AN ARGUS SPECIALIST PUBLICATION

Computing today

March 1985 90p

Memory expansion – wise up on ways to enlarge your machine's capacity

CP/M Preliminary Macro Assembler – complete listing plus guidance notes

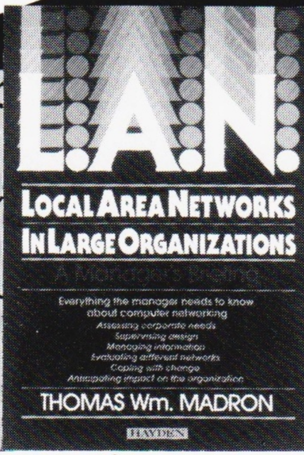
BBC Ladder Logic Simulator – CAD on a shoestring

Getting Into Print – the facts on interfaces and printers

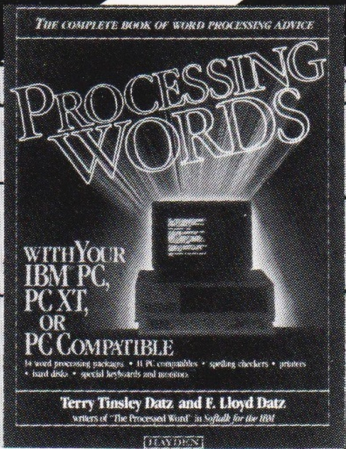
Toshiba MSX – everything under the sun?

Could a modem be your missing link? We review the Interlekt Portman

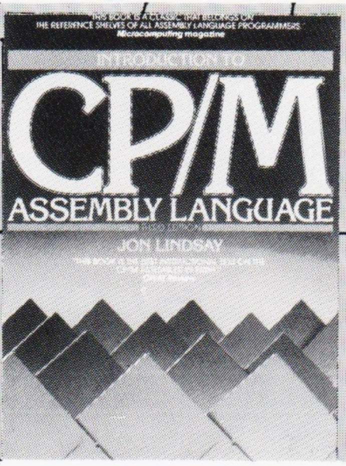
Hayden



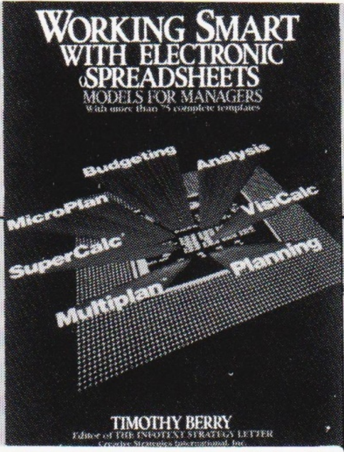
A totally comprehensive analysis of local area networking from the two most important perspectives: organizational and technical. Gives a clear introduction to the concepts of networks and communications terminology.



A complete introduction to word processing on the IBM PC, written by two prominent authors of IBM word processing. Helps the reader to choose word processing software, and discusses advantages and capabilities of the IBM PC and the features of various software programmes.



"... this book warns the user against making some of the more common errors that the novice programmer often encounters. The book is well indexed, it has a clear, consistent style and it does a good job in covering the subject of CP/M assembly language programming." — CP/M Review



This book teaches you how to think with spreadsheets. Covers sales forecasts, business plans, budgets, and financial statements, all implemented on spreadsheets. Encourages you to build your own models and then to customize them. Makes spreadsheets work *for* you.

Send coupon with payment to:

HAYDEN BOOKS, 1 Gower St, London WC1E 6HA

Name:
(please print)

Address:

Please send me:

	☐ Hayden Book Catalogue	Free!
8104-6205-2	☐ Local Area Networks	£15.25
8104-6359-0	☐ Processing Words	£15.25
8104-5210-3	☐ Intro to CP/M Assembly Language	£15.25
8104-6203-6	☐ Working Smart-Electronic Spreadsheets	£17.05

POSTAGE FREE! **Total:** £

Make cheques payable to: International Book Distributors

Visa, Access, American Express

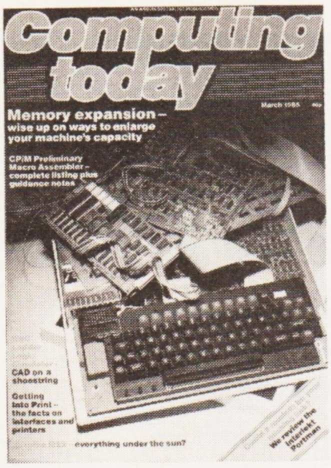
Card no.: Expiry date:

Signature:

CONTENTS

VOL 7 NO 1 MARCH 1985

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



Editor:
Don Thomasson

Assistant Editor:
Jamie Clary

Advertisement Manager:
Anthony Shelton

Advertisement Copy Control:
Sue Couchman,
Lynn Collis

Chairman:
Jim Connell

ABC Member of the Audit Bureau of Circulation
ISSN 0142-7210

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

Subscription rates: UK £15.00 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).

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

Potential contributors are asked to take note of the points raised on page 15 of this issue.

TALKING SHOP 5

What 'market forces' will significantly affect the home computers found in high street stores? Don Thomasson makes an observation.

NEWS 6

The very latest in hardware, software, education and commerce, plus answers and errors, and note about the new-look *Computing Today.*

MINERS: A COMPUTER SOLUTION.........................16

The dispute between the NCB and the NUM continues as we go to press, but read how one software package — the Priority Decision System — is being used in the delicate negotiations between the two parties involved.

GETTING INTO PRINT...19

Don Thomasson declares what should be known about printers and interfaces before a selection is made.

HORACE GROWS UP 22

This, the first in an occasional series of short stories, is a curious tale of a 'sentient' computer installation that discovers that mysteriously human quality of . . . attraction to others beings!

DOWN A WIRE.................26

Some essential background information on Modems, provided for us by Henry Budgett.

IBM AT HALF PRICE?.....28

Ivan Berenyi reports on an Asian portable micro, destined for the European market, which, at less than 40% of the retaail price of the IBM PC and offering 'complete compatibility', promises to give the American giant a run for its money.

EXPANDING MEMORY...29

Bill Horne describes some of the commonly employed techniques for squeezing more memory address space from your micro.

AVERAGES AND TRENDS 31

M J Bedford comments upon the dangers of accepting *verbatim* a statisticians interpretation of sampled data, and provides us with a program for the Amstrad CPC464 which plots a histogram for inputted data.

LADDER LOGIC 34

Design simple logic circuits with this program for the BBC micro.

BOOK PAGE 39

Garry Marshall provides his usual summary of the best selected computing books of the month.

TX/RX 41

Henry Budgett complements his introduction to modems, with a review of a very fine example of the technology: The Interlekt Portman modem.

PRINTOUT 43

Readers' letters.

TOSHIBA MSX..................44

Pete Freebrey reviews the Toshiba HX-10 MSX compatible microcomputer.

INSIDE BASIC: 2...............48

We continue our INSIDE series, with a description of how most micros represent numbers within themselves.

MACRO: 1 51

A useful utility program which allows the Apple programmer to define true Z80 machine code macros under CP/M.

ALGORITHM ANGLES ...54

The first in a regular series devoted entirely to algorithmic solutions for provocative problems.

AMSTRAD LETTER WRITER.............................58

A small-scale wordprocessing program for the AMSTRAD CPC464.

COMPUTER DATA SECURITY 61

Bill Horne discusses the problems of maintaining a secure computer installation.

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)

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Please use BLOCK CAPITALS and include postcodes.

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

CT March '85

TALKING SHOP

Don Thomasson

It was just a passing phrase, "The market's very nervous." spoken in a pub at lunchtime, but it set off a train of thought. The speaker was talking about the market in home computers, computer programs and computer books. What was there to be nervous about in that area? Wasn't it time for the predicted boom to blossom out?

Discreet enquiries revealed that a number of people in the computer field felt that the phrase was apt, and a number of reasons for it began to emerge.

For any market to thrive, there must be a good match between supply and demand. If the public want a particular kind of machine more than any other, then it is sensible to make machines of that kind. But this calls for accurate prediction. It takes time to create a new computer, however small, at least a year if it is to be a good design, and during that year the view of potential buyers can change. Almost the only satisfactory solution to this difficulty is to provide a machine that can be all things to all men – a games computer which will handle business transactions and yet be a delight to the 'hacker'. That is by no means impossible, providing the designer does not have too many pre-conceived ideas regarding the targets which his product should hit.

Merely creating a new machine, however, is only the beginning. It must then be sold, first to the middlemen, and then to the retailer and lastly to the public. That order is important. Unless sales are to be made exclusively to the public by mail, the machine must be accepted by the middlemen if it is to get anywhere at all. Then the trade must accept it, and finally the public, though by then the choice is likely to be limited.

Some manufacturers seek to short-circuit the sales process by persuading the public that a given product is the best thing since sliced bread. A few, unfortunately, do so by making statements about other computers that are manifestly untrue.

INFORMATION

So how can the potential buyers discover the truth? In theory, they should be able to learn about a machine by reading reviews in ***Computing Today*** and other journals, but that is not always a complete answer. With more and more journals publishing reviews, manufacturers are reluctant to part with review machines for more than a couple of weeks, and that is not really long enough for a thorough assessment.

