adhesive (Evo-stick).

That completes the trolley. Mine carries a 17'' colour television on the top shelf, a BBC 'B' computer and a cassette deck on the centre shelf, and various other items, cassettes and books on the bottom shelf. In fact I have now made and fitted a drawer under the bottom shelf to hold cassettes etc. Slide under drawers are available to purchase, which would be far easier than attempting to make them. I have also clipped a light to the top shelf, shining on the computer keys — quite handy for us non-touch typists.

Total cost of the trolley is approximately £35/40 which is a lot less than any commercially available comparable models that I know of.

MATERIALS USED

2 x 2 metre lengths 2'' x 2'' planed soft wood.
8ft x 15'' Contiboard
Set castors
Four right angle brackets
12ft iron-on edging or
30'' iron-on and 24ft push-on edging

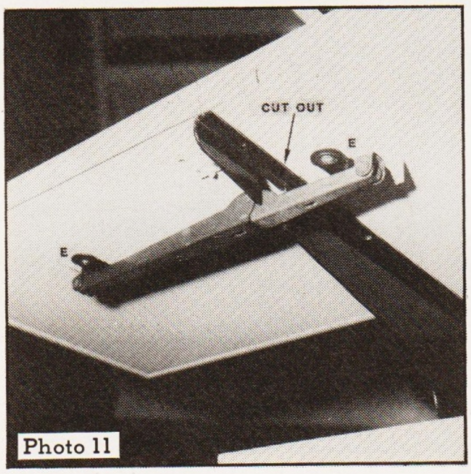


Photo 11


The finished article

2 x 36'' uprights for adjustable shelving
2 x 9¾'' brackets
2 x 14¾'' brackets
2 x 11¾'' brackets
2 x 1'' plastic hinges
1 x spring support
12 x 3'' no. 8 countersunk steel woodscrews
12 x ½'' no. 4 black round head wood screws
2 x 1½'' no. 8 countersunk steel woodscrews
8 x ½'' no. 4 steel countersunk wood screws
4 x 1'' no. 6 countersunk steel wood screws
8 x 1½'' no. 6 cross headed countersunk wood screws
Packet plastic screw caps
Wood glue, Evostick,
Medium & fine sandpaper, primer, undercoat, top coat

Note: If desired the shelves can be made 28 inches in length if you use a large tape recorder.

MORE I/O

Peter Simpson, GIKGC

I/O expansion can be a costly process — unless you do-it-yourself, which is precisely what is suggested here: an I/O expansion module for the Sharp MZ80K.

No matter what other personal-computer users may say, the MZ80K still lives! One disadvantage, however, is the lack of input/output and the Sharp requires the Expansion Unit and a suitable I/O card to add it. It is then that the expense starts. To overcome this problem, I buried myself in manuals, locked myself in my workshop and finally produced the contraption described here.

It is a cheap method of interfacing simple devices to the Sharp, and provides an eight-bit input and an eight-bit output port *without* using the rear connector, but people who don't want to delve inside the Sharp and do some soldering onto the mother-board should look elsewhere!

THEORY

To control the music, the keyboard and the cassette, to name a few things, an 8255 PPI (Programmable Peripheral Interface) is used, which provides three input/output ports. Two of these ports control the keyboard in the manner shown in Fig. 1.

It is the two ports A and B which hold the key to the design. Sharp thoughtfully included another set of holes in the main PCB, just by the keyboard plug, into which a parallel keyboard socket can be soldered. This allows the hardware enthusiast to use his/her own keyboard instead of the horrible one on the machine (the only fault I will admit to!). It also allows access to the two ports already mentioned.

Fig. 1 shows that the input port of the 8255 can be accessed easily, so this will not be discussed in depth. The output port, however, involves the demultiplexing of the LS154 multiplexer and it is for this purpose that the hardware has been developed.

The approach that I took was to use eight bistable latches with the lower eight outputs of the LS154 (bit 0 to bit 7) going to each respective A input of the eight latches and bit 9 going to all of the B inputs.

In this way, the numbers of the bits to be set (0-7) are POKEd to location 57344, and may be sent in any order.

For example, to set the number 60 (binary 00111100) on the output port, the following commands are issued:

```
FOR I=2 TO 5:POKE 57344,I:NEXT I
```

Bit 8 is used to indicate to the receiving device that data on the latch outputs is ready, and is therefore the last bit to be sent.

The latch is cleared by POKE 57344,9, and data is read from the input port with PEEK(57345).

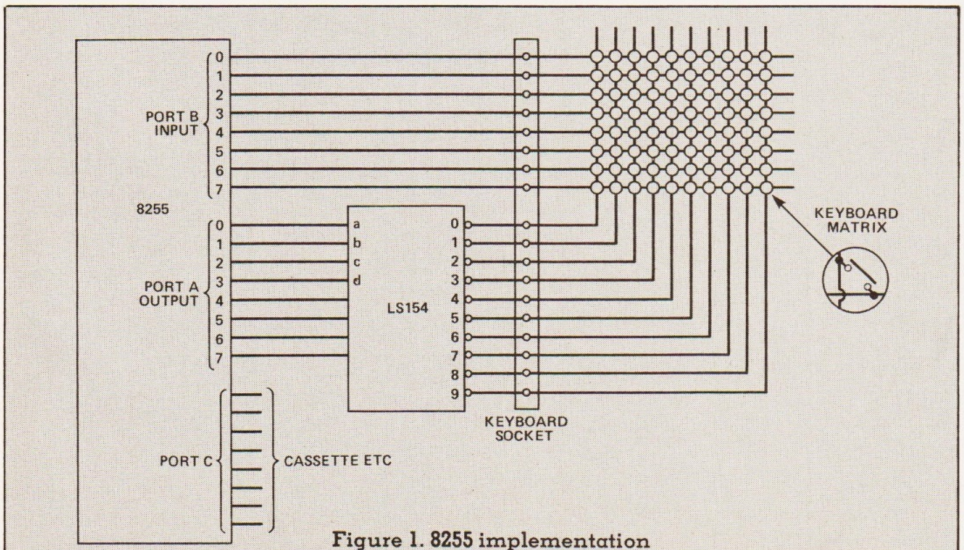


Figure 1. 8255 implementation

FITTING THE SOCKET

If the suggested PCB socket is used, it is first of all necessary to remove the mounting lug on the end nearest to pin 20 to allow the socket to fit in by an integrated circuit. The other lug need not be removed.

The next stage is to remove the four PCB plugs from the main PCB and the six screws which hold the board in place. The board can now be removed completely **taking care not to touch any unvarnished tracks.**

The set of holes into which the socket will fit must be located (see Fig. 2) and unblocked of varnish using a piece of bare wire. The socket can then be inserted with pin 1 leftmost and soldered ***carefully.***

A piece of eighteen-way ribbon able may be used to connect the PCB plug to the I/O unit and wires are not connected to pins 18 and 20.

I/O HARDWARE

The board may be etched or constructed on Veroboard, but no construction details are provided because of the simplicity of the circuit and the wide variation of connectors available.

Fig. 3 shows the circuit diagram of the unit and ribbon-cable connection details.

POWER SUPPLY

A power supply is necessary for both this unit and any device fitted to it. This is taken from the 0V and 5V pins on the PSU connector on the mother-board by means of a jack plug (or similar two-way connector) and a suitable length of two-strand wire.

First of all, the power supply socket which attaches to the mother-board must be removed and the 0V and 5V pins located. The wires are either soldered to suitable pin terminals and placed on the appropriate pin, or are simply wrapped around the pins two or three times to assure a good contact. The socket can then be replaced firmly on the plug.

The next step is to solder a "line" type 3.5mm jack socket to the other end of the wire about 20cm from the power supply connector, making sure that the +ve lead is innermost.

When finished, the socket should rest in the unused front right-hand corner of the computer.

The power supply leads for the I/O unit are simply plugged into the jack socket when needed to avoid unnecessary lengths of wire in the computer.

If the other two voltages (−5V and 12V) are required, a different type of connector is necessary.

Care must be taken not to exceed the current rating on each supply line.

If a jack plug and socket are used, IT IS

IMPORTANT THAT THE PLUG IS NOT REMOVED WHILE THE MACHINE IS TURNED ON. It cost me a fuse and quite a lot of time replacing it because of this mistake.

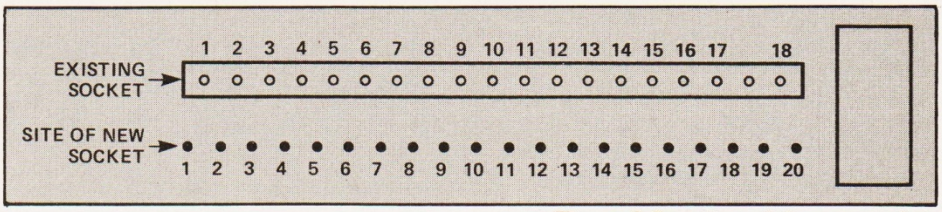


Figure 2. Location of the new socket

USING THE I/O DEVICE

The following BASIC subroutines demonstrate how to use the input and output ports. The first subroutine sets the number contained in the variable A on the latch. The second returns the value at the input port in A.

```
100  REM OUTPUT ROUTINE
110  IF (A<0)+(A>255) THEN RETURN
120  POKE 57344,9:REM CLEAR THE
     LATCHES
130  FOR X=7 TO 0 STEP -1
140  IF A<2 X THEN 160
150  A=A-2 X:POKE 57344,X
160  NEXT X:RETURN
200  REM INPUT ROUTINE
210  A=PEEK(57345) : RETURN
```

Because the keyboard operates in parallel to the two ports, an erroneous number may be reported when a number is input if a key is pressed on the keyboard.

It is possible to fit a Kempston type joystick to the input port for use with home made games.

COMPONENT LIST

IC1-4	74LS00 quad NAND gates
PLG1&SKT1	20-way PCB plug and socket (RS no. 488-365)
PLG2&SKT2	plug and socket for external connection
PLG3&SKT3	3.5mm jack plug and matching "line" type socket
18-way ribbon cable	
Cconnecting wire	