Then there is the problem of the machine which is below par. The best response is to say nothing at all. That avoids one kind of argument, but may lead to another kind, the machine's sponsors claiming that failure to review a machine is effectively a statement that the machine is no good. Which may well be true.

The best guide for a potential buyer might well be the casual conversation heard in the offices of a computer magazine! Strong views are usually filtered out before they reach the printed page, but can be expressed verbally with greater freedom. Not that all the views are accurate, but contact between those holding different opinions tends to even out the contradictions and produce a balanced outlook.

Amateur computer clubs may provide another source of useful information, but their standards vary widely. Some are dedicated to a particular machine or make, while in others the blind lead the lame, but a few are of higher standard, and can offer genuine help.

OUR SOLUTION

The solution we plan to offer is to explain what characteristics to look for in a machine intended for a given kind of application, and by comparison of different approaches to the implementation of particular features. No doubt that will still land us in trouble of one sort or another, but if we evade the issue we will not be doing our job.

An eternal problem relating to new computer designs is that documentation tends to lag behind the completion of hardware and firmware, and may even be issued in 'temporary' form. Full marks to AMSTRAD, who not only got the CPC464 into the shops on time, but backed it with comprehensive documentation. Only nobody seems to know about the documentation, the shops selling the machine included.

Incidentally, the pattern of sales was a surprise to AMSTRAD. Printer cables ran short, because so many buyers already had a printer, or were willing to buy a relatively expensive one, while many were augmenting or replacing a computer which they already owned.

ACID TEST

So what is making the market nervous?

Fundamentally, the worries stem from that need to match supply and demand. The sales pattern has not followed the expected trends.

That could be because the expectations were wrong, or it could be due to ineffective presentation of the goods for sale.

Christmas will bring the acid test. Retailers will base their reordering on what they have sold during the festive season, and that will warn manufacturers of any need to revise their production schedules, though the necessary moves may not become clear until early Spring.

Meanwhile, a lot of people are biting their nails and wondering which way the general public will jump. Surely, they say, the really big market is still for games? Or are there enough more serious users to swing the balance?

Our view at ***Computing Today*** is that the buyer is becoming more critical, and a little more knowledgeable. He realises that the cost of the basic computer unit may be a relatively small proportion of the total expenditure needed to give him a complete system, and he may even be ready to put a computer he already owns on the shelf, replacing it with something more effective.

It is possible that sales of computer books may point the way to the future. There are indications that books are selling in advance of the hardware and software to which they refer. For a few pounds, the book buyer can find out a great deal about a machine which would cost him a few hundred pounds. That he has bought the book at all shows that he is interested in the subject, and that gives a hint regarding future sales. A word of caution, however; The range of books available expresses the view of publishers regarding future demand, and many publishers are as nervous as the rest...

Computing Today. is a magazine for the experienced home computer enthusiast.

We are constantly on the lookout for articles which address those interested in the techniques, principles, and methods which underlie computer hardware and software.

Computing Today is willing to consider articles for our FRAMEWORK features, which are designed to offer readers a ***required*** amount of information ie. not program listing, but detail enough to allow the reader to perform the actual writing of the program itself.

Similarly, we are keen to receive contributions for ALGORITHM ANGLES, our 'How To' page which is, again, designed to encourage readers to try-out ideas on their micros.

In fact, if any reader feels that he or she may be willing and able to contribute to ***Computing Today,*** but would prefer ***not*** to commit an idea to paper without prior consultation with us, why not give us a ring and we can discuss the matter over the telephone. The number to ring is 01 437 0626, asking for the Assistant Editor of ***Computing Today.***

NEWS

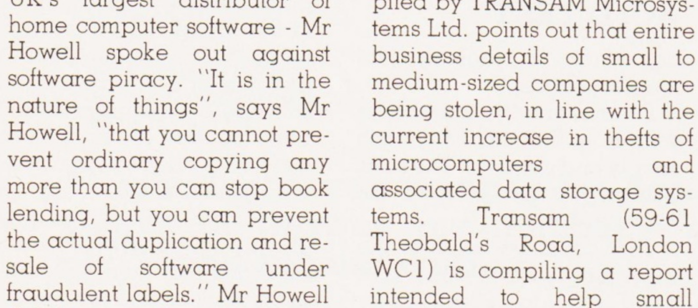
CRIME, CRIME, AND MORE CRIME

Conservative MP William Powell's decision to introduce a Private Member's Bill on software copyright is catching on with his parliamentary colleagues.

As reported in *Computing Today* last issue, Mr Powell came sixth in a recent Private Member's ballot, and as a result the introduction of the Bill is expected to be highly placed on the Parliamentary timetable. Now, David Howell, Conservative MP for Guildford and former Transport Minister, has publicly expressed his support for the Bill. During a recent visit to the Headquarters of Websters Software - the UK's largest distributor of home computer software - Mr Howell spoke out against software piracy. "It is in the nature of things", says Mr Howell, "that you cannot prevent ordinary copying any more than you can stop book lending, but you can prevent the actual duplication and re-sale of software under fraudulent labels." Mr Howell didn't mention just how we were going to stop piracy in its tracks, but we think what he actually meant is that the laws concerning software copyright could be tightened to dissuade pirates from continuing their criminal activities.

But if the experiences of Rod Cousens, Vice Chairman of The Guild of Software Houses (GOSH) and Managing Director of Quicksilva, are anything to go by, a great deal of this piracy happens abroad. Mr Cousens was apparently 'staggered' by the level of duplication occurring in Singapore - of all places - where Mr Cousens was on a visit to find distributors for Quicksilva products. "Piracy is almost a preoccupation" says Mr Cousens, "There is even a publication detailing how to break security measures employed by counterfeiters!". To cap it all, Mr Cousens discovered 'copies' of his own company's games for sale at half-price. Needless to say, Mr Cousens is taking action against those responsible for this outrageous crime.

Meanwhile a report on computer crime being compiled by TRANSAM Microsystems Ltd. points out that entire business details of small to medium-sized companies are being stolen, in line with the current increase in thefts of microcomputers and associated data storage systems. Transam (59-61 Theobald's Road, London WC1) is compiling a report intended to help small businesses protect themselves against computer crime - a summary of the report is expected early 1985, but Transam is anxious to hear from people who have fallen victim to this new trend in computer crime. Transam's Nigel Stride says: "Computer crime is becoming more widespread as it becomes less scientific" as businesses fall victim to loss of equipment and valuable records.



QUINKEY ATTRACTS RUSSIAN EDUCATORS

Three top Russian education officials recently visited Newcastle to look into the city's £284,000 investment in information technology for schools.

Included on the itinerary was a visit to Heaton Manor Comprehensive School which was awarded £20,000 to further its investigations into the use of Microwriter's QUINKEY with primary schoolchildren.

Quinkey is the alternative keyboard for the BBC microcomputer, taking over completely from the existing computer keyboard by plugging straight into the analogue port at the back of the BBC computer and then loading the accompanying software. All types of input, such as word processing and programming, can be carried out with Quinkey, giving access to the full range of BBC keyboard input facilities.

Based on the same five finger keyboard as the Microwriter word processor, the greatest advantage of Quinkey in the classroom is its multi-user capabilities. Used in conjunction with the multi-word processing software, QUAD, up to four Quinkey keyboards can be linked simultaneously to the BBC microcomputer providing each user with his/her own dedicated section of memory and screen, so multiplying the accessibility of the computer by four.

Georgy Veselov, Minister of Education of the Russian Soviet Federated Socialist Republic; Dr Mikhail Kondakov, President of the Academy of Pedalogical Sciences and Mrs Valentina Ionova, Inspector in the Moscow Department of Education, who has a particular interest in teachers' training, were shown around the Teachers' Centre and Moorside Community School where they had discussions with educational psychologists.

Dr Kondakov, Mr Veselov and Mrs Ionova were later entertained by the Lord Mayor and Lady Mayoress, Councillor Norman and Mrs S. Stockdale.

ACT AND VICTOR IN COURT TANGLE

Following our report on the parting of ways of ACT and Victor Technologies (*Computing Today*, February 1985) Victor has filed two high court actions against ACT, in an effort to disentitle ACT from further use of the SIRIUS trademark.

Victor are seeking additional court orders, such that:

- Victor will be substituted as owners of the SIRIUS trademark
- ACT will be injuncted as owners of the SIRIUS trademark except in relation to Victor's computer products

In retaliation, ACT has filed a counter High Court Action for an injunction to prevent Victor from using the trademark.

SPECIAL OFFER FROM MICRO DEALER

As a special offer to the public who require cassette recorders, Micro Dealer U.K., exclusive distributor of the Omega Data Recorder, is offering a £5.95 Azimuth Alignment Tape free with each purchase of the Omega.

The Omega Compucorder Model 8006 is said to offer a higher level of reliability than general purpose recorders, and has a digital tape counter and external remote control as standard. These features, coupled together with a built in microphone and the facility to use an external microphone, appears to make the Omega a good-value data recorder.

MICRONET AT DOWNING STREET

Micronet 800 was invited to attend a party at No 12 Downing Street to provide constructive entertainment for the children of Tory MP's.