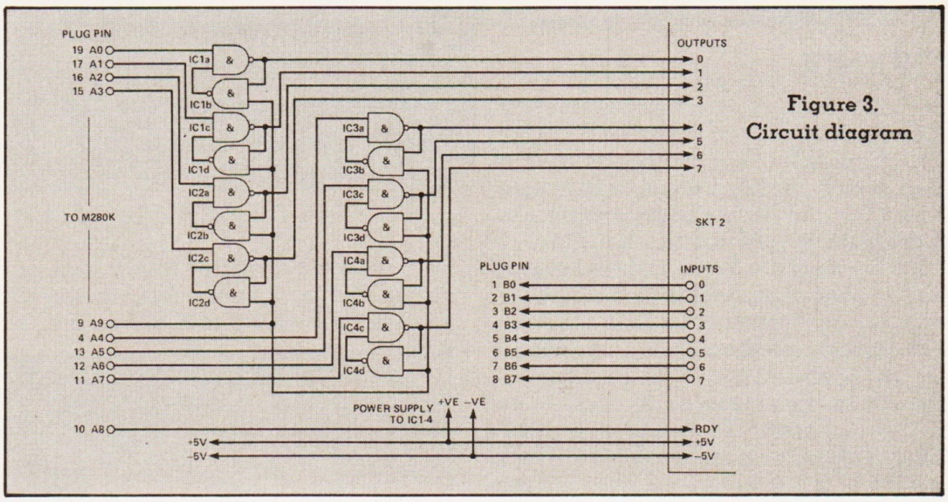


Figure 3. Circuit diagram

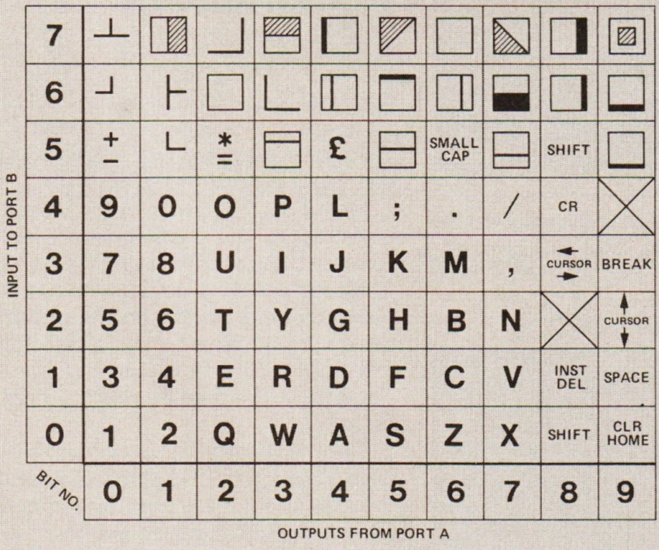


Figure 4. Arrangement of keys in the keyboard matrix

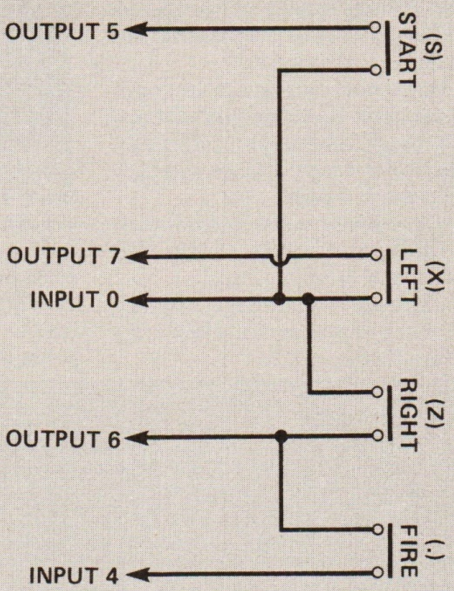


Figure 5. Method of wiring the keys for space invaders

Another application of the parallel keyboard socket is the addition of a smaller keyboard on which the keys duplicate the most commonly used games-playing keys. In this way, the main keyboard is spared from heavy usage.

Fig. 4. shows the details of the keyboard matrix and Fig. 5. shows how an external keyboard consisting of four keys would be wired for playing Space Invaders.

ƎRRATA

TWO PASCALS: we would like to apologise to HiSoft for misquotint the price for HiSoft Pascal, and giving the wrong address. HiSoft Pascal costs £39.95 including VAT, and is available from:

HiSoft,
180 High Street,
North Dunstable,
Bedfordshire LU6 1AT
Tel. (0582) 696421

Similarly, Grey Matter's Pascal costs £49.00 + VAT, and not £45.00 as quoted in the article.

THE MALTRON KEYBOARD is brainchild of Stephen, and not Brian, Hobday as described in our PRODUCTS NEWS IN BRIEF section.

ALGORITHM ANGLES: the up-arrow symbol in the topmost expression of column three would have been the < 'less-than' symbol.

LEARN C: there were a few minor yet significant typographical errors in the first part of this series. In the program that prints **"It works."**, there should be a backslash character before the n. \n is used ot feed the output to a new line.

Also, in the function **sqr,** the argument declaration was misprinted as 'h' rather than 'n', and the comment should have read:

/* function to return n↑2 */

The erroneous comment should then read:

/* /* function to return n↑2 */ */

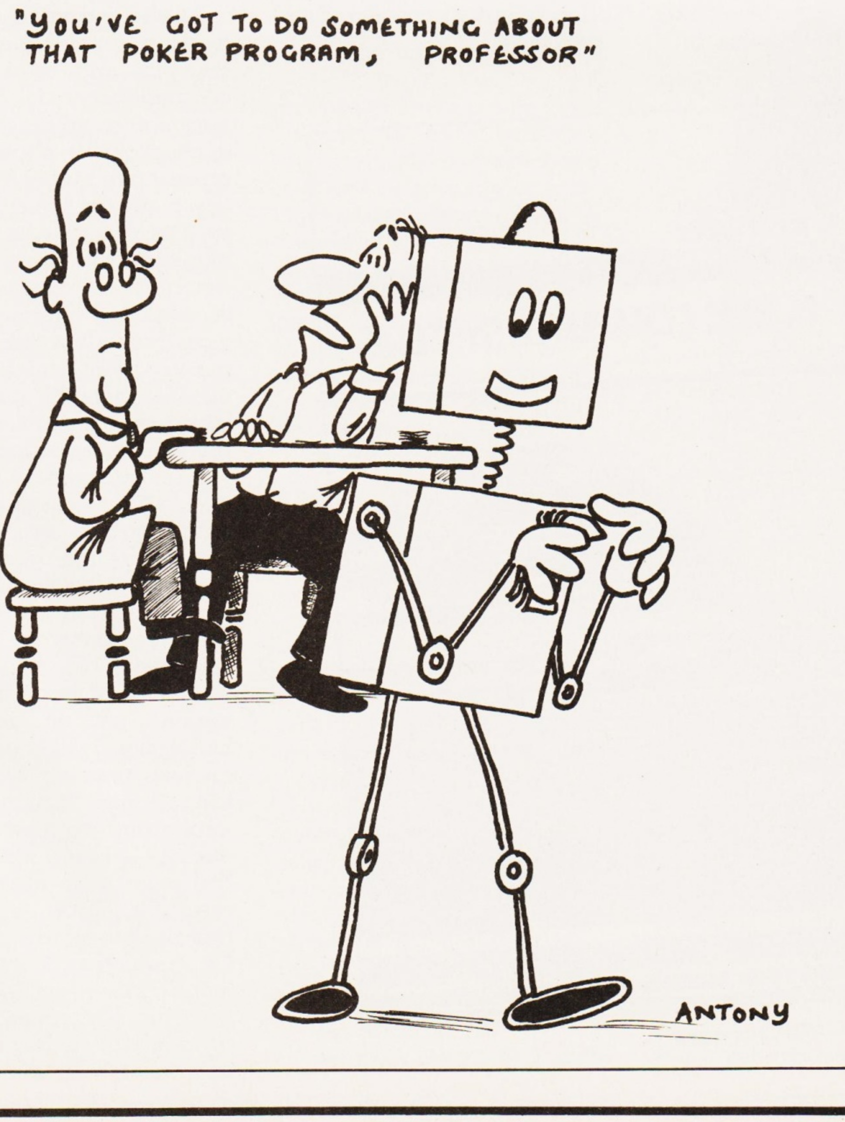
In the main function within **squares** there should be a n at the end of the first **printf,** and the output is assigned the value of **sqr** by an equals sign, not a colon.

To right-justify numbers in **printf** the number embedded in the format specifier has a minus sing, and so the example should have read:

eg. &−4d

INSIDE BUBBLE MEMORIES: although our language has a certain 'built-in' flexibility the word 'propagation' should be spelt as such, and not 'propogation'. Potential contributors are advised that this is the preferred spelling.

We wish to apologise to readers who were affected by these errors.



FIVE YEARS BACK

In these days, when you can buy a ready-built computer for £100-£200, it is strange to realise that a mere five years ago people were paying-out appreciably larger sums for kits of parts, and reckoning that they saved money by doing so.

Computing Today for May 1980 carried an owners' review of the Triton, launched by Electronics Today International in November 1978 and marketed by Transam. The system began in a modest way, with a seven-function monitor and 'Tiny' integral BASIC held in three 1K by 8 EPROMs, but expansions soon followed, such as an 8K RAM extension to the original 2½K. Oddly enough, it seems that the extensions were announced in *Computing Today,* who clearly saw a band-wagon on the move. By 1980, disc drives and built-in Pascal were available — and the price had risen to £611!

This was not the only kit computer of its day by any means. There was the Comp 80 — published in Wireless World and marketed by Powertran — and the Nascom, plus the newly arrived ZX80, and the UK101. Nowadays, anyone building his own hardware is regarded as a little odd, though quite a number of home-brew merchants are still active. Unfortunately, they tend to isolate themselves by writing 'unique' operating systems . . .

Another change during the last five years is the evaporation of interest in programmable calculators. Our May 1980 issue covered the 'Stockmarket' game for the T159 calculator, though the program was expressed in flowchart form show that it could be written for other devices (unfortunately, a subsequent version in BASIC threw away some of the more interesting features of the original!)

Perhaps we should try the same approach — but flowcharts are not everyone's cup of tea . . . □

BOOK PAGE

Garry Marshall

Three quite varied books, from communications to education.

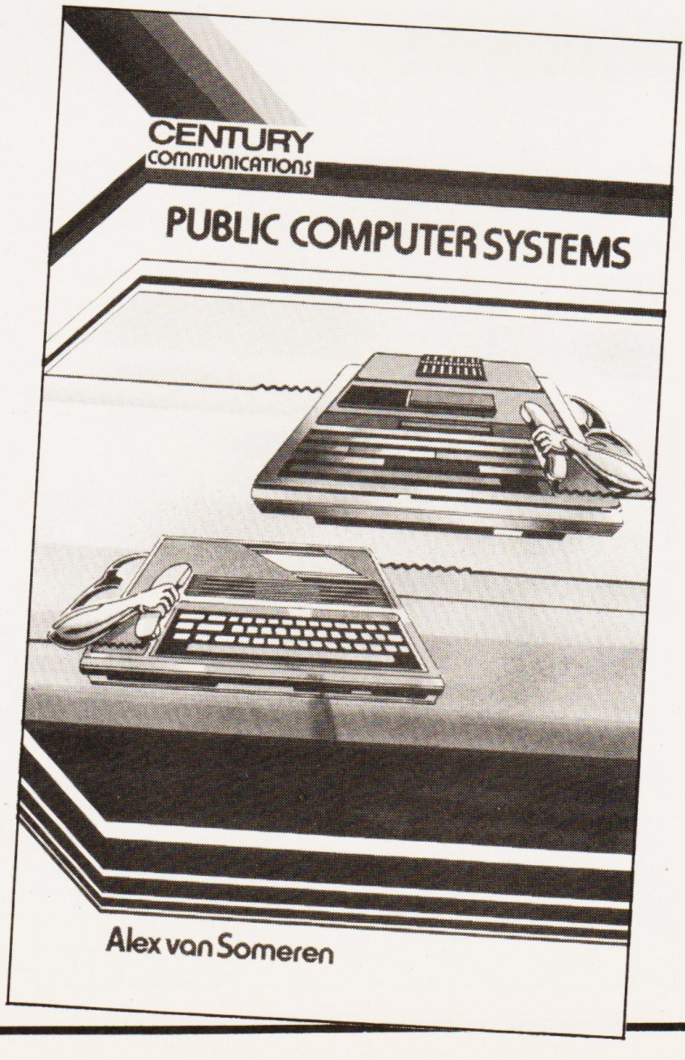
One of the best things to happen to personal computers is communications. It has been possible for computers to communicate either with terminals or with each other if not from their beginning, at least from soon after it. Now it is becoming increasingly easy to use a home computer as a communications terminal and, additionally, there are many good reasons for doing so.

In terms of the technical issues, it is not difficult for a computer to communicate. It requires little more than to lead the electronic signals flowing within a computer outside it and to conduct them to another computer. It is not quite as simple as this, of course. Because no two types of computer work in the same way, some conversions are necessary but, as this is the kind of thing than can be done by a computer anyway, it is not an insuperable problem.

The benefits that result from enabling computers to communicate with each other are considerable and they follow from the fact that the linked computers can share their resources with each other. When a network of linked computers has been established, the user of one of the computers can make use of the facilities of any computer in the network. Beyond this, he can also communicate with anyone using one of the other computers to discuss problems and share ideas with him. The benefits that can follow from this may not be altogether apparent at first sight. A number of computers linked together by good communications and with the necessary communications software can become, in effect, one very large computer. By communicating and interacting with each other, the users of the network can produce co-operative results that can exceed anything they could accomplish individually.

Today, the operations of banks, airlines, police forces and, indeed, government depend heavily on the use of interlinked computers and would be practically impossible without them.

Personal computers have had the ability to communicate from the first, and a few of their owners discovered this at a very early stage. A fair amount of electronic expertise, not to mention some contempt for the law, was necessary to construct and use the equipment needed to connect them to the telephone network, but once this was done the computer could communicate with any other that was attached to the same network. We may only hear about the activities of the hackers who illicitly break into the computers of large institutions, but there has been a good deal of innocent activity going on for some time.



One of the most exciting developments in the personal computer field recently is the development of products that allow a home computer to be easily connected to the telephone network. This means that we can all join in the fun. In addition, information banks and information services that are worth communicating with are also becoming available, so that besides having fun we can actually benefit at the same time.

Public computer systems by Alex van Someren aims to give those with no experience an introduction to using the personal computer for communications. It is intended to present a readable account of how to use the home computer for accessing and communicating information, and to describe the communication services and information services that are available for this purpose. Well, while it presents all the material needed to introduce communications to the newcomer, readable it isn't. The author seems to know his stuff, but the services of a copy editor and a proof reader would have improved his book immensely. It contains several sentences, and at least one complete paragraph, that are complete gobbledegook.

We are first introduced to the jargon of on-line communications. Like all closed interests manned by enthusiasts, it has its unnecessary jargon and we must pay our dues by learning it before we can enter their world. There are CIGs, CUGs, sysops and userids. It is all reminiscent of CB, which on-line communications resembles in other ways, too. The real hackers will certainly have nothing but contempt for those of us who would make our entry with a modem bought off the shelf that only needs to be plugged in at the right points in order to make everything happen.

We are then told how to go about getting our micros to communicate, and given an account of how the communication signals are transmitted and of the networks over which they can be transmitted. The services described are the telephone network and PSS (Packet Switch Stream). After this the services that can be accessed are dealt with. Those covered include Prestel, as an example of a Viewdata system, Telecom Gold, as an example of an office system, and, in general terms, free public systems.

Prestel is probably familiar as the information service that can be accessed by linking a television set and the telephone. It provides information which, if it is likely to be of any use to you, must usually be paid for. It also provides message exchange and teleshopping facilities. But its most interesting aspect is Micronet, which is CUG (closede user group) within Prestel created to allow those interested in micros to exchange and share information with each other. Besides giving us an example of the use of the jargon, it has given Prestel a much-needed shot in the arm, and may well indicate one general direction that public information utilities must take to be successful.

Telecom Gold, in common with any office system, offers the facilities that are likely to be needed in the office to its users. Word processing and database programs can be accessed, an electronic diary facility is provided and electronic mail can be exchanged between subscribers. Other computing facilities are provided, including further applications programs and access to computer languages. In general, an office system helps to bring the dream of the paperless office rather loser to reality.

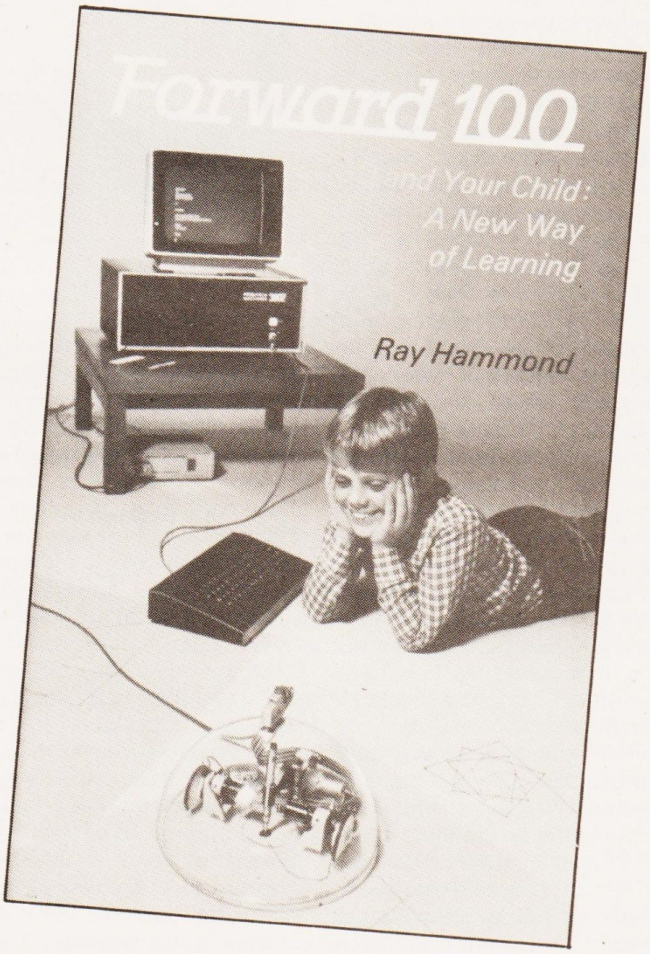
The more-or-less free public systems include bulletin boards, which are usually divided into SIGs (special interest groups) and operated by sysops (system operators). As with any kind of system, you must provide your userid (user identification) before you are allowed into the system. A bulletin board is the electronic equivalent of a notice board where you can leave and read messages. This sort of communication with faceless folk is reminiscent of CB. Besides bulletin boards there are sales boards which, discouragingly, you sometimes have to pay to use.

This outline does give a sketch of the services that are available and of what on-line communication has to offer. It doesn't really indicate how easy it is to access similar services in other countries, in America mainly, so that it does not really lead into the book's final chapter on the future. This presents the ideal situation where you would be able to put your information needs or requests to your computer which could then decide which service to access, do so, and thereby meet your every need. This aspect of the 'global village' is attractive and not so difficult to imagine, either.

So, if you are interested in finding out about using the computer for communication and are not too worried about writing style, 'Public computer systems' may be what you need. The book has omissions which, given its shortness, is not surprising. There is little mention of the communications software that is needed to make a micro behave as a communications terminal. The costs involved, not just in getting started but also in accessing the various services, are not mentioned. The latter can mount up rapidly, so that the newcomer ought to at least be aware of this. Also, the selection of services that is mentioned does not really do justice to the variety and richness of the range of services that is available. But the book does contain enough to get us started, and for that we should be grateful.

Moving on to other matters, it seems to me that a groundswell of opinion is forming for the idea that a personal computer must be useful; that it is too expensive to be used just as a games machine. It is none too easy to make the home computer into a useful tool (at a reasonable price, anyway) even when the tasks that you want it to perform are information-related or information-rich. On the other hand, the youngsters (and the not-so-young) who do use their micros for playing games may well feel that they are getting full value from them. Somewhere there ought to be a point of balance between serious and frivolous use.

This brings to mind a recent speech on education by Enoch Powell in which, essentially, he was attacking the policy on education of the present government. The government is pressing for more education that provides specific job-related training (let us call this 'serious' to try and establish a parallel). Mr Powell was arguing that it is the philanthropic duty of a civilised society to allow its students to study the subject of their choice regardless of its utility (in the present terms, to allow them to make a frivolous choice).



Now, in my view, Mr Powell is undoubtedly right on this matter: it is even possible that some of those believing in the Thatcherite realities may agree with him. But, having had their chosen education, students must surely be prepared to join the job market with an open mind about working in the wealth-creating sector and with no unrealistic allegiances to their chosen subjects. Fortunately, those who choose to study computer-based subjects seem to have the best of both worlds, because not only do they study their preferred subject, but they can also be assured of a job that requires the exercise of their chosen education, even a choice of such jobs, at the end of their studies.

By this circuitous and opinionated route, we come to our next book, which is on education. It is **Forward 100** by Ray Hammond, and it is about the impact that he thinks computers, and more specifically Logo, will have on the education of our children in the future.

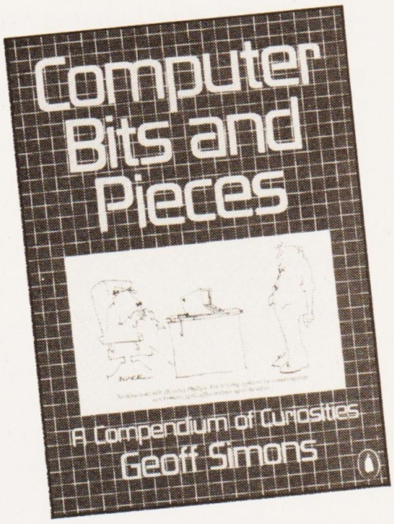
Now, Ray Hammond is a prolific author, and he writes very well. I have read and enjoyed several of his books, among them a recently published and very readable account of on-line communications. In fact, he has used the ability to consult databases all over the world to gather much of the information for this book. And, I am sad to say, that accounts in some measure for what I think is wrong with it.

My other objections to it are encapsulated in these two sentences from page 29. "Many teachers in both primary and secondary schools are time servers, lured into the career by the 'soft option' of teacher training and held ever afterwards by the seductive holidays and short hours. Ostensibly, preparation for lessons is supposed to demand many hours a week, but several surveys have shown that the vast majority of teachers prepare nothing outside of their nine-to-four timetable." There are skivers in every line and I don't think that teaching is an exception, but to accuse 'many' teachers is unforgiveable. The arguments by which the teachers justify the hours that they work are well-rehearsed and need not be repeated here. You would hardly expect primary teachers to spend hours in preparing material, although they might spend a great deal of time in thinking about how to get ideas and facts across to particular classes and partiular individuals. Such activity is clearly work, even if it doesn't look like it while it is being done.

Further, how does such a remark square with this on page 24: "...any discussion of computers in education must start at the point where the teacher works — in the classroom." Of course it must. And large numbers of teachers are not just talking about it, they are actively using computers for sound educational purposes in the classroom. If Hammond thinks the teachers are skiving, then how does he imagine that the computer will be introduced to help with learning in the school? Perhaps the computer will allow education to be taken away from the schools altogether? Well, we will come back to that in a minute.

Why such a fuss? It's because I think that the role of the computer in future education *is* important. Too important to be treated in an ill-considered fashion in a book that is sold as a 'major discussion and debate about how computers can best be used to stimulate children.' Worse than this, the contents of the book seem to me ill-digested and unshaped by any overall vision. It is all very well to have access to large amounts of relevant information, but to use that information in the creation of a worthwhile book requires the author to sort, select and mould it into a coherent whole.

The book is actually about the opportunities for learning in new ways that are made possible by Logo. Now, Logo is a great vehicle for allowing children to learn in much the same way that they learn from play. This view is well illustrated by the remarks of one teacher quoted in the book.



"One benefit of having Logo in the classroom is that it gives students an opportunity to share ideas easily. They are excited about the procedures their classmates have written and they eagerly explain their procedures to one another". I am sure many teachers will recognise both the individual stimulation and the desire to communicate the resulting excitement that are described here. They will recognise that the computer can cause them in ways other than with Logo. They will also know that they can themselves cause the same effects in their pupils without recourse to computers at all.

But the problem with learning from play is that it is undirected. If children could learn all they needed just by playing, there would be no need for schools. But direction is necessary, and someone must be there to give it, as well as to answer all the questions that children may ask and which are a fundamental part of their learning experience. Certainly children can learn things from Logo, and some of them are illustrated in the book. But if all the learning is achieved by exploring ideas, what happens when the ideas run out? Where does the child turn then?