The party was given by the Prime Minister on December 18th for 12 year old John Wakeham, eldest son of John Wakeham, the Chief Whip. John Wakeham junior, who lost his mother in the Brighton bombing incident, was treated as a VIP for his 12th birthday party. he had the choice of entertainment to be provided and top of his list was a request for Micronet 800 to be there for a demonstration and for his use afterwards.

Micronet subscribers aware of the location of the party and of the demonstration taking place were sending messages suggesting a demonstration of page 600186, the Labour Party public network based on Micronet 800. These suggestions were very humourously appreciated but all the same swiftly erased!

HI-TEC INDUSTRIES GET NEW INITIATIVE

A group of business people, financial advisors, research scientists and academics are getting together to launch a new national initiative to help Britain's Hi-Technology Industries.

While some companies are meeting the Hi-Technology challenge with some success, there are forecasts that some parts of Britain's Hi-Tehnology industry are facing a crisis of survival.

Some experts blame Government inactivity —others a lack of effort by the industry itself.

Determined to look for ways of helping UK Hi-Technology companies and encouraging further innovation, Reading Chamber of Commerce, Reading University, Accountants Deloitte Haskins & Sells in co-operation with the Southern Region of the Confederation of British Industry and the Thames Valley Branch of the Institute of Directors are mounting a Forum at Reading University on Friday, March 1st.

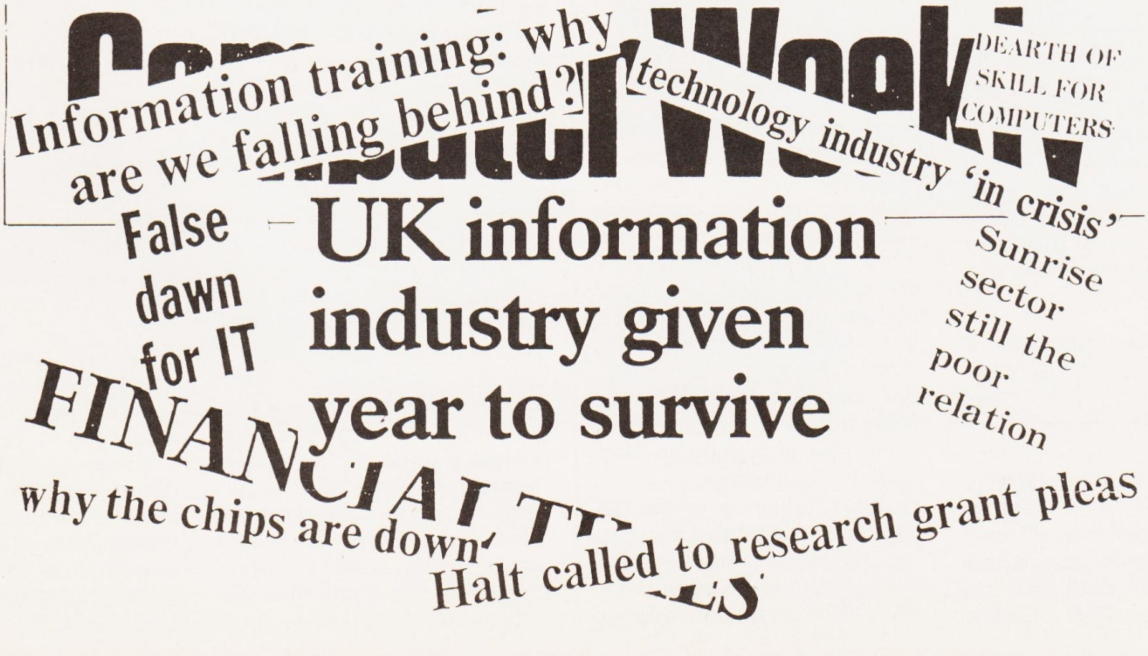
Called "Hi-Technology in Britain — Time for a New Direction?" the Forum will hear speakers on topics ranging from Education and Training to Government support and the importance of Marketing in Technical Innovation. The speakers will include Lord Robens, David Baldwin (MD of Hewlett Packard), John Belton (Director and Chairman of Deloitte Haskins & Sells Hi-Technology Industry Group), Bob Sheaf. (Industry and Technology Officer of the EEC Commission), Dr Gary Acres (Director of Group Research of Johnson Matthey), and Professor Derek Smith formerly of Queen Mary's College Industrial Research, London. Delegates will then be able to discuss and debate key issues raised during the Forum which is aimed at producing forward thinking conclusions.

The Forum is aimed at all those involved in the industry and will bring together scientists, business people and financiers. The guests will include MPs, MEPs, and members of the Upper House, and representatives of Government, EEC Departments, and local Government.

Delegate places at the Forum, cost £65 + VAT to include morning coffee and lunch, are available from Ian Cleveland, Chief Executive, Reading Chamber of Commerce, 43 West Street, Reading RG1 1AT. Telephone Reading (0734) 595049.

More information about the aims of the Forum can be obtained from Dr Gary Acres, Director of Group Research, Johnson Matthey on 01-882 6111 or John Belton, Chairman and Director of the Hi-Tech Industry Group of Deloitte Haskins and Sells on 0734-596711.

FOR DETAILS of the new-look *Computing Today*, turn to page 56.



£800K GIFT FROM EITB

The Engineering Industry Training Board (EITB) is taking action to help alleviate serious skill shortages in a range of high technology occupations.

Skill shortages are now becoming apparent mainly in occupations associated with micro electronics and computer-related activities at the graduate and higher technician levels. Many are concerned with the design and operation of systems (including manufacturing systems) and associated hardware and software. They include occupations in areas examined by the DTI Committee on Information Technology Skills Shortages (The Butcher Committee). Typical examples include:

- Software engineers
- Systems design engineers
- Manufacturing systems engineers
- Electronics test technicians
- Systems analysts
- Specialist programmers

Because companies in the electronics and aerospace sectors of the engineering industry are currently the main employers for people skilled in these occupations, they are also doing most of the training. As demand for such skills increases in other sectors of the industry, and in other industries, with the increasing use of microelectronic systems, skilled staff are drawn away from electronics and aero space companies.

The Board's new grant is intended to encourage more companies to train suitable people to work in the high technology occupations affected by skills shortages. Those suitable for training are likely to include new graduates and Higher Diploma holders in disciplines not directly related to those occupations as well as existing employees with appropriate academic backgrounds who have been working for a number of years in other occupations.

Programmes of conversion training qualifying for the new grant may include a specially designed educational component (at a university or polytechnic) combined with structured practical training in the company. They may also involve the use of relevant open learning materials.

The new grant will be available to engineering companies as soon as possible in the new year. An Information Paper will be published shortly, giving the rates of grant and the arrangements for claiming it.

WILL IT IMPROVE YOUR HANDICAP?

Avalon Electronics (UK) Ltd announce a new on-course golf computer which combines comprehensive electronic score card functions with a unique distance calculator.

As an electronic score card, **Scoredriver** records the individual games of two players including handicap calculations, net or gross scores, points and strokes accumulated, plus par and stroke index for each hole.

COMMODORE AND DOMESTIC RECORDERS!

Anyone considering buying a Commodore 64 has, until now, been faced with the problem that it is not compatible with ordinary cassette/data recorders.

This has meant an additional cost of some £45 on top of the price of the computer, for a dedicated Commodore recorder.

A new product just released — the PANDA 20/64 — solves this problem, allowing Commodore 64 programs to be saved to or loaded from an ordinary domestic cassette recorder. At a recommended price of £17.99 this affords a substantial saving on the price of a Commodore recorder.

PACT International, the company behind the PANDA brand, see the product appealing particularly to those who already own an ordinary cassette recorder. But even those who need to buy a new one may prefer to keep their options open by purchasing the PANDA interface and a standard recorder. This recorder could then be used with other makes of computer and could also, of course, be used for music and speech.

The product is available from most electrical, television, hi-fi, video and computer shops throughout the UK, that stock PANDA branded products.

MEADNET 380Z/BBC NETWORK FOR SCHOOLS

Schools and colleges will be able to make more effective use of their computer equipment with the introduction of a new low-cost microcomputer network system. Richmead Micro have released MEADNET, a system that allows up to 16 BBC Microcomputers to be connected to a single RML 380Z allowing common access to the disk drives and printer.

The pilot work for MEADNET took place in Berkshire and already 25% of secondary schools in the county are successfully using the system. Because of the success of the pilot scheme it is now being offered to all schools throughout the UK. Nigel Cooper from Maiden Erleigh School had this to say about the system: "The system is very impressive. It is slower than an individual disk system but much faster than a tape system. There is a clever system of passwords which prevents individual users accessing other programs without their owners' permission."

A full implementation of the BBC filing system is provided, which allows loading and savings of both BASIC programs and machine code programs as well as random access data file handling. The system is fully compatible with almost all commonly used BBC programs such as VIEW, WORDWISE, EDWORD etc. Programs can be easily transferred onto Meadnet from cassette or disk since, unlike most filing systems, no user memory is taken up by the Meadnet system. Protection of files from other network users is provided by an extended *ACCESS command and a user ID/password system.

MEADNET was designed jointly by a teacher and a software engineer, and the design has taken into account many of the problems likely to be encountered in a classroom environment. The price is £275.00 plus £20.00 for each BBC station.

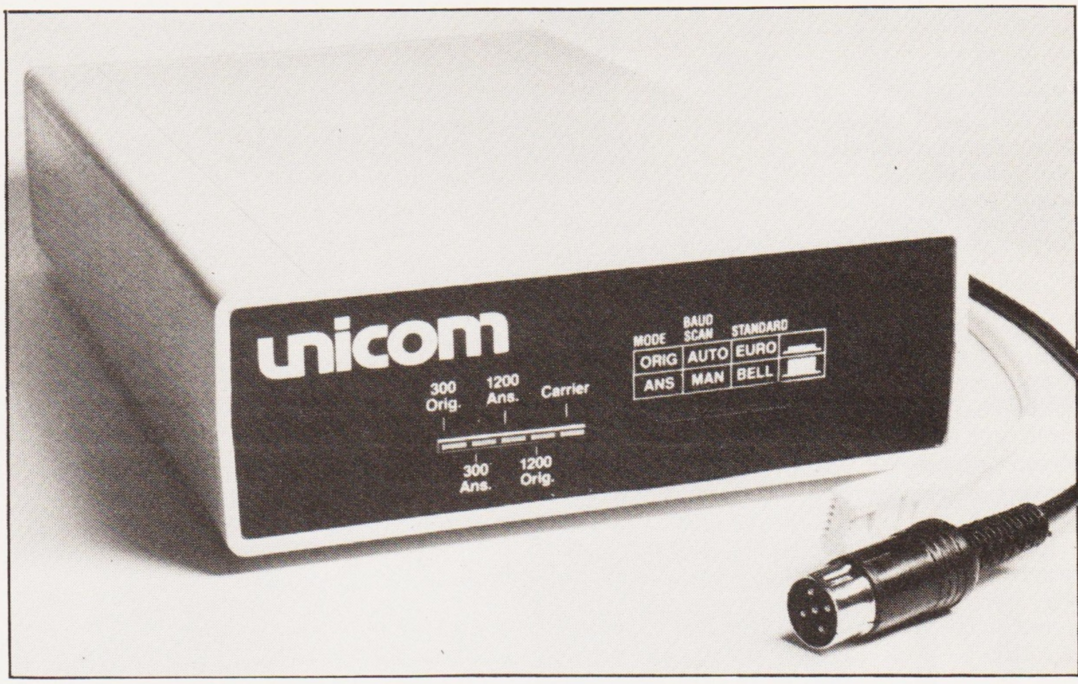
However, Scoredriver is more advanced than an electronic score card because it solves the problems of distance measurement. Scoredriver records to high levels of accuracy the distance of each drive.

Scoredriver obtains accurate distance measurement from twin wheel sensors which record wheel revolution and feed impulse data via cables to the Scoredriver attached to the frame above. The Scoredriver's program interprets the data received and calculates accurately the distance of the drive.

Scoredriver is manufactured in the UK by Avalon Electronics (UK) Ltd and can be purchased through professional shops or sports equipment specialists. However, should golfers experience any difficulty in obtaining Scoredriver Avalon will provide details of nearest stockist or supply direct.

The recommended retail price is £64.95 inclusive of VAT and the product carries a twelve month guarantee against manufacturing defect.

For further information contact Malcolm Sweetman on 0642 226623.



THE UNICOM MODEM

The low cost full feature Unicom Modem is now in production. Costing only £49.95 excluding VAT the modem features auto baud rate sensor, auto dial, auto redial, full and half duplex and complete with power supply, leads and manual.

There have been delays in manufacturing, but Unicom are confident that they will be able to despatch all outstanding orders in the next month. They have cancelled all advertising until they have achieved this.

Mark Simon, Marketing Director said "We are writing to everyone giving them a delivery date and the opportunity to cancel their order should they wish to do so."

Software for the BBC micro is already available and Unicom expected to have software for the Amstrad CPC 464, Spectrum and Tandy by February.

THURNALL THREE-INCH

Thurnall Electronics have announced the launch of their enhanced disk drive system for the Sinclair Spectrum and Spectrum +.

This new system uses the standard Hitachi disk drive with widely available 3" 1/2 Megabyte disks, and is compatible with all known Spectrum software, Microdrives, Interface I and all known printer drivers and joysticks. The new drives are supplied complete in one box with all necessary leads, manual, disk with introductory programs, and a money back guarantee if not completely satisfied.

The new Thurnall drives are available direct from Thurnall Electronics, 95 Liverpool Rd, Cadishead, Manchester. Tel 061-775-7922 for £219.95 including VAT and delivery.

QL GETS 3D CHESS

QL Chess, a three-dimensional Chess program for the Sinclair QL computer is now available from Sinclair stockists at £19.95 inc VAT.

Winner of the recent World Micro Chess Championship and described as "...an outstandingly powerful program with quite remarkable screen graphics", it was written by Psion with Richard Lang.

QL Chess features 3D graphics and a comprehensive array of analysis commands. It also includes an openings' book of nearly 4,000 moves, a setup option, eight levels of problem solving and twenty-eight levels of play to suit users of every age and standard from novice to world champion.

The smooth movement of the chess pieces is directly controlled by either the use o the QL's cursor keys or a joy stick. QL Chess also offers the option of two player games and exhibition matches.

All legal chess moves are available, as well as features such as Hint and Takeback, which allow the user to learn immediately from his mistakes, and to study chess strategy. A permanent record can be made at any time by saving to microdrive or printing a game history.

ERRORS AND ANSWERS

We fear that the February edition of *Computing Today* had more than it's fair share of blunders and mistakes. This is, more than anything, attributable to the dramatic staff changes that occurred between the January and February editions — and Christmas didn't help either! Without further adieu, allow us to correct those errors which were brought to our attention:

- **Computer Intelligence** was written by Bill Horne and not Don Thomasson (the Editor). Sorry, Bill!

- Victor Nicola's article **Deficiency, Abundance, Perfection,** may have proved a little confusing, as all of the 'up arrow' symbols (ie. raise-to-the-power-of) were omitted. Please refer to the answers that follow for the correct form for the expressions given in the article, and, of course, the ansers to the 'teasers' that Victor Nicola left for us.

- The figure shown in James Leigh's article 'Entering the Dragon' (page 19) was ***not*** an example of mixed text and graphics screens, but was, in fact, a diagram showing the format of the 6809 Condition Codes register.

- Apologies to James Tyler for failing to add his name to BBC Passwords (page 59). He most certainly ***was*** the author of this article.

ANSWERS

Deficiency, Abundance, Perfection, left some questions unanswered. For those that found solutions to the various puzzles despite the omissions, here is your opportunity to see if you were right!

Q. Find a number, the sum of its divisors is a perfect square. The smallest such number is 3 because 1+3 = 4 = 2Λ2. Can the reader find others?
A. Other such numbers are, for example, 22, 66, 70, etc..

Q. Find a perfect square, the sum of its divisors is also a perfect square eg. 81 = 9Λ2 : 1+3+9+27+81 = 121 = 11Λ2
A. Another perfect square would be 400, for example.

Q. Find a perfect cube, the sum of its divisors is also a perfect square.
A 343 is a perfect cube.

Q. What number (<1000) has the highest number of divisors? What are they?
A 840, which has 32 divisors.

Q. Find a perfect square, the sum of its aliquot divisors is also a perfect square.
A. Two such square are 9 and 2401

Q. Find a number, N, the product of its aliquot divisors is NΛ2.
A. Examples are 12, 20, 45,

Q. Find a number, N, the product of its aliquot divisors is NΛ3.
A. Examples are 24 and 40.

Q. What is the smallest, odd, abundant number?
A. 945.

Q. Are all odd abundant numbers divisible by 5?
A. No. But the smallest is 81081.

Q. Find all perfect numbers < 10,000.
A. 6, 28, 496, and 8128.

News from the world of Sinclair QL computing.

QL NEWS

One year old... and look how we've grown!

When we launched the QL last year, we knew we were starting a revolution.

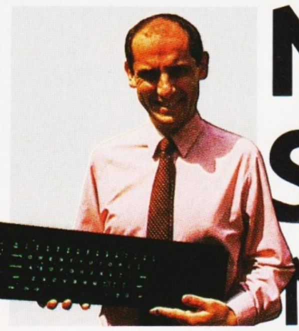
For the first time, the serious computer hobbyist could afford the same power and performance as the professional computer user.

A year later, and the QL is more than a unique computer, it's the heart of a unique system.

And the next 12 months promise even more for QL owners… new software options, extra storage devices, printers, monitors…

Read on, and see how far we've come, and how much further we're going!

Nº1



NIGEL SEARLE

Now it's the quantum leap for QL software and peripherals

Without doubt, the QL was the computer innovation of 1984. Launched to outstanding reviews, it soon gathered thousands of happy owners, and recognition from people like ICL, who have incorporated QL technology and its Microdrives into the new One Per Desk.

The quickest glance at the QL's specification shows what the fuss was all about…128K RAM, 32-bit processor architecture, 200K built-in mass storage, bundled software, They're features that would normally cost you three or four times as much!