The 'microworld' of Logo provides one way of directing the learning process. It consists in essence of Logo plus specific knowledge about some area. The microworld can then be explored, and learning about the world in question can take place in the way that all learning does with Logo. The difficulty is that children cannot create such microworlds for themselves; someone must do it for them. Who is to do it? Will it be teachers, parents or educational consultants? Or will the money become available to buy in the software for microworlds in the same way as individual educational programs can be purchased now?

If Logo could be used to educate children successfully, could the place of education be changed from the school to the home? The single-minded concentration that can be displayed by adolescents when playing video games could then be applied to their education. It might not be too healthy after a while, though. And what about the interactions with other children, and real play? Will there be anyone to answer the questions?

I do accept that schools will have to adapt to the ways that computers can be used for education. But I think that a number are already doing so. Also, I think that it is most unfortunate to fail to recognise that a great many schools do provide their children with a good education, do stimulate them, and do ensure that they learn how to learn. It is not yet clear that computers can provide education that in any sense improves on that given in school. It is not even clear that, with Logo, computers can educate effectively in ways that schools cannot. So the idea of the computer as the sole, or even a major, tool for learning is one that will not be fully realised for some time yet.

In the mean time, Hammond has some suggestions on how to go about making Logo one component of a child's education now. Would you believe that you should find a school that already makes use of Logo on its computers or that, failing this, you should convert one of the teachers in the school to the Logo cause!

I imagine that it is clear that this book has raised very strong feelings in me: perhaps that is a recommendation for it. It is certainly about a very important subject. I do think, though, that it would be better to go back to the sources that the book draws on so heavily, not least to the writings of Seymour Papert in 'Mindstorms' and in subsequent publications.

This month's third and final book is **Computer bits and pieces** by Geoff Simons. Its sub-title is 'A compendium of curiosities', and the title and sub-title together describe the book quite precisely. It is a collection of short sections, each of which gives a description of a computer application in a paragraph of about 15 lines. They are the kind of thing that you might find as a short news item in the pages of ***New Scientist,*** although a few are more reminiscent of ***Reader's Digest.*** A few section headings are 'Robots for domestic chores', 'Chess computers play', 'How to sort rice' and 'Expert medical diagnosis'. No small selection like this can possibly catch the range of applications that is touched on; in its own way the book does encapsulate the amazing variety of tasks in which computers are, and will be, applied. With this on your bedside table to dip into you should be in a position to start any number of interesting conversations at parties, and to stop quite a lot, too.

This month's books are:

Public computer systems by Alex van Someren (Century Communications) 80 pages, £4.95

Forward 100 by Ray Hammond (Viking) 304 pages, £12.95

Computer bits and pieces by Geoff Simons (Penguin) 190 pages, £3.95

ALGORITHM ANGLES

Bill Horne

More algorithms as meat for your micro.

In a recent review, Peter Freebrey expressed regret that the manual for a certain computer gave hexadecimal numbers instead of decimal values, and a reader has since expressed disapproval of this sentiment. There is something to be said for both points of view, but those who cling to decimal notation too tenaciously may lose out to some extent. It is sometimes worth using the less familiar forms, such as octal and hexadecimal, since they are sometimes actually easier to use.

A number is not affected by the notation in which it is expressed. The only change is in the way the number is written or printed. In the most general terms, a number X can be expressed;

$$A + B*N + C*N^2 + D*N^3 \ldots$$

where A,B,C,D are the digits of the number as written, in reverse order, and N is the base of the notation used. In decimal, the number 1234 has a value;

```
4*1        4
3*10      30
2*100    200
1*1000  1000
        1234
```

which seems almost too obvious to be worth stating, but if N=2;

```
4*1    4
3*2    6
2*4    8
1*8    8
      26
```

However, we are breaking an unwritten rule here, because no digit may be equal to or greater than the number base, so if N=2 the only digits permissible are 0 and 1.

The number 1234 is permissible in hexadecimal, when it gives;

```
4*1        4
3*16      48
2*256    512
1*4096  4096
        4660
```

We have no ordinary numbers to represent the digit values 10 to 15, which are permissible in hexadecimal, so we use letters A to F. This is more of a complication than it sounds, because there is a gap between 9 and A in the ASCII code table.

To convert a number to digital form, the process is;

a. Divide the number by N in integer fashion, result X
b. Subtract N*X from the number. The result is the right hand digit.
c. Make the number equal to X, and repeat the above to obtain the rest of the digits

The functions DIV and MOD are useful here, because they allow the process to be simplified;

```
100 X=Number DIV N
110 Y=Number MOD N
120 PRINT Y
130 Number = X
140 IF X<>0 GOTO 100
```

Without MOD and DIV, it is necessary to use;

```
100 X=INT(Number/N)
110 Y=Number – N*X
120 ...
```

But this is no good. We get the right hand digit first. We need to set up an array to hold the Y values, and we can then output them in reverse.

We still have a problem if N is greater than 10, that of converting the higher digit values to letters. It can be done like this;

```
200 Y=Y+48
210 IF Y>57 THEN Y=Y+7
220 PRINT CHR$(Y)
```

Adding 48 gives the ASCII numeric code, and 57 is the code for nine. If we get 58, say, we add 7 to give 65, the code for A.

This will work for any value of N, as is demonstrated by some FORTH implementations, but too large a value will bring in codes beyond the alphabet, which can be disconcerting.

The reverse process, setting up a number during input, is simpler, because no digit reversal is needed. First, a variable is zeroed. Call it Z. The first digit is read, and 48 is subtracted. If this gives more than nine, a further 7 is subtracted. Then Z is multiplied by N, and the modified input digit is added. This is repeated for each digit in turn, starting with the most significant. The final result is in Z.

The value of hexadecimal is greatest when it is used to make the interpretation of binary numbers simpler. Each hexadecimal digit represents four binary bits, and whereas a long binary number can confuse even an expert, the grouping in hexadecimal makes the task simpler.

There is an interesting parallel to this. Twenty-five years ago, explicit decimal displays were only just beginning to appear, and a popular arrangement used four binary bits to represent each decimal digit. This 'binary-coded-decimal' system was easier to read at a glance than a string of individual binary bits. If the string had been divided into groups of four, without regard to decimal notation, the task of interpretation would have been simpler, and hexadecimal does just that.

Hexadecimal is particularly useful for expressing store addresses. The number 49152 may not mean a lot as a store address, but in hexadecimal form it becomes C000, and is more obviously the three quarter point in the address range. It could also be suggested that crucial addresses are more easily remembered in hexadecimal form.

Before hexadecimal came along, 'octal' notation was popular. Each digit represented three binary bits, the value of N being 8. Though still used (e.g. in the data for the Programmable Sound Generator) it has rather faded away, because it fits less well with the standard byte and word lengths, which are usually a multiple of four.

Objections to octal and hexadecimal notation are largely emotional. There are obviously contexts in which straight decimal is to be preferred. After much heart-searching, we at last have a decimal coinage, and no longer have to perform conversions from pounds, shillings and pence. To apply anything but decimal to money matters would clearly be wrong, but where binary data is concerned hexadecimal can be a great help.

All that is needed is a little practice, and you will find that hexadecimal can convey much more to you than decimal, given the right context.

BACKNUMBERS

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

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

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

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

MAY 1984
Debugging, Spectravideo SV318 review, Extending the Commodore 64's BASIC part 1, Z80 text compactor.

JUNE 1984
Adler Alphatronic review, Digithurst's Microsight review, Commodore search and replace, CP/M directory, Interrupts.

JULY 1984
Commodore BASIC extensions reviewed, The Art of Islam, a fast sort, Brother HR5 review, Random Thoughts, extended palette on the Dragon.

AUGUST 1984
Apricot xi review, BBC Mode 7 screen editor, Genie sprites, Microdrive-file line editor, TRS-80 screen scroller.

SEPTEMBER 1984
CUBE's Beebflex, Electron drawing utility, MTX real time clock, Commodore SX64 review, BBC disassembler, TRS-80 Fastsave.

OCTOBER 1984
AMSTRAD CPC464 review, Dragon sprites, Commodore 64 adventures, BBC Draughts, Nascom screen dump.

NOVEMBER 1984
Apple IIc review, Epson PX8 review, MTX utilities, Z80/TRS-80 memory move routine, 16-page Business supplement.

DECEMBER 1984
Acorn Bitstick package review, Art and the AMSTRAD, BBC Draw, Psion Organiser review, Koala Pad review.

JANUARY 1985
BBC Commodities, Tatung Einstein review, Fujitsu Micro 16 review, Commodore 64 prettyprint, MTX500 Life, Nascom string-save.

FEBRUARY 1985
The Intelligent Computer, Dragon interrupts, BBC Machine-code monitor, Tasword 464 review, Spectrum/BBC cassette volume meter, Sakata SCP800 printer/plotter review, Spectrum ON ERROR, TRS-80 mail list, BBC passwords; Deficiency, Abundance, Perfection.

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

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

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

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

A COMPUTING Today

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £ ....

NAME ....

ADDRESS ....

....

....

.... POSTCODE ....

Signature ....

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

Access ☐ Barclaycard ☐

Buy it with Access — BARCLAYCARD VISA

Insert Card No.

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

Please allow 21 days for delivery.

PRINTING BY AMSTRAD

Bill Horne

Sales of Amstrad printer cables are greater than sales for the printer itself. Surprising? Maybe. Remarkable? No. Read why...

One of the things that a newcomer to the microcomputer world soon learns is that a printer is almost essential for anything but the simplest use of the system. He also learns that this can double the cost of his equipment if he sets his sights too high. Amstrad have offered a solution to this dilemma by offering a printer of moderate performance at a price below average, and the device will be examined in that light.

FIRST IMPRESSIONS

It must be said at once that the printer is entirely practical for typical microcomputer work. It has a Centronics-type interface, so it will work with a number of popular computers; It provides for 80-column printing on standard continuous paper stock with tractor perforations; it produces a clear print with only minor reservations.

Evidence of cost-saving shows first in the fact that there are no obvious 'front panel' controls, no on/off line select, no manual line feed or form feed. There is a mains switch on the left-hand side and a set of tiny mode-select switches at the rear, and a red light shows when power is on. All other control functions must be exercised by the controlling computer.

INSIDE

Looking inside, we find that the print head is only motor-driven towards the right, return to the left margin is by a coil spring. The time-saving device of bidirectional printing is not implemented here, which will not be noticeable to the average user. A print speed of 50 characters/second keeps action going nicely, even with tediously long printouts.

The next thing we discover is that there are only seven data lines, pin 9, the usual connection for Data 8, being linked with the ominous warning 'Do Not Use'. Now this, with respect, is a bit naughty. It is true that the Amstrad CPC464 only outputs seven data bits to the printer, but that is a comparatively rare phenomenon. Most small computers output all eight bits, and it would appear that this might lead to trouble, unless the connection to pin 9 is broken.

Alerted by this, we study the remaining connections, and discover that pin 14 is specified as an earth. It is, in fact, earthed by the CPC464, forcing other makes of printer to execute double line feeds in response to the usual CR/LF sequence. Amstrad contend that the earthing of this pin is in accordance with the Centronics standard, but it would appear that they are not supported by other printer manufacturers.

Other pin allocations appear to match the norm, though one is a little unusual: Self-test printing is initiated by earthing pin 35. Since the CPC464 makes no connection to this line, self-test presumably requires a special connector. (On an Epson printer this line is pulled high by a 4k7 resistor to +5V).

The data lines are matched by the print head, in that there are only seven wires or hammers. At least, there appear to be seven hammers, in terms of the result, but there appear to be only three wires leading to the print-head, and that creates a mystery which could only be solved by a strip-down, which we were reluctant to undertake. There is no sign of vertical movement of the print-head or paper, which puts paid to one suggested answer...