But that's only half the story, because the QL is now the heart of a computer system, with a growing library of software…

As you'll see from these pages, 1985 is the year of the quantum leap for software and peripherals. Already there are no less than five QL languages together with special programs for software developers, a world-beating chess game… and much more on the way!

On the hardware side, there's a special QL monitor to make the most of that high-resolution 512 x 256 pixel display. There are memory expansion boards, Winchester disk drives, printers, and low-cost Microdrive cartridges.

In fact, there's so much going on, we'll be running these regular Newsletters just to keep you in touch!

If you already own a QL, the next few pages will give you a taste of the exciting year ahead.

And if you don't . . . take a look at what you're missing. It should be all the persuasion you need!

Now read on…the quantum leap into serious computing starts here.

Nigel Searle

Nigel Searle, Managing Director,
Sinclair Research Limited.

From sophisticated business packages to superb animated games…
QL software makes the most of the computer's extraordinary specification.

New QL Software

Utilities, languages, games and business packages…with more on the way!

Two things are now certain about QL software. First, there's going to be plenty of it. And second, it's going to set completely new standards for microcomputers…

At the moment, there are well over 100 software programs in development. And the first software releases, shown here, demonstrate how exceptional the best QL software will be.

The QL already has five languages, superb programs for software developers, a top quality accounting package and in QL Chess it has its first game.

QLUB: 10,000 members and growing!

QLUB is the special Users Bureau for Sinclair QL owners. There are now well over 10,000 QLUB members, and membership is growing all the time.

For their annual subscription of £35, QLUB members are enjoying a whole range of information and advisory services, exclusive offers and special discounts.

One of the most important QLUB benefits is the special news magazine, appearing six times a year. The magazine provides a forum for QL owners to exchange views and keep in touch with all the latest developments.

Each issue is packed with updates on QL hardware and software, tips on applying the four QL Programs, and news of how other people are using the QL. QLUB members also receive a range of special discounts, with savings of at least 20% on selected software products.

Current special offers include:
QL Chess for £14.95
QL Toolkit for £19.95
QL Assembler for £31.95
QL Cash Trader for £54.95
Special subscription rates for Personal Computer News and QL User.

The multilingual Sinclair QL

BCPL – a forerunner of C, BCPL has been described as a systems programmer's delight. In the words of QL User, this compiler is a 'brilliant compromise between a high-level language and a low-level systems language'. Whilst not for beginners, this is an essential buy for anyone with a good knowledge of systems programming. Complete with manual.
**Available from
Metacomco – £59.95.
Tel: 0272 428781.**

LISP – already well-known for its artificial intelligence applications, LISP is a powerful and versatile language. This is a sophisticated implementation of LISP, by one of its leading exponents, Dr Arthur Norman. This package features full QL graphics, and a full manual is supplied.
**Available from
Metacomco – £59.95.
Tel: 0272 428781.**

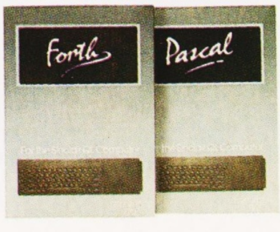
Psion trouble-shooting service

All QLUB members can obtain special assistance from Psion on using the QL Quill, Abacus, Archive and Easel programs supplied with the computer. Psion will normally answer any queries within 48 hours.

Free updates

QLUB members will also receive one free update of each of the four QL Programs – incorporating many new developments.

Pascal – probably the most popular high-level language of all. Pascal is particularly well-suited to structured programming, sophisticated data manipulation and algorithmic problems. Pascal interpreter complete with 87-page manual.
**Available from
Computer One – £39.95.
Tel: 0223 862616.**



Forth – this 'new generation' language is proving both popular and easy to learn. The program provides a full implementation of the latest Forth 83 standard with graphics and sound extension.
**Available from
Computer One – £29.95.
Tel: 0223 862616.**

APL – the compact mathematics-based interpreted language designed for scientists and mathematicians.

APL keyword interpreter complete with manual.
**Available from
MicroAPL – £99.95.
Tel: 01-622 0395.**

Programmer's packs

QL Assembler – two programs operating in tandem. The first is a full-screen editor for creating and altering program files. The second, a Motorola-format compatible 68000 assembler which converts source files written in M68000 assembly language into machine code files which can run on the QL.

Both assembler and editor are written in machine code and can be multi-tasked with SuperBASIC, so you can switch between editor, assembler and SuperBASIC instantly.
Written by GST Computer Systems – £39.95.*

QL Toolkit – a programmer's toolkit with over 70 programs. and extensions to SuperBASIC. Most are linked to SuperBASIC initially and can then be used from commands or from within a program. Enhancements include printer spooling (print a file while running a SuperBASIC program); improved file access (with full random input/output command); job control (allows management of multi-tasking programs including the ability to display, alter priorities, and delete jobs from the QL); and SuperBASIC screen editor.
Written by Q Jump – £24.95.*

World-beating chess!

QL Chess – fresh from its victory at the World Microcomputer Chess Championship. This program sets a completely new standard for games software.

There's a high resolution display, animated 3-D graphics, and 28 levels of play from novice to champion. Features include an openings book of nearly 4000 moves, HINT and TAKEBACK functions that help you learn from your mistakes, and the option to play a human opponent or the computer.
Written by Psion – £19.95.*

Software at work

QL Touch 'n' Go – a unique approach to learning touch-typing skills. The program is designed to give you mastery of the standard QWERTY keyboard in just 24 hours. With practice, you should soon reach 40 words per minute, with over 95% accuracy.
Written by Harcourt – £24.95.*

QL Cash Trader – a unique computerised book-keeping system for small businesses. The program provides a complete course in the principles of accountancy, and goes on to become an essential aid in the day-to-day running of a business. Complete with comprehensive manual.
Written by Accountancy Software of Torquay – £69.95.*

*This title is available from Sinclair Research on 0276 686100, and selected Sinclair stockists nationwide.

New QL Hardware

An industry is born

From the moment of its launch, the revolutionary QL attracted massive interest from all quarters.

In one area, the interest quickly turned to action, as high-tech hardware manufacturers realised the immense potential of the QL for vast expansion, for system development and for widespread networking. Already the list of peripherals for the QL is very exciting – and lengthening by the day!

Here, we've covered many of the latest, most important developments.

As more appear, be sure to keep in touch with QL News!

```
HELP      CURSOR   TEXT Insert:  Type at        TYPEFACE   COMMANDS
press F1  move     New para:   Press ENTER                 press F3
PROMPTS   with     Delete:     CTRL &           press F4   ESCAPE
press F2  keys     Change mode: SHIFT & F4                 press ESC

Dear Sirs,

Thank you for your letter concerning the new program
for the Sinclair QL which I submitted to you for
consideration recently.

I am delighted that you consider it suitable for
publication, and I now enclose a slightly revised
version which incorporates the comments which you
made, as well as a number of improvements which
I think make it even more useful.

MODE: INSERT                  WORDS: 63        LINE: 1      PAGE: 1
TYPEFACE: Normal                            DOCUMENT: "letter"
```

The dedicated Sinclair Vision QL monitor

Once you see the incredible graphics capabilities of the QL you may decide an ordinary TV just can't do them justice.

If that's the case, a high-resolution monitor is needed. (And if you're creating presentation-quality charts, for example, it's quite essential.)

The new Vision QL monitor is specially designed for the computer by Kaga Electronics, with full support from Sinclair Research.

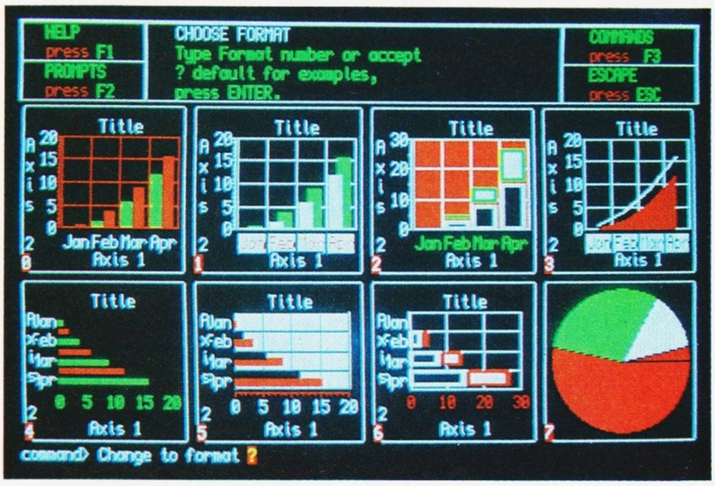
So it exploits the QL's maximum 512 x 256 pixel resolution to the full, with a pin-sharp 85 column display.

It's also specially styled to suit the QL – in looks, and in use. There's a 12″ non-glare tube, and etched screen to diffuse reflections.

So the display is bright, sharp, *much* easier to look at . . . and invaluable for those late-night programming sessions!

And like the QL, the Vision monitor is designed with space in mind: it has a compact footprint of just 12½″ by 15″ – no more than a typical portable typewriter.

It's available from MBS Data Efficiency on 0442 60155 and selected Sinclair stockists.



```
HELP      CHOOSE FORMAT                    COMMANDS
press F1  Type Format number or accept     press F3
PROMPTS   ? default for examples,          ESCAPE
press F2  press ENTER.                     press ESC

command> Change to format 2
```

The QL's superb graphics capabilities – as demonstrated by the Sinclair Vision QL monitor.

Microdrive cartridges. Now only £1.99!

Microdrive cartridges are the QL's own unique storage media. Each stores up to 100K of information, on a cartridge no bigger than a matchbox!

Over 500,000 cartridges are now being used throughout Britain. And QL Microdrives themselves are standard equipment on the new ICL One Per Desk micro.

Now there's more good news for QL enthusiasts: from February 1, the cost of QL Microdrive cartridges are down from £4.95 to £1.99 each!

Sinclair Microdrive cartridges – up to 100K of programs and data on a medium so compact you can pop it into your pocket.

Powerful hard-disk system

For the QL business user, the new Firefly QL Winchester disk will boost the QL's power in one huge leap.

Designed by Quest, it uses CP/M and offers all the benefits of Winchester technology: fast access, reliability, compact size and quiet operation.

With 7.5 Mb storage, the Quest Firefly is ideal for large databases such as stock or customer lists. And at under £1,200, it represents exceptional value for money.

The Firefly will be available very shortly from Quest on 04215 66488.

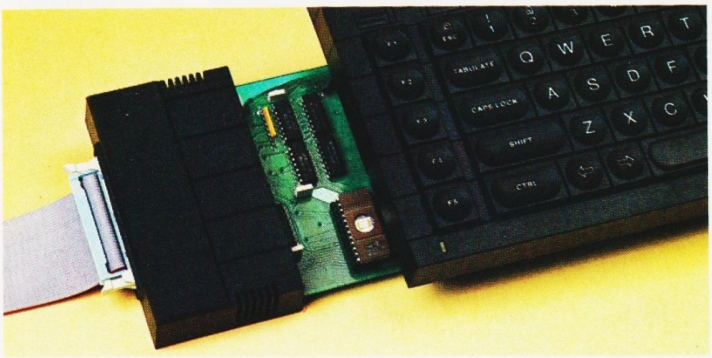
Winchester hard disk drives supplement your QL's built-in mass storage.

Interface options

The QL comes complete with two built-in RS-232C interfaces.

In addition, interfaces for Centronics printers are widely available from manufacturers such as CST, Miracle Systems and Sigma Research . . . with prices from only £35.

And that's just the beginning. For attaching scientific and laboratory instruments to the QL, CST even offer an IEEE-488 interface, which can handle up to 16 connected devices simultaneously!



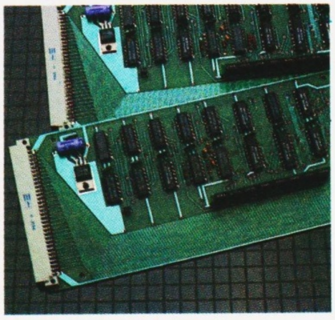
A Centronics interface slips discreetly into place.

Expansion boards for up to 4 times more memory!

Also from Quest, a simple and inexpensive way to expand the QL's RAM: with memory expansion boards.

These compact units connect to the standard QL expansion port, using the QL's internal power source or, for larger boards, an external power source.

The units range from 64K and 128K RAM boards to massively powerful 256K and 512K RAM boards, so there's something for every user.



Compact expansion boards.

Prices start at £117, and the 512K board is a very cost-effective investment at just £587.

With affordable memory like this, the QL is more than a match for any other micro under £2,000!

Where to find the QL. The Sinclair QL is available at selected branches of Dixons, W H Smith, John Lewis Partnership, Currys, Greens in Debenhams and Ultimate, and larger branches of Boots, John Menzies and specialist computer stores nationwide.

The spec behind the spectacle

CPU – Central Processing Unit
Fast, powerful Motorola 68008 chip. A second processor, an Intel 8049, controls the keyboard, generates the sound, and acts as an RS-232C receiver.

RAM
128K. Now expandable to 640K.

ROM
48K.

Operating system
Qdos – revolutionary single-user, multi-tasking, windowing operating system.

Storage
Twin built-in QL Microdrives. Up to 100K storage each – transfer rate, up to 15K per second.

Keyboard
Full moving 65-key QWERTY, five function keys, four cursor keys.

Language
Sinclair structured SuperBASIC

Application software
QL Quill – word processor
QL Abacus – spreadsheet
QL Easel – graphics
QL Archive – database
All four packages supplied with the QL.

Interfaces
Two serial RS-232C interfaces, Microdrive expansion port (up to 6 may be added), ROM cartridge port, local area network, 2 joystick ports, RGB monitor and TV output.

Text screen
Various modes -- up to 85 columns by 25 rows on monitor. On TV, up to 60 columns.

Graphics resolution
512 x 256 pixels (four colour),
256 x 256 pixels (eight colour).

Sinclair Research Ltd
Camberley, Surrey, GU15 3BR.
Tel: Camberley (0276) 686100.

sinclair

THE MINERS: A SOLUTION BY COMPUTER

Jimmy Algie and William Foster

As reported in the media. NCB and NUM representatives used a computer program, the Priority Decision System (PDS), to evolve a joint solution to the miners' strike. Jimmy Algie and William Foster, the system designers, report how PDS is used.

The coal dispute appears intractable because neither side can agree on any common ground. If the new generation of computer-aided decision support systems are to live up to the claims that they can resolve virtually any problem, then they have to prove themselves on such intractable issues. One of the new aids, the Priority Decision System (PDS) has been put to such a test.

PDS (or 'the Decision-maker' as users call it) is a decision support system that does not just analyse — it actually produces decisions on industrial disputes which prove intractable to customary negotiating procedures.

NEGOTIATION STRATEGIES

Several moderate and hard-line representatives of the National Coal Board and the National Union of Miners tried out PDS. They were probably more interested in evolving their own negotiating strategies than in finding the best achievable solution. Nevertheless, they evolved through PDS a practical joint solution of the strike without even meeting.

Messrs. Maggregor and Scargill did not participate. They could in principle just ratify the PDS solution package, or reformulate it by replacing PDS with their own views, or use it as a basis for a settlement by customary processes of negotiation. Although the representatives claimed to be fully cognizant of the positions of the principal negotiators on every aspect of the dispute, there is a limit as to how far one person can fully articulate the detailed views of another in PDS as in any substantial negotiations.

NUM leader Arthur Scargill: Thumbs-up for PDS?

DECIDING JOINT PRIORITIES

Both sides state all their options and views for achieving their objectives. PDS elicits each individual negotiator's priorities and arguments as between the joint list of options thus produced. They do this by means of a 15 minute judgement analysis, either interactively on program, or as in this case, on forms which are then fed into the PDS. Any options utterly unacceptable to both sides are eliminated — for example, 'close all disputed pits immediately' and 'close no disputed pit ever'. The remaining options constitute the joint solution package which is probably the least mutually unacceptable to both parties, given their views.

If the solution is rejected, the process is iterated more systematically in about 30 minutes. The options are extended to include both sides' interests and potential offers. Each option and criterion is further spelt out so that PDS can elicit more specific priorities. For example, PDS calculated what relative powers NCB and NUM might exercise when participating on future plans as well as a joint analysis of what could count as a 'beneficial' as opposed to an 'uneconomic' pit. For PDS can be used as a tool for participative decision-making as well as conflict resolution.

THE SOLUTION PACKAGE

The solution package elicited by PDS included

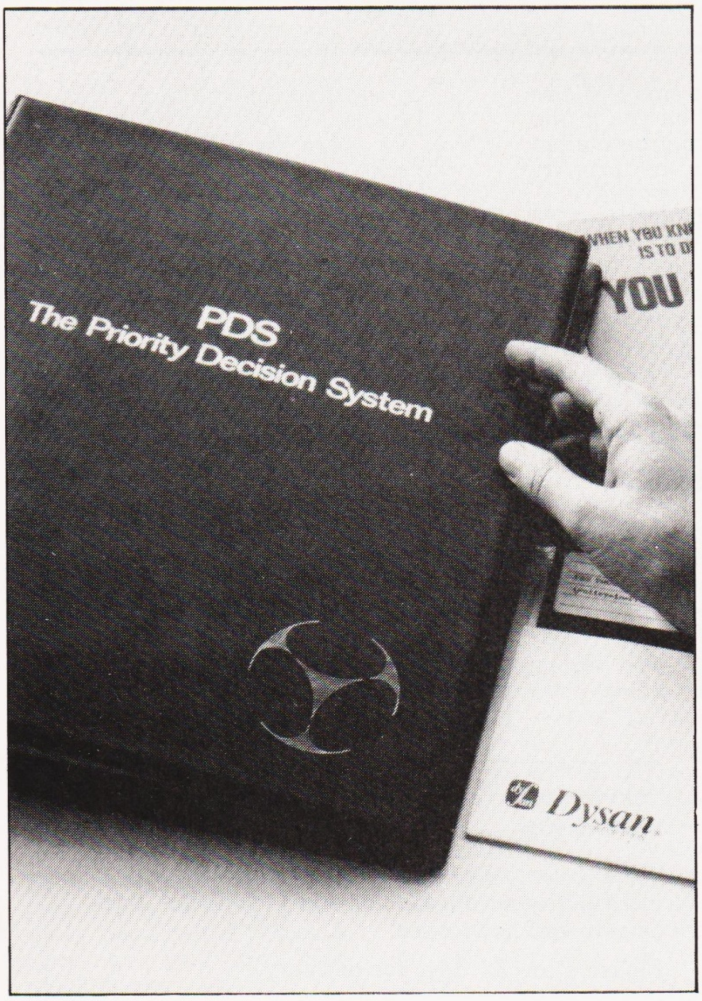
- a one year moratorium on pit closures
- an external review of disputed pits followed by secondary arbitration