There are, nevertheless, seven vertical dot positions, and the character matrix is clearly seven dots high and five dots wide. This, unfortunately, means no 'descenders', one consequence being that 'p' and 'P' look almost identical, which can cause mayhem with typesetting...

Against this must be set the availability of eight language variations, selected by three of the four miniature switches at the rear of the machine, the variants being U.S.A., U.K., Germany, Sweden, France, Denmark, Italy and Spain.

IN USE

Two type-face standards are available, single and double width. There are no double-strike or reduced width modes. However, there is a graphics dot mode available, in which input data produces a vertical row of seven dots. In this mode, line feeds are reduced to make the graphics lines vertically adjacent.

When the semi-transparent lid is closed, the noise of printing is not excessive, but the rattle with the lid open is fiendish.

In use, which has largely consisted of printing a disassembly of the CPC ROMs and the listing of various programs, the DMP1 has performed satisfactorily. Paradoxically, the printing was a little faint at first, but has become blacker as the stack of paper diminished in height by about 1'', and is now clearly legible.

The biggest annoyance has arisen from the difficulty of aligning the folds of continuous stationery between print lines. No guidance on this has been traced in the operating manual, which is otherwise quite comprehensive. There is a stubby pointer above the print head which appears to be intended to provide an alignment point for the fold, but this does not always work. In consequence, a 'paging' system has had to be developed which skips the fold area completely, but that does not work with listings, which have to be done a page at a time...

The weakest feature of the printer is the ribbon cartridge. This uses a comparatively small ribbon loop, which is inked from a reservoir as it is drawn into the cartridge. It has been found that the ink is liable to dry out to some extent in storage, even when it is kept within its protective plastic envelope.

A worse fault encountered was a ribbon jam, which persisted when the cartridge was removed and an attempt was made to advance the ribbon by turning the small knob. Pulling the ribbon out completely and winding it in again provided a temporary cure, but when this became necessary three times a page in a 150-page printout a new cartridge had to be purchased — at £6.50 plus VAT.

In theory, it should be possible to refill the reservoir, but that would be a messy business. It is said that replacement reservoirs should be available, but if anyone knows of a source, a lot of people would like to share the knowledge...

CONCLUSION

It should be said that the DMP-1 is now sharing the workload with a more expensive printer which has been in regular use for more than four years, and it is inevitable that comparisons exist in the mind of anyone who uses both. The more expensive printer is obviously better in performance, as befits its price, but that does not mean that the DMP-1 is inadequate. At £169, it is certainly cheap, and its performance nevertheless makes its users quite cheerful.

LEARN 'C'

Mark Woodley

Part two: expressions, strings, and other things.

All programs have two basic constituents: Data and Instructions. The data is any information encoded in the computer's memory and the instructions are inevitably machine op-codes. So far we have seen the simple data structures that C uses to form its types. As we will now see in the ensuing discussion of C instructions (or **statements**), they are important to understand at an early stage so that the programmer can take advantage of 'loose' type restrictions.

ASSIGNMENT STATEMENTS

Unlike other languages whose mathematical and logical expressions must be written in terms of the same data type, the type of the components of a C expression can be mixed.

For example, a programmer may assign to the variable **result** the value of the expression **a + 1**, with:

```
result = a + 1
```

If **a** was a variable declared to be type **char** then this tells the compiler to compute the value of the **char**acter code in **a** plus the **int**eger **1**. When, as in this case, the types of an expression are mixed, the C compiler looks for the 'widest' type in the expression (i.e. the type that requires the most memory for its representation) and puts the result in that form. This ensures that minimum accuracy is lost in calculating the result.

C types follow a natural ':widening hierarchy', which is **char, short, unsigned, int, long, float** and **double**. So if **a** was of type **float, b** was of type **integer** and **c** was of type **char,** then the value of the expression **(a + b + c)** would be represented as type **float,** irrespective of the type of variable that it was being assigned to.

The consequenes of such actions can only be understood by the programmer if he has an in-depth knowledge of the way the types are represented in the computer's memory. As representations vary from machine to machine, it would be wise to avoid such eventualities by ensuring that the variable being assigned is always wide enough to hold the value that an expression gives.

To help the programmer, C has a **cast** operator which allows the type of an expression's result to be defined. To cast to the required type, the expression is preceded by a type-identifier, in brackets. For example, to case the expression **a + b** to render a **float** result, we would use:

```
result = (float) a+b;
```

Assignments, are all of the form:

```
variable = expression
(lvalue)    (rvalue)
```

and are called assignment statements, because they are a way of stating what the program should do. The terms **lvalue** and **rvalue** are often used in compiler diagnostics.

STRUCTURED STATEMENTS

Some of the C's statements fall into natural formats or structures and are called structured statements. They have evolved from the theories of structured programming. Each structure has one entry and one exit point.

The conditional statement is used in all languages to change the flow of control, since it allows the program to make decisions. The C syntax is very much the same:

```
if (expression)
   statement1;
else
   statement2;
```

If the result of the expression is 'true' (denoted by a non-zero value), then statement1 is executed, otherwise if the result of the expression is 'false' (denoted by a zero value), then statement2 is executed. This representation of true and false is used throughout C.

Note that unlike other languages, the word 'then' is not reserved in C and holds no meaning to the compiler. The use of 'then' in other languages evolved from the need for the compiler to distinguish where the expression (that represents a condition) ends, and the statement (that requires some action) begins. The use of brackets in C solves the problem and provides a better shorthand. Note also that statement1 and statement2 can be written in any form; an assignment statement, a compound statement, or a structured statement, etc.

When **if** statements are nested (an **if** statement occurs as a statement within another) it can be confusing to see which **else** is associated with which **if.** These ambiguities can be removed by writing each **if** statement as a compound statement (i.e. between braces).

Multiple branches, that depend on some expression or variable, can be implemented with the C **switch** statement:

```
switch (expression)
{
   case label1;
        statement1;
   case label 2;
        statement2;
   case case label 3;
        statement3;
   default:
        statement;
}
```

The first line of the **switch** statement consists of the reserved word **switch** followed by an expression that represents a value. There is then a sequence of statements between braces. Inside the braces numeric labels can be put at any point where execution should begin if the expression equals the lable. Execution will then begin at this point until either the reserved word **break** is encountered or the end of the braces are reached. A **default** label can be used where execution will begin if there is no label associated with the value of the expression. Each of the labels (apart from **default**) are distinguished by the reserved word **case.**

C also provides for loops. An 'iteration' can be performed a known number of times, or it can be executed while an expression is true or until a condition is false.

The **for** structure is used to iterate a predetermined number of times. Its Syntax is confusing to the beginner, as it forms yet another of C's shorthands. Generally, it can be written:

```
for     (expression1;      expression2;
    expression3)
    statement;
```

Any of the expressions can be excluded, but the semicolons are compulsory. The first expression is evaluated first. Its purpose is to initialise some counter variable. The result of the second expression is then calculated before each iteration. The expression will return a non-zero value if the expression is true. Only if the expression is true will the statement be executed. If the statement is executed, the third expression will be evaluated immediately afterwards. This expression is usually an assignment to the counter variable to increment it by the desired amount. As an example, we could perform the function ANY three times with the code:

```
int x;
fopr (x = 1; x<=3; x=x+1)
ANY();
```

Here **x** is the counter variable, and x<=3 is the looping condition.

A similar structure, the **while** loop is offered independently in C. This takes the form:

```
while (expression)
    statement;
```

Its syntax is similar to the **if** structure. However, the expression is evaluated before each execution of the statement while the expression renders a true result. The program then continues after the **while** statement, when the expression yields a false result.

To perform a statement repeatedly **until** an expression is true, we use the **do..while** structure, known to others as the Repeat structure. The test is ***always made after the statement is executed:***

```
do statement
while (expression);
```

OTHER STATEMENTS

It is possible to force your way out of a control structure by jumping to a label within the program. A practise scorned upon by advocates of structured programming.

Somewhere in the program, there is a label and identified as such by its following colon, e.g.:

```
error : statement ;
```

You could now jump to this statement, using **goto,** i.e.:

```
goto error;
```

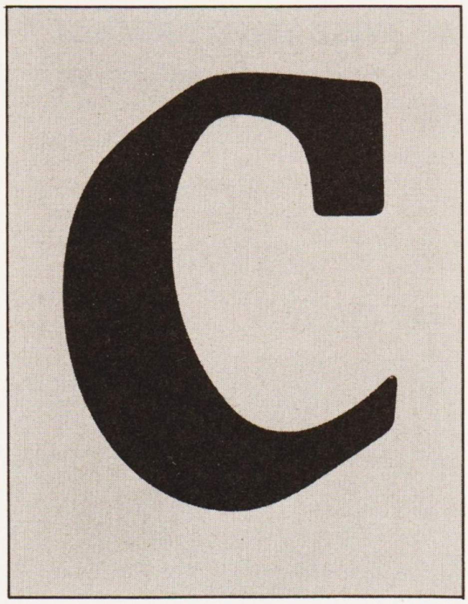
Another useful statement is **continue.** This invokes another ungraceful termination and should thus be avoided. On reaching the **continue** statement from within a control structure, the compiler will create a jump to the end of the control structure, so that normal program flow can resume.

EXPRESSIONS

It is now time to introduce the form of C expressions, expressions are used in assignment statements, and all of the structured statements.

A number on its own forms the most simple expression, a constant expression.

Decimal integers are represented as a digit from 1-9, followed by any number of digits in the range 0-9. Octal integers start



with a 0 and then contain the digits 0-7. Hexadecimal integers start with 0X and then contain the digits 0-9 and the letters A-F.

Single characters are represented between apostrophes, for example, a is written 'a', q is written 'q' etc.

Special characters are also represented between apostrophes, but with the following symbols:

/b	backspace	\0	null
/n	newline	//	/
/r	carriage return	/''	''
/t	tab	/'	'
/r	form feed		

or /and-up-to-three-octal-digits (e.g. /30 cancel)

One example of the use of a constant expression is in providing initial values for variables when they are declared. This is done by writing an assignment into the declaration.

For example, the following declaration **initialises** the character variable **c** to the letter **a**, and the integers **a** and **b** to the numbers 2 and 5.

```
char c='a';
int a=2, b=5;
```

It is recommended to only initialise variables that do not change, so as not to confuse a prospective program reader.

ARITHMETIC EXPRESSIONS

Most of us meet expressions for the first time in algebra. For example:

A * C + B / D − E

is a valid expression using the operators, **+** plus, **−** minus, ***** multiply and / divide.

To avoid any potential ambiguities, each operator in C has a precedence level. Multiplication and division are grouped together, and these have a higher precedence than addition and subtraction, which are also grouped.

So, if the computer was given the expression

A * C + B / D

the multiplications and divisions would be evaluated first as:

(A * C) + (B / D).

and then the addition would be done last.

When all the arithmetic operators have the same precedence level, the association is from left to right. So:

A + B − C + D

would be interpreted as

((A + B) − C) + D

C provides an additional operator to the basic four, which is the modulus operator **%**. It is provided to supply the value of the remainder after a division. For example, **5 % 2** is 1, since 5 / 2 is 2 remainder 1.

A good knowledge of precedence is essential in C because some operators associate from right to left for example, the equals sign.

An expression of the form:

variable1 = variable2 = variable3

would be interpreted as:

(variable = (variable2 = (variable3)))

which would set variable2 to variable3 and then variable1 to variable2 (which is equal to variable3) so setting all variables to the same value.

And we could have assignment statements:

lvalue = rvalue

e.g. v1 = (v2 = v3 = a * b).

Another of C's spurious featurs, is that several expressions can be evaluated in the same assignment statement. This is beause, whenever the syntax of C calls for an expression, several expressions can be substituted, if they are separated by commas, e.g.

```
a = (t = x, s = y—f(t), s*7.3)
```

Assignments in themselves are valid expressions as we have just seen, so we have three expressions (two of them assignments) in the assignment statement. The value supplied to **a** will be the value given by the rightmost expression, in this case s * 7.3.

Another similar example involves the conditional operators **?:**. Depending on the value of an expression one of two other expressions will be evaluated, e.g.

```
(expression1 ? expression2 :
expression3)
```

If the result of expression1 is true then expression2 is evaluated, otherwise if the result of expression1 is false, expression3 is evaluated. The whole line will represent the value of whichever expression was evaluated.

SPECIAL OPERATORS

C has some special operators which are used to mimick machine code, in an attempt to optimise code produced by the compiler.

When a variable is incremented or decremented by one, there is likely to be a corresponding machine code instruction. Instead of writing:

```
v = v + 1
or
v = v − 1
```

the programmer is encouraged to use the shorthands:

```
++v v++
and
−−v v−−
```

The expression ++v represents v after it has been incremented, while the expression v++ represents the value of v and then increments it. This works in exactly the same way for decrementing v when v−− and −−v are used.

LOGICAL EXPRESSIONS

As we found out earlier in our discussion of loops, expressions can be used to represent Boolean true/false values. (Remember false is represented by a zero value and true by a non-zero value.)

There are also the following relational operators:

<	less than
>	greater than
<=	less than or equal
>=	greater than or equal
==	equal to
!=	not equal to

Notice that the 'equal' to operator is different from the assignment operator, which is a single equals sign, so that the compiler can tell them apart.

These operators are used for comparing expressions, for example,

```
X < Y
```

and can thus be incorporated into structured statements, e.g.

```
If (X < Y) printf ("X is less than Y");
```

Also included under the same heading are the logical operators, **AND, OR,** and **NOT.** Which are:

```
&& AND
|| OR
!  NOT
```

These operators give the true/false values according to their truth tables.

BITWISE EXPRESSIONS

The bitwise **and, or, exclusive-or** and **complement** operations are also available with the operators &, +, ∧, ∼, respectively.

For example, a bitwise AND:

```
1100010 &
0101000    = 0100000
```

and, a bitwise OR:

```
1101010 +
0101000    = 1101010
```

and, a bitwise EXCLUSIVE-OR:

```
1101010 ∧
0101000    = 1000010
```

and to complement a single variable:

```
∼ 1101010 = 0010101
```

There are also operators that have the ability to perform shifts. These are the operators << and >>, written **n<<expression** to shift the value of the expression left **n** places and **n>>expression** to shift the value of the expression right **n** places. For example:

```
(1101010) >> 2
is
0011010
```

Note that it is wise to declare all variables used in shifts as unsigned integers, so that complications will not arise from the computer's insertion of a sign bit.

AMPERSAND NOTATION

One feature that I have not mentioned thus far, is the ampersand operator. The address of any variable in memory can be denoted by a variable name, preceded by an ampersand. For example, if **a** was an integer and **b** was a character variable, then their addresses could be represented with **&a** and **&b** respectively, irrespective of their type.

ARRAYS

The concept of an 'array' or 'table' of variables of the same is common in other programming languages and is also facilitated in C.

The dimensions of an array are specified between square brackets. For example, the declaration:

```
int a[5];
```

makes available five integer variables, which can be referred to within the program as:

```
a[0], a[1], a[2], a[3], a[4] respectively.
```

The declaration:

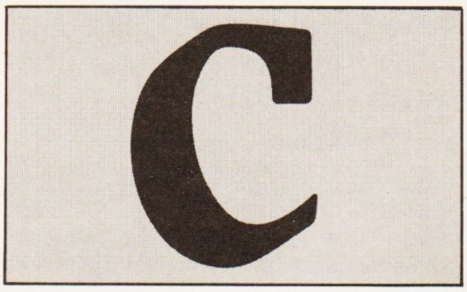
```
float a[3][4]
```

declares a two dimensional 3x4 element array composed of float variables:

```
a[0][0],a[0][1],a[0][2],a[0][3]
a[1][0],a[1][1],a[1][2],a[1][3]
a[2][0],a[2][0],a[2][0],a[2][3]
```

Notice that the subscripts (the numbers used to index each array element) start from 0 and go as far as the bound specified minus one.

In an array declaration, the subscript bounds specified have to be constant so that the compiler can allocate the storage required for the array before the program is run.



ARRAY STORAGE

For the last subscript of an array the elements are stored continuously.

The C compiler supplies a simple transformation on the base address and the supplied subscripts, to find the location of a particular element.

If there is only one subscript (i.e. the array is one-dimensional) then the array is as shown in figure 1.

and the transformation is quite simple. For example, for an array whose elements occupy exact locations in memory, with base address **a**, then the transformation is:

```
subscript location = a + (subscript no.
                                   − 1)
```

If **n** memory locations were used to hold an array element, then the above formula would be modified to:

subscript location = a + (subscript no. − 1) * n.

The transformation used by the compiler includes no boundary checking and so incorrect subscripts can give spurious results!

A two dimensional array a[3][4] is as shown in figure 2.

Note that for the last subscript the elements are listed together. Now because the compiler knows that there are three elements for each final subscript, it can calculate the position of the start of each list. Since there is no boundary checking the last subscript is only needed when the array is declared for the allocation of storage to elements.

This is why, when an array is passed as a parameter to a function, the declaration of the array parameter may be left with the last bound blank.

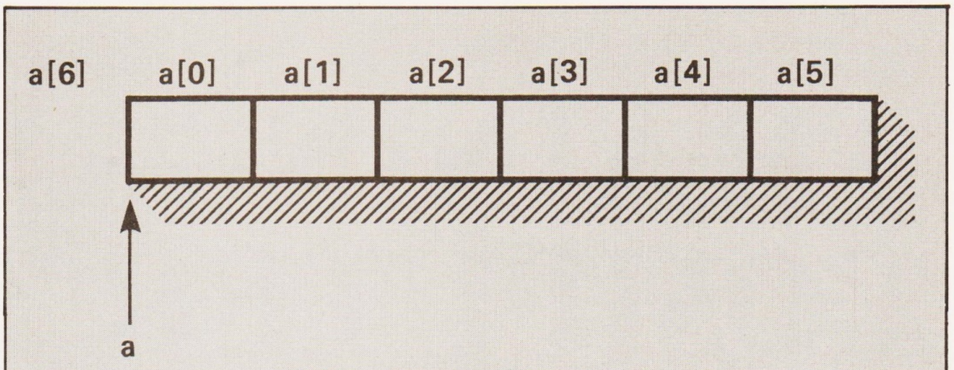


Figure 1 One dimensional array

STRINGS

```
int length (string)
char    string[];
(
   for (c=0: string[c]=='\0': c++)
       ;
   return (c);
)
```

In this example a function is used to return the length of a string. The function can return the length of any string irrespective of how the array holding the string was declared.

You will note that strings in C are held in an array of type **char** and are terminated with the null character '\0 '.

To call length, you pass it the base address of the array. This is represented in C by the array name (without brackets). The base address of an array can also be denoted by ampersand notation. For example, the base address of an array **a[2][2]** would be:

&a[0][0].

C provides some standard functions for string handling, the most popular being **strcmp:**

int strcmp (string1, string2)

Strcmp compares strings 1 and 2 and returns a value of zero if they are the same. Others include **fgets, fputs, sscanf, sprintf** which we will see later.

With **printf** strings can be output quite simply by using the %s format specifier.

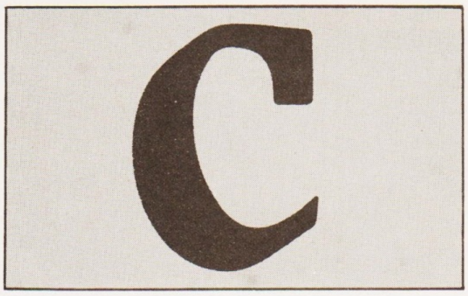
For example, if there was a string:

salutation HELLO /0

then it could be output with:

printf ("%s", salutation);

printf prints out the string until it sees the final null character. It then stops without printing the null character.



PROBLEM

Here is a problem to tax you. The solution will be published in the next issue.

A program is required to produce a list of telegram documents. The input consists of records terminated by newlines, that make up telegrams. (The 'newlines' are to be ignored by the program.) The words in each record are blank separated and telegrams include the special words STOP and ZZZZ. STOP represents a period and ZZZZ marks the end of the telegram. A null telegram is used to terminate the input stream. Such input would be typically produced by telegraph links.

The program output should consist of:

- A neat listing of each telegram, followed by:
- A summary of each telegram (for charging purposes). Each word costs one unit. Each long word (currently >12 chars.) costs two units and costing excludes the special words STOP and ZZZZ.

e.g.

INPUT
fly at once STOP all is revealed STOP
ZZZZ ZZZZ

OUTPUT
Telegram number 1.

Fly at once. All is revealed.
Charge: 6 words at 1 unit (as table) 6 units
0 words at 2 units 0 units
TOTAL 6 units

□□□

NEXT ISSUE we look at some of the more advanced features that make C such a powerful language.

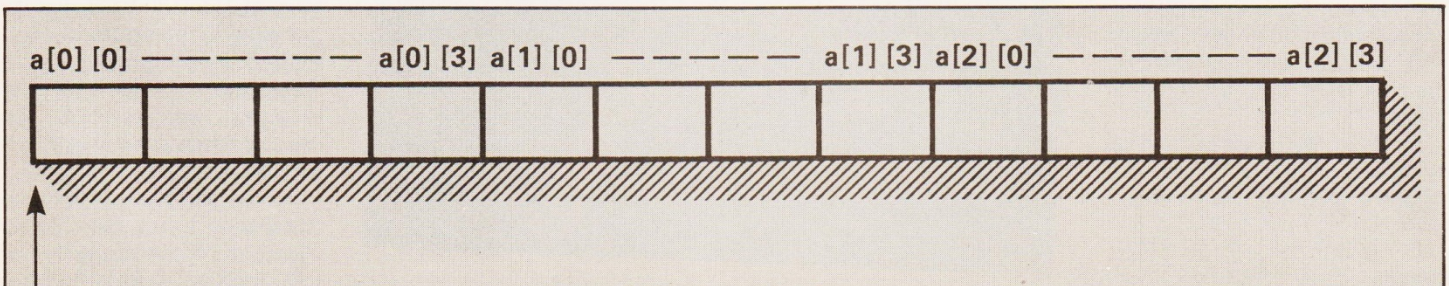


Figure 2 Two dimensional array

TRAINING COMPUTER STAFF

Bill Horne

The difficulty in getting properly trained computing personnel is no less a problem today than before — and here's why...

To judge by the outcry in certain quarters, one might imagine that the shortage of able computer personnel was something new, but it has been with us as long as computers have existed. A very big noise in the computer world once remarked; "Computer people are born, not made." There is a lot of truth in that, but it provides no solution to the problem of identifying the most fertile material and giving it the best possible training. It is equally important to avoid wasting too much effort on stoney ground, because the number of able teachers is limited.

Some people have expressed the hope that access to home computers will encourage and identify those who would benefit most from further training, but that depends to some extent on the way they see their equipment. If it is treated like a gramaphone, being used only to play pre-recorded games, the hope is unlikely to be realised. Those who, lacking a computer of their own, haunt the arcades are not always interested in how the games work.