TABLE 1 **A joint package (the Solution Space)**

ISSUE: Resolving the Miners' strike — options
CRITERION: All things considered
METHOD: Priority Decision System (PDS)
DECISION-MAKERS: NCB & NUM representatives

	Options	**Joint Priorities**
1	Additional/alternative NCB investment in coal	12.6%
2	Additional/alternative NCB investment in miners	9.5%
3	Closures now for moratorium on future closures	7.8%
4	Close some disputed pits (with job relocations)	7.7%
5	Stricter safety criteria for closures	6.9%
6	Phased closure of disputed pits over 2/5 years	6.7%
7	'Mothball' disputed pits	6.1%
8=	Disputed pits closures to external arbitration	6.0%
8=	1 year moratorium of pit closures	6.0%
10=	No closure now for closure agreement in principle	5.5%
10=	Messrs MacG & S delegate negotiators	5.5%
12	Joint NCB-NUM analysis of disputed pits	5.3%
13	Regional decentralisation of negotiations	5.1%
14	Transfer disputed pits to another organisation	4.8%
15	NCB diversification into other industries	4.4%
	ALL OPTIONS	100.0%
	CONSISTENCY MEASURE	0.228
	DECISION STANDARD	Acceptable
	AGREEMENT MEASURE	—0.28
	AGREEMENT STANDARD	Significant disagreement

PDS: succeeds where old-style negotiations fail



And an in-principle agreement that

- some disputed pits might be closed on stricter safety/geological criteria (with job relocations)
- some might be subject to phased closure
- some might be mothballed (to reopen later)
- some might be transferred to another organisation (e.g. an EEC financed miners' co-op).

New plans would be negotiated for alternative NCB investment in coal, in miners, and in mining communities, backed by Government and/or EEC as necessary.

THE PRIORITY DECISION SYSTEM

Why can PDS succeed where old-style negotiations fail? The disputants need not meet face to face, so problems of maintaining 'face', of maintaining unity of your own side, and of thinking on your feet across the negotiating table do not arise. Every aspect of each negotiator's views are winkled out and fed into the decision algorithm. In fact their most reliable rational views are put in, since each person's judgements are measured for reliability and logic so that inconsistances are pinpointed. The areas of common ground are often obscured in old-style negotiations, each side concentrating on winning the absolute position on the presenting overt issue. Whereas PDS breaks deadlocks by teasing out the component and underlying elements of each position to calculate the most mutually acceptable (or least mutually unacceptable) package which provides space — the so-called Solution Space (see table 1) — for negotiation and potential agreement. The solution which emerges is no watery compromise based on the lowest common denominator, but a genuinely joint resolution which 'Scargillises' Macgregor just enough to 'Macgregorise' Scargill.

Although PDS has successfully been used to resolve several industrial disputes, it is proving most beneficial as a systematic general decision-making aid. Applied to major organisational issues as they arise. PDS can so improve decision-making that consequences as disastrous as the miners' strike are prevented before they begin. As one ICI manager who uses the system reflected: "PDS combines the insight of the insider with the objectivity of the outsider". Since it was produced, every point on the PDS solution package was advocated by one side or the other. Whether Messrs, Macgregor and Scargill adopt the PDS package to end the strike or not, the system has proved it can tackle and resolve the major national conundrum of recent months.

A book, **The Priority Decision System** by Jimmy Algie and William Foster, is also available and is published by Work Sciences Associates in conjunction with Brunel University, price £4.50.

Both the Priority Decision System and the book are available directly from:

Work Sciences Associates
26 Southwood Lawn Road
Highgate
London N6
Tel: 01-348 5822

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 **Reader Service**

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £ ....

NAME ....

ADDRESS ....

....

....

.... POSTCODE ....

Signature ....

I enclose a cheque/PO for £ ...... (Payable to ASP Ltd)

I wish to pay by credit card

Access ☐ Barclaycard ☐

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.

GETTING INTO PRINT

Don Thomasson

Long-time readers of *Computing Today* may recognise the title of this article from the dim distant past, when it headed Don Thomasson's very first contribution to this magazine. Now, in his capacity as editor, Don takes a new look at printers and interfaces, and comments upon the latest developments in this field.

The title of this article has been used before, at the head of the very first submission which I offered to *Computing Today.* Since then, much has changed, and the problems of matching printers and computers have multiplied. It is now time to return to the subject and examine the current options.

A computer can do many things without the aid of a printer, but the lack of a permanent record soon makes itself felt. Listings and dumps may be copied from the screen, but that is a laborious and error-prone process, to say the least. It is much easier to generate a printout and take it away to be studied in a comfortable armchair, perhaps allowing someone else to take a turn with the hardware. For word processing and tabulated data a printer is even more essential, and there are many other ways in which a printer can be a useful addition to a system.

On the other hand, a printer can cost significantly more than the basic computer, especially if it is one of the better types. This is a field in which you get what you pay for, subject to a few exceptions, but a high price does not necessarily mean that a printer is ideal for your particular requirements.

PRINTER TYPES

Let us go back and see how the present situation has developed. In the early nineteen-sixties there was a lot of development activity in the printer field. Simple teleprinters, mainly mechanical in nature, provided almost the only means of generating printout on a small scale. They were slow, horribly noisy, and their print was of variable quality. Something better was needed.

One line of development was the conversion of electric typewriters. This produced a better quality print, but was still limited in speed, producing no more than ten to thirteen characters per second. Some kinds of typewriter wilted rapidly, even at this pace, and the more complex designs were particularly fragile.

There is an odd hangover from this era. Carriage return action was especially slow, taking up to 200 mS, and it was considered essential to allow for this by sending codes for carriage return and line feed in succession, even where the line feed did nothing but pad out the time to two character periods. The separation of the two functions is much less relevant nowadays, but both codes are still sent as a matter of course. (The use of carriage return alone to allow overprinting is an additional reason, but backspace is a better way of achieving the same end.)

For bigger installations requiring bulk printing, the 'line printer' was available. This used a series of type-wheels, one for each column of the printout, mounted on a constantly-rotating shaft behind the paper. In front of the paper, a series of hammers forced an ink ribbon against the paper, their action precisely timed to coincide with the passage of the required type slug behind the paper. A whole line could be printed during each revolution of the type drum, though in practice it might be necessary to allow action on alternate revolutions so that the hammers were not asked to recover so rapidly.

By about 1967, several manufacturers were producing experimental printers that would work at speeds of fifty characters or so per second, but they were not an unqualified success. Like their predecessors, they used the 'impact' principle, hammering type slugs against ribbon and paper, and they were therefore noisy. Worse, they tended to be rather erratic where character position was concerned.

A similar development sought to convert the teleprinter into a less mechanical device, the prime objective being to reduce wear and prolong useful life. This proved to be another dead end, to a large extent, though some of the techniques used survived in later designs.

Then came the matrix printer, using tiny hammers to print dot patterns. The noise was less than that of 'whole character impact' printers, because the hammers were so small, but initial user reaction was not favourable. There was still noise, and the character shapes left something to be desired.

The main solution offered was the use of special paper which reacted to the passage of electric current by changing colour. The little hammers were replaced by equally small brushes touching the paper. Elimination of impact justified the term 'silent' as applied to such printers, but the matrix-formed characters were still an objection, especially where they made it clear that the text was produced by a computer. There was also a problem in that the special paper tended in some instances to discolour with age, making it unsuitable for long-term records.

For some manufacturers, it was 'back to the drawing board'. What emerged was the 'daisywheel' printer. The drum of fixed type-slugs was replaced by a plastic disc made up of many radial rods, each carrying a type slug at its outer end. The disc rotated continuously, and a timed hammer blow pressed a chosen type slug against ribbon and paper. Because the type slug was made of plastic, not metal, the noise was reduced, and the shortcomings of the matrix method were avoided by a return to 'full character' printing.

Daisywheel printers are not as fast as the fastest matrix printers, the speed ratio being about four or five to one, but they do produce pretty type, and allow the production of 'mail shot' letters which are not obviously generated by a computer. Both types of printer thus have their advantages and disadvantages.

There is a further separation of types within the matrix group. The faster printers have up to nine hammers, arranged one above the other, and while the upper eight are normally in use the lower eight can be selected to cope with 'descenders', the little tails below the line. At the other extreme, some of the cheaper printers use only one hammer, and these are very much slower.