In any case, possession of a computer is not enough in itself. When something fails to work, answers may not be forthcoming. Despite the avalanche of books about computers, the answers available are limited. The most serious failing in this area is that books tend to concentrate on the actual coding stage of program creation, ignoring the much more important phase in which the concept of the program is created. Writing BASIC is child's play in itself, but not too many children know how to overcome the absence of functions such as ACOS and ASIN, or can visualise a strategic algorithm.

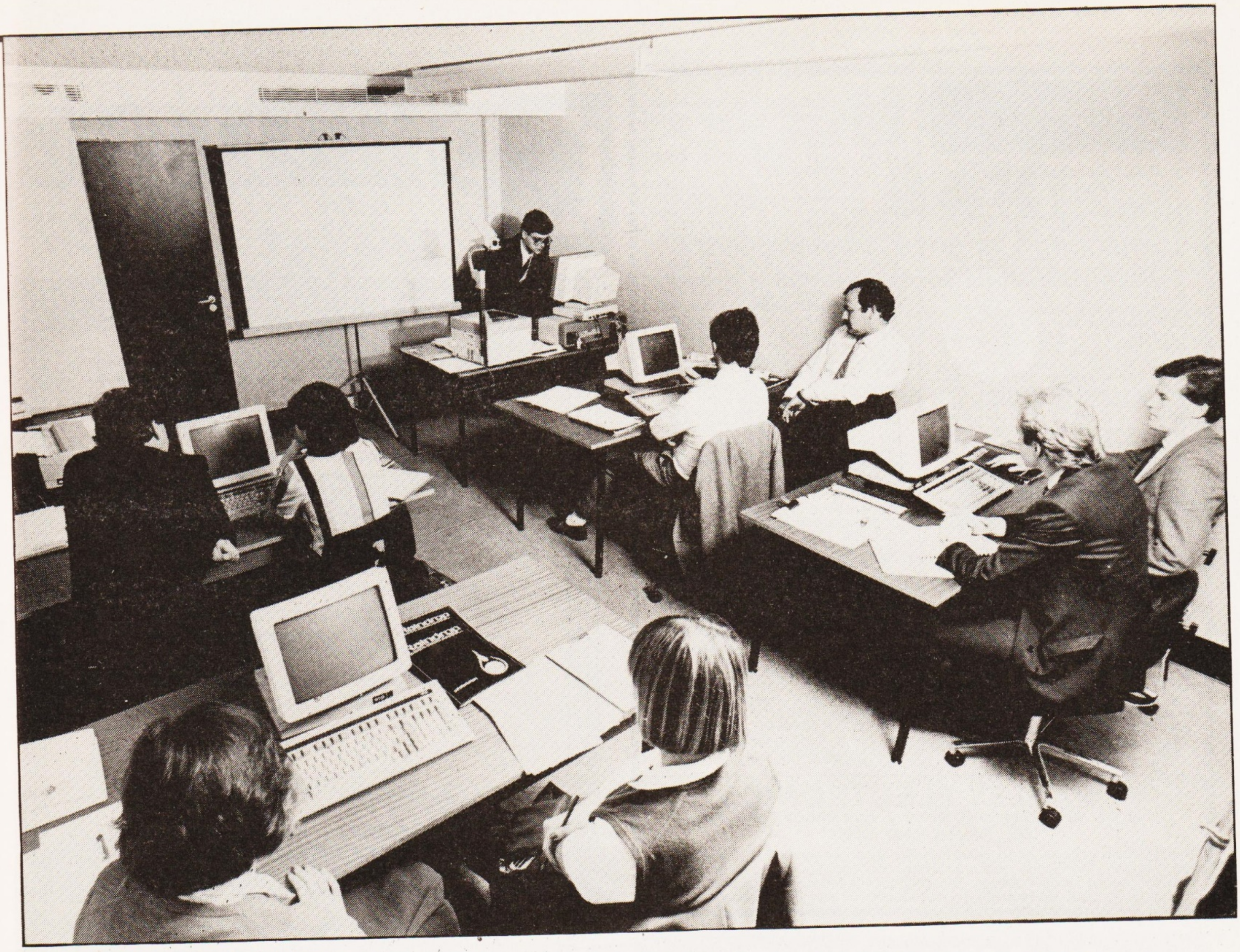
TRAINING: a student learns new skills at Control Data's PLATO learning centre — but remains part of a minority.

RESISTANCE TO COMPUTERS

In theory, the remedy should spring from computer lessons in schools, but this is hampered by a number of factors. There is still a lot of resistance to computers on the part of many teachers, stemming largely from a failure to realise the broader possibilities of computer training. There are few teachers well qualified to exploit those possibilities, and the rest see a threat to their future employment. Cases are known of teachers who have advised that computers are a threat to the world, creating unemployment and misery without doing anything genuinely useful. This, to put it mildly, is a negative attitude, even though it may seem justified at a simplistic level.

Universities can be expected to take a more rational view, but they often seem to be woefully behind the times. Having bought an obsolete type of central processor in quantity, they require all their students to learn how to use it, disregarding the fact that a re-learning process will be needed when the students get into industry.

Some academics are too anxious to teach the 'higher theory' of computing to bother with practical details. And it is in the more sordid practical aspects of computing that the



TRAINING: Raindrop Computers' training centre in London's West End. The facilities exist, but within a society sadly lacking in recruitment material.

need for additional effort is needed.

So what is the best route for young people who want to enter the professional world of computing? The answer depends to some extent on the aspects of computing which interest them most.

BASIC REQUIREMENTS

In the more formal areas of the industry, a good degree is useful, not because it necessarily signals a high level of ability, but because the employers are impressed by it. Once the degree has secured an appointment, it becomes devalued, and the progress of the graduate depends much more on his day-to-day performance in the working environment. This is often conditioned more by personality than by knowledge or education.

More commercial firms pay less attention to degrees, though they may provide opportunities for study if an employee shows special merit. The old student apprentice concept lingers on, but many companies feel that the best way to learn is to do the job, picking up the necessary information as the time goes by. There is sound evidence that they are not entirely wrong.

On the other hand, the stories of boy-wonders who qualify to have a company car before they are old enough to drive it may be seen as unrepresentative. Boy-wonders do exist, but they are the exception, rather than the rule, and their abilities sometimes burn out early. Their activities are often quite specialised, and outside that specialisation they have nothing to offer. The first requisite for a long-term stayer is that he always wants to explore something new.

The second necessary attribute is an ability to handle complex systems, visualising a long chain of events without losing track. This, in turn, calls for a willingness to ignore irrelevancies, to accept that a process works without always asking why, unless the question is relevant to the issue in hand. A measure of personal self-discipline is called for here, though some individuals seem to produce that automatically.

The best computer people are not always popular with their fellows, because they tend to see things with special clarity, and that can lead to friction, especially with those who take a more fuzzy view of life. When the latter find themselves in control of computer activities, a certain amount of acrimony is inevitable. Such people try to simplify a situation to a point where it no longer exists, and then say that they can't see the problem.

It must be remembered that computing is a relatively young phenomenon. The first real computers came into being no more than forty years ago. It was perhaps ten years before the industry really began to expand. Twenty years ago, there were no more than a few hundred really capable computer experts in Britain, and not so many more than that in the rest of the world. Now, one medium-sized firm working in the field may need two or three thousand specialists in computing. If even a small proportion of the unemployed could be trained to fill the vacancies, the industry would be greatly relieved.

MISMATCH

This is the real tragedy. The vast number of people out of work does not solely reflect a lack of jobs. It reflects a lack of a match between ability and the skills needed. If a way could be found to teach the unemployed new tricks — not necessarily very difficult ones — the out-of-work army could be significantly reduced, and the computer industry could grow more freely.

Against this must be set a certain lack of confidence within the industry at the present time, but it is the smaller firms that are worried. The industry as a whole, taken in the broadest sense, still needs more and more staff.

COMPUTING *Today*

NEXT ISSUE OUT FRIDAY MAY 10th

THE ALVEY PROGRAMME — billed as Britain's largest concerted effort towards the Fifth Generation of computing. But what has been achieved since the programme was established in April 1983? Find out, in the next edition of Computing Today.

LANGUAGES — We take a look at Metacomco's LISP, BCPL, and 68000 assembler packages for the Sinclair QL.

PLUS

BBC TEXT EDITOR — a simple wordprocessor for the BBC micro

AMTRAD/TANDY printer fix. Remove those unwanted line-spaces from your listings when using the Tandy printer with the Amstrad CPC464.

AND

THE ACADEMIC APPROACH — how far removed is the Academic world from the practicalities of computing? Revelations in the next edition of

Computing Today

THE EVOLUTION CONTINUES.

Organised by The Association of London Computer Clubs

18th, 19th & 20th APRIL 1985
FROM 10AM TILL 6PM

Software, Hardware, Peripherals and bits for all micros
Free unbiased advice from the club stands
Bring-and-Buy Sale on the Saturday

Plus

Other Features

Robotics Conference **Network Features**
Community Computing

Admission Adults £2.50
Under 12's £1.50

COMPUTING FOR LONDON.

The Personal Computer Show for Everyone

MCT 64K RAM BOARD

DESIGNED FOR USE WITH THE COMMODORE C16

Easily fitted and without modification of the main PCB.

The Ram Board will allow 60671 bytes free to BASIC programming

Fully compatible with existing C16-Plus 4 software.

Dealer and Distribution enquiries welcomed

including p&p and VAT **£59.95**

Micro Component Trading Company
Group House, Fishers Lane, Norwich, Norfolk
England. **Telephone (0603) 666966**

ORDER FORM

Please supply................ Ram board(s) @ £59.95

I enclose my cheque/postal order/Girocheque for £....................

I understand that a full refund less post and packing will be made by MCT if I am not fully satisfied and provided the Ram board is returned within 10 days in its original packing and in a working, new condition.

Signed ....

Postal address ....

....

SHARP MICRODEALER

SHARP MZ-3541

CPU	Z80A (two), 80C49
MEMORY	128K RAM, 8K ROM
LANGUAGE	Sharp BASIC
MASS STORAGE	Twin integral 5¼" floppy disk drives, total capacity 1.28 Mb
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, interface for extra external floppy disks
DISPLAY	Monochrome monitor, colour optional
GRAPHICS	80 by 25 text, 640 by 400 high-resolution graphics
SOUND	Single channel

Notes: The Sharp MZ-3541 is aimed at the businessman. RAM is expandable to 256K, while two disk drives may be added externally to complement the integral pair. Colour is only possible with the optional graphics expansion RAM. One Z80 handles the main CPU activities while the other handles peripheral activities. The third processor handles the keyboard. The availability of CP/M means a ready supply of business software.

LONDON

SHARPSOFT LTD.

With the Sharp user in mind.
Sharp MZ 700 including cassette
£199.99

Sharpsoft Ltd, Crisallen House, 86-90 Paul Street, London EC2. Tel: 01 - 729 5588.

LANCASHIRE

STATION ELECTRICAL
Dept CT, Coastal Road, West Bank,
Lancs LH26HN. Tel: 0524 824519

Large range of software, books and peripherals for SHARP MZ 700. 2nd hand computers bought, sold and part exchange. Also repairs.
SAE for lists

MICRODEALER — ORDER FORM

Please include my business details in the next available issue of Computing Today:

Business Name: ....

Address: ....

....

Details: ....

Tel. No.: ....

Open Hrs: ....

Contact (Office Use Only): ....

Post To: Computamart, Computing Today, 1 Golden Square, London W1.

ONLY £20.00

ACT MICRODEALER

xi APRICOT

CPU	8086
MEMORY	256K RAM
LANGUAGES	Microsoft BASIC, Personal BASIC
MASS STORAGE	No cassette drive Integral Sony 3½" 315K microfloppy disk drive Integral 5 or 10 Mb hard disk
OS	MS-DOS 2.11 with GSX bundled CP/M-86 (not yet available) Concurrent CP/M-86 (not yet available)
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, Microsoft mouse
DISPLAY	Monitor (supplied)
GRAPHICS	80 by 24 text with block graphics 800 by 400 high-res graphics under GSX
SOUND	No

Notes. The Apricot xi is a development of the award-winning Apricot, and replaces one of the latter's disk drives with an integral hard disk, providing vastly increased storage with faster access. Memory may be expanded in 128K increments to a maximum of 768K. The languages and operating systems mentioned above come bundled (except for Concurrent CP/M) and four software tools are also bundled, including an asynchronous package for use with the optional modem card.

CBM MICRODEALER

Notes: The Commodore 64 is a popular micro with a great deal of games software available. There is also some business software available.

The Commodore 715B is the top model in the 700 range of business machines.

NASCOM MICRODEALER

NASCOM 3

CPU	2 MHZ Z80
MEMORY	8K or 32K inbuilt RAM (expandable to 60K)
LANGUAGE	Full Microsoft BASIC
MASS STORAGE	Single or twin 5.25" disc drives 350K capacity per drive
OS	NAS-DOS or CP/M 2.2
KEYBOARD	Full size QWERTY
INTERFACES	RS232 and 16-bit parallel
DISPLAY	40 or 80 column 25-line display
GRAPHICS	High resolution graphics with 8 foreground and 8 background colours (400 x 256 pixels) Double density graphics with 2 colours (800 x 256 pixels)
SOUND	No

COMPUTING TODAY

Lineage: 40p per word.
Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
CAROLINE
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

HARDWARE

NEWBRAIN & SANYO
Professional Micro Computers for the price of hobby machines.

NEWBRAIN ON SPECIAL OFFER
With over £200.00 free software. (Accounts, databases, etc, etc)
Limited Offer-ring now!

SANYO 550/555 COMPUTERS
Micropro Wordstar, Calcstar etc at no extra cost! **Printers:** Epson, KDC, Juki, Shinwa, Daisystep 2000 etc. **Sanyo Monitors & Recorders.**
Call STEVENAGE (0438) 812439
anytime for hardware/software lists
Mail Order and **Access** facilities.

ANGELA ENTERPRISES
4 Ninnings Lane, Rabley Heath, Welwyn, Herts AL6 9TD.

HANTAREX RGB MONITOR £175. Nascom 48K ram programmable graphics, basic zeap, dis/debug, NAS-SYS 3, cased, 3A PSU, £170 or offers. Phone Nottingham (0602) 255693.

FLOPPY DISKS

AAAA

authorised **APRICOT** and **SANYO** dealers

FLOPPY DISKS

	DYSAN*	3M	NASHUA	SONY
SS40	17.00	14.50	13.15	3.5"
DS40	24.50	18.00	16.50	↓
SS80	24.50	21.00	17.45	34.50
DS80	29.50	23.50	18.95	45.50

*Special offer **DYSAN** branded disks deduct £1.50. Prices per box of 10 excluding VAT. **POST FREE** discounts for **any** larger quantity. £2.00 extra in "SEE 10" Library case. Send C.W.O. or write for full list of supplies to:

A. M. A. COMPUTER SYSTEMS AND SUPPLIES
Dept F. 8 Glebe Street, Beeston, Nottingham NG9 1BZ.
Tel: (0602) 255415.

MISCELLANEOUS

FOR MEN!
VASECTOMY
FREE DETAILS
Everlasting alternative to the Pill.
No fuss. No waiting lists. One visit. Low cost. In . . .
BIRMINGHAM, LONDON, PLYMOUTH, EDINBURGH, MANCHESTER, SOUTHAMPTON, GLOUCESTER, NEWCASTLE, SWANSEA, LEEDS, NORWICH, TAUNTON, LIVERPOOL
Write or phone for booklet (plain cover).
LONDON 01-388 2585
LEEDS 0532 440685
MANCHESTER 061 832 4260
Name__________
Address__________
__________ CT 5
Marie Stopes House,
108 Whitfield St., London W1P 6BE
Caring clinics since 1925
MARIE STOPES
SEND NOW

SPECTRUM P.I.O. 16-Bit Programmable £9.95. Also **SPECTRUM ADC/DAC** 8-Channel, 8-Bit, with 240V relay output, includes 16-Bit P.I.O. £24.95. Also **SPECTRUM JOYSTICK INTERFACE** using Kimpston standard with any Atari-type controller £5.50. Cheque/PO to: R. Chandler, 5 Norfolk Place, Leeds, LS7 4PT. Tel: 693540. Trade welcome.

SOFTWARE

VIC-20 USERS
A renumber facility for your basic programs requiring no memory expansion. Cassette £2.00 including post and package.
Dept 2, Bolling Software, PO Box 42, Bradford, West Yorkshire, BD4 7UD

FUZE
FUZED
DEFUZED
CONFUZED
CONFUZION

ALARMS

BURGLAR ALARM Equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. CWAS Ltd., 100 Rooley Avenue, Bradford BD6 1DB. Telephone: (0274) 731532.

REPAIRS

COMPUTER REPAIRS IN SOUTHEND
SPECTRUM.BBC.ELECTRON.
CBM 64. VIC 20
Minnie Micro Electronics
12 Eastern Esplanade, Southend
Tel: 0702 62033/615809
Open 7 days a week

FOR SALE

NEW MICROS utilities software, S.A.E. for bargain prices. WANTED quality games/utilities, tapes, £1.00 paid plus postage, realistic market values, now starting G. Kennard, 18 New Road, Bude, Cornwall. Tel: 0288 4676.

SD DATA SERVICES is launching an exciting new range of educational software for Spectrums. Different ages and different subjects. Also we are still looking for new programs — original or educational — to complement our new range. Write for details to SD Data Services, Chestnut House, Bitteswell, Leics.

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: 40p per word per issue (minimum of 15 words). Please state classification and post to **COMPUTING TODAY, CLASSIFIED DEPT., 1 GOLDEN SQUARE, LONDON W.1.**

Please use BLOCK CAPITALS and include post codes.
Name (Mr/Mrs/Miss/Ms) ....
(delete accordingly)
Address ....
....
....
Signature .... **Date** ....
Daytime Tel. No. ....

Please place my advert in **COMPUTING TODAY** for . . . . issues commencing as soon as possible.

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

CHESHIRE

Computer Junk Shop

We Buy, Sell, Break Computers & Peripherals.
10 Waterloo Rd, Widnes, Halton. Tel: 051 420 4590.

HERTFORDSHIRE

NEWBRAIN & SANYO
HARDWARE & SOFTWARE
Printers, Epson, KDC, Juki etc. Monitors, Tape Recorders, Books, Expansions, CP/M. **Sanyo 550/555 Computers.** Access/Mail Order. Ask for details.
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

TO FILL THIS SPACE PHONE CAROLINE ON 01-437-0699

SOUTH LONDON

CROYDON COMPUTER CENTRE
Authorised Acorn Service Centre
29a Brigstock Rd., Thornton Heath, Surrey. Tel: 01 - 689 1280
BBC, Acorn, Electron, Genie, Oric, Kaga Microvitek Zenith Monitors. OKI 80, 82A + 84 Printers. Paper, Ribbons, Software etc. BUY-HIRE.

MIDDLESEX

SCREENS MICROCOMPUTERS
6 Main Ave., Moor Park, Northwood, Middx.
Tel: Northwood (09274) 20664
Telex: 923574 ALACOL G.
Official Dealers for: Acorn, Atari, Amstrad, Apricot, Commodore, Dragon, Einstein, Memotech, Oric, Psion, Sirius, Sanyo & Sinclair.
Open 6 days per week

NORFOLK

ANGLIA COMPUTER CENTRE
88 St Benedicts Street, Norwich.
Tel: (0603) 29652/26002.
Open: 6 days 9am-5.30pm.

SCOTLAND

VICTOR MORRIS GLASGOW
TANDY TRS 80, VIC 20, VIDEO GENIE, APPLE PANASONIC, CUMANA, EPSOM ETC.
340 Argyle Street, Glasgow G2: 041 221 8958

SUSSEX

GAMER
24 Gloucester Road, Brighton.
Tel: 0273-698424.
Open: Mon-Fri 10am-5.30pm, Sat 9am-5.30pm.

TYNE AND WEAR

HCCS ASSOCIATES
533 Durham Rd., Low Fell, Gateshead. Tel. Newcastle 821924.
Open: 6 days 9am-5.30pm (Sat 10am-5.30pm). Specialists in: Acorn, BBC, Video Genie, VIC 20.

CLASSIFIED ORDER FORM COMPUTAMART

Please include my business details in the next available issue of Computing Today:

Business Name: ....

Address: ....

....

....

Details: ....

....

....

....

ONLY £17.50!

Tel. No.: ....

Open Hrs: ....

Contact (Office Use Only): ....

Post To: Computamart, Computing Today, 1 Golden Square, London W1.

Software News

INNOVATIVE SOFTWARE

from the professionals

SANYO 550/5 SOFTWARE

Cashman — Very colourful arcade game for Sanyo.	£ 26.00
DC-10 — Flight Simulator. Over 30 radio aids.	£ 26.00
Demon Seed — Another excellent arcade game.	£ 26.00
DS-DOS — Double sided DOS (1.25)	£ 37.00
DS-DOS80 — Quad DOS 80 track 9 sector (1.25).	£ 39.00
DS-DOS PLUS — Quad DOS 80 track 10 sector (1.25 or 2.11 state)	£ 39.00
Emperor — A full size wargame from Roman times.	£ 26.00
Freeze Frame — Text & graphics screen dump, colour too.	£ 32.00
French Tutor — Just that.	£ 26.00
Grafiti — Computer assisted graphics. 256K.	£ 35.00
Joystick — de Luxe quality — 2 button	£ 37.46
— 3 button	£ 46.18
Joystick Extender Cable	£ 14.00
King Arthur — 20 minute wargame.	£ 18.00
Major Motion — A fast road battle arcade game. 192K.	£ 26.00
Master Graph — Even plots 3-D curves.	£ 62.00
Maz — A fantastic four part maze game.	£ 20.50
M-DISK — Extra disk in RAM. 256K.	£ 26.00
MI-KEY — Re-define the whole keyboard or part of it.	£ 27.00
MI-TERM — Communications terminal program.	£ 44.00
Mighty Mail — The best mailing list handler available.	£ 82.50
Mysterious Adventures — A series of 10 classic Adventures. From damsels and dragons to space travel!	
Any 1	£ 18.00
Any 3	£ 45.00
All 10	£120.00
Personal Money Manager — Personal finances handler.	£ 40.00
Picasso — The ultimate Paint program.	£ 77.00
Quick & Simple — Very simple database.	£ 39.00
Soft Spool — A software print spooler. 192K.	£ 26.00
Solitaire — Patience, Poker Squares, Blackjack, etc. Terrific graphics. 192K.	£ 29.00
Super Calc III Zaps — Convert the spreadsheet to the Sanyo.	£ 24.50
Super DS-DOS — DS-DOS plus Utility Disk No. 1.	£ 62.00
Super Zap — Full disk access and edit.	£ 37.00
Thunder Chief — A shooting, bombing arcade game. 192K.	£ 26.00
Utility Disk No. 1 — Collection of utilities.	£ 37.00

All programs supplied on a single sided formatted disk for 128K, unless mentioned, and compatible with colour monitor. DOS's require ownership of either MS-DOS 1.25 or 2.11.

All prices are VAT exclusive. P&P 75p per program.

TEL: (0424) 220391/223636

MOLIMERX LTD

A J HARDING (MOLIMERX)

TELEX 86736 SOTEX G

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

Send 24p A4 sized SAE for full list.