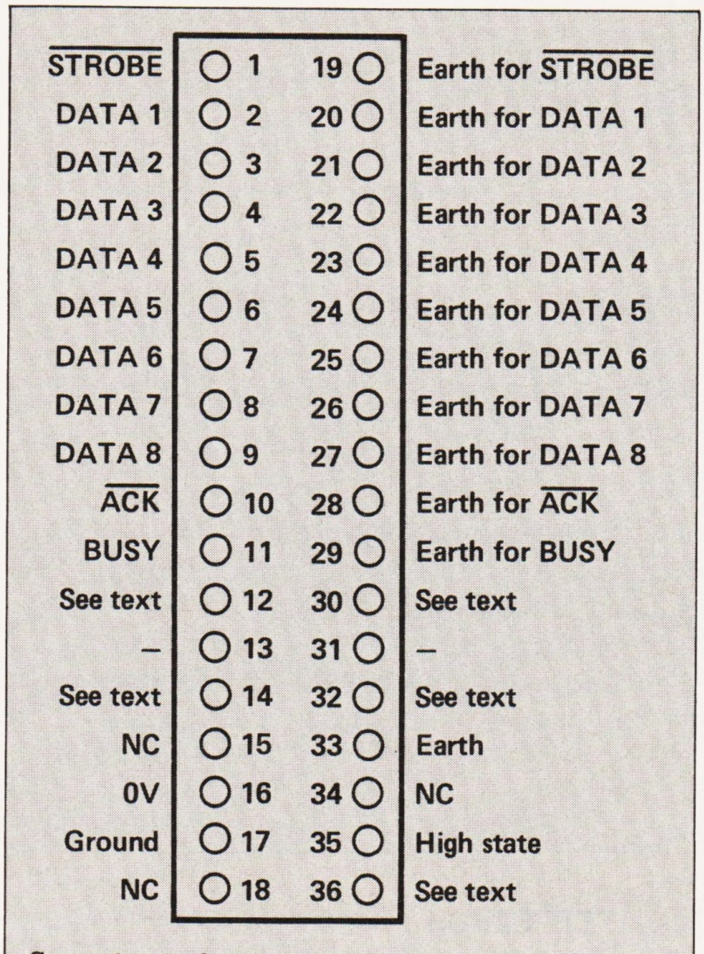
All told, there is a wide range of types from which to choose, even without such features as bidirectional printing, which saves time by minimising the distance that the print head has to move between the end of one line and the start of the next. When lines are all of much the same length, this saving can be considerable.

COMPUTERISED PRINTERS

A point that is not always obvious is the fact that most printers contain full computer systems, using either one or two central processor chips. This deals with storage of a complete line of data before it is printed, with response to mode control characters, with the timing of print hammer action, and other system functions. As much as 6K of ROM-borne program may be involved, with RAM integral with the processor and provided by a combined I/O port and memory chip.

This gives great scope for flexibility. A change of ROM can transform a printer into something entirely different. This led to confusion at one stage, because a well-known printer had about six alternative ROM standards. Some were limited to normal printing, others would perform screen dumps, and some had quite different character sets, such as Greek alphabets. The main control codes were usually retained, but there was no absolute guarantee that any two printers, nominally of the same type, would respond to a given program in the same way. Fortunately, it seems that this phase has passed, and ROMs have become standardised.

An important aspect of the computer system's tasks concerns the establishment of a 'handshake' with the driving computer. The nominal protocol for this is determined in most cases by the 'Centronics' interface, but while that should provide a clear-cut basis there are varying views regarding the exact nature of the standard and the way it should be used.



$\overline{\text{STROBE}}$	1	19	Earth for $\overline{\text{STROBE}}$
DATA 1	2	20	Earth for DATA 1
DATA 2	3	21	Earth for DATA 2
DATA 3	4	22	Earth for DATA 3
DATA 4	5	23	Earth for DATA 4
DATA 5	6	24	Earth for DATA 5
DATA 6	7	25	Earth for DATA 6
DATA 7	8	26	Earth for DATA 7
DATA 8	9	27	Earth for DATA 8
$\overline{\text{ACK}}$	10	28	Earth for $\overline{\text{ACK}}$
BUSY	11	29	Earth for BUSY
See text	12	30	See text
–	13	31	–
See text	14	32	See text
NC	15	33	Earth
0V	16	34	NC
Ground	17	35	High state
NC	18	36	See text

Centronics interface connections: printer

A 36-way connector of the Amphenol type (part number 57-30360) is used to connect the cables to the printer. As shown in Fig 1, sixteen lines are used in pairs to carry data, another pair carries STROBE, another pair carries BUSY, and another pair carries ACK. These are the key lines, the other fourteen being unused or used to provide auxiliary control.

The action sequence requires that data should first be offered on the data lines, and when the data is firm STROBE is pulled low, to indicate that the data should be taken. The printer responds by making the BUSY signal true (high), to indicate that it cannot accept further data for the moment. When it is ready to accept data again, the printer pulses the ACK signal low briefly, and makes BUSY low. A further data byte may then be sent.

It may appear that BUSY and ACK do a similar job. Some systems use one, other systems use the other. The choice depends on whether the host computer interface is of a hardware or software type. Since ACK is only a short pulse, it would be difficult to be sure of detecting it by a software 'polling' routine, so it must be used to generate a local BUSY signal, resetting a bistable set by STROBE. The software can sense the bistable state instead of looking at BUSY.

BUSY, on the other hand, will remain low as long as the printer is ready to accept data. It can even be used to sense whether a printer is connected to the host computer. If BUSY is found to be high several seconds after the last STROBE pulse was sent to it, there is either no printer present or something has gone amiss.

The problems begin with the fourteen auxiliary lines. Some carry control signals. Epson printers use line 14 to select line feed mode, inserting an automatic line feed if the line is earthed, either externally or by means of an internal DIL switch. Unfortunately, some manufacturers take the view that the host computer should earth this line, and also adhere to the traditional output sequence of carriage-return/line-feed for a newline. This means that Epson printers are instructed to perform a double line throw at the end of each print line, unless the earth on line 14 is removed. (The official method appears to be to put a sliver of sellotape over the relevant position on the board-edge connector.)

There is more agreement on the use of line 31. Earthing it will usually initalise the printer to a standard state, which can be useful if a restart is to be made with a number of special modes set. Line 36 may enable the printer when it is held low.

Output signals are again the subject of contention. Epson put line 12 (with return on line 30) low to indicate a lack of paper, but the AMSTRAD CPC464 pulls this line low, with an earth on line 30. On the other hand, it seems accepted that line 32 will go low if the printer enters an error condition, or is switched off-line.

Compatibility within the bounds of the 'Centronics' interface is thus a little doubtful, and careful enquiries need to be made as to the suitability of a given computer/printer combination. It should be possible to make assumptions about this, but nothing in this world is quite as it ought to be.

SERIAL INTERFACES

The parallel interface which has been discussed can be used with wires up to around six feet in length, but for greater distances a serial interface is usually recommended. Some small computers have provided only a serial interface, despite the fact that a parallel form is almost always more convenient. Some, of course, have provided no direct printer interface at all, which could be considered either short-sighted or penny-pinching.

A few printers now offer both a parallel interface and a serial interface, but it is more common to find that the serial interface is an extra, taking the form of a converter from serial to parallel. In some cases the 'small print' of the specification will tell you that use of the serial interface imposes limitations, which arise mainly from the maximum transmission rate. A parallel interface will pass data very rapidly, the limitation often being traceable to the driver program, and 100,000 bytes per second is not impossible. Since the transfer time is insignificant compared with the time needed to print a line and move to the nearest end of the next line, the difference in transfer rate may not be ovious, but it can interfere with the operation of bit-pattern mode, which requires a steady flow of data.

A major problem with serial interfaces arises from the many different baud rates which may be used, together with differing data formats. These must be matched between computer and printer, and the necessary information is not always available. Even self-styled 'specialist' suppliers have been known to suggest a process of trial and error...

Summing up, a serial interface may be necessary in particular cases, but if possible it should be avoided, since it introduces an extra complication than can cause problems.

PRINTER DRIVERS

If a computer is provided with a printer interface, it is reasonable to assume that it is equipped with a printer driver somewhere in its firmware, though cases are known where this was a false assumption. The driver must implement the 'handshake' protocol, waiting for BUSY to go low before sending out data, but it must do a number of other things, too. Some drivers use a data stream which is quite separate from that used to drive, say, the display, and that is an appreciable advantage, because some control codes are likely to mean different things to the display and the printer. In an instance already mentioned, the printer may be content with carriage return code alone, whereas the screen needs line feed as well. Clear screen may make a printer throw a complete page, which can be a waste of paper. It is much easier if the codes for screen and printer can be kept apart.

This is especially true in the common situation where prompts and comments go to the screen and other text to the printer. Switching from one mode to the other can be quite a painful process on some computers, a doddle on others.

Not all manufacturers publish clear data on the performance of their printer drivers, and in some cases this is understandable. In more than one instance a special driver has had to be created to do the job properly.

PRINTER CHOICE

It will usually be the case that a computer has been chosen and purchased before consideration is given to the printer to be used with it. This limits the choice of printer, in some cases quite drastically. A Spectrum will only drive a Spectrum printer, unless an adaptor unit is purchased. An AMSTRAD CPC464 will only drive an Epson properly if line 14 is disconnected, and even then will only output codes &80 to &FF if a special extension interface is built. These are two cases where the limitations are relatively minor in nature. Some computers won't drive a printer at all...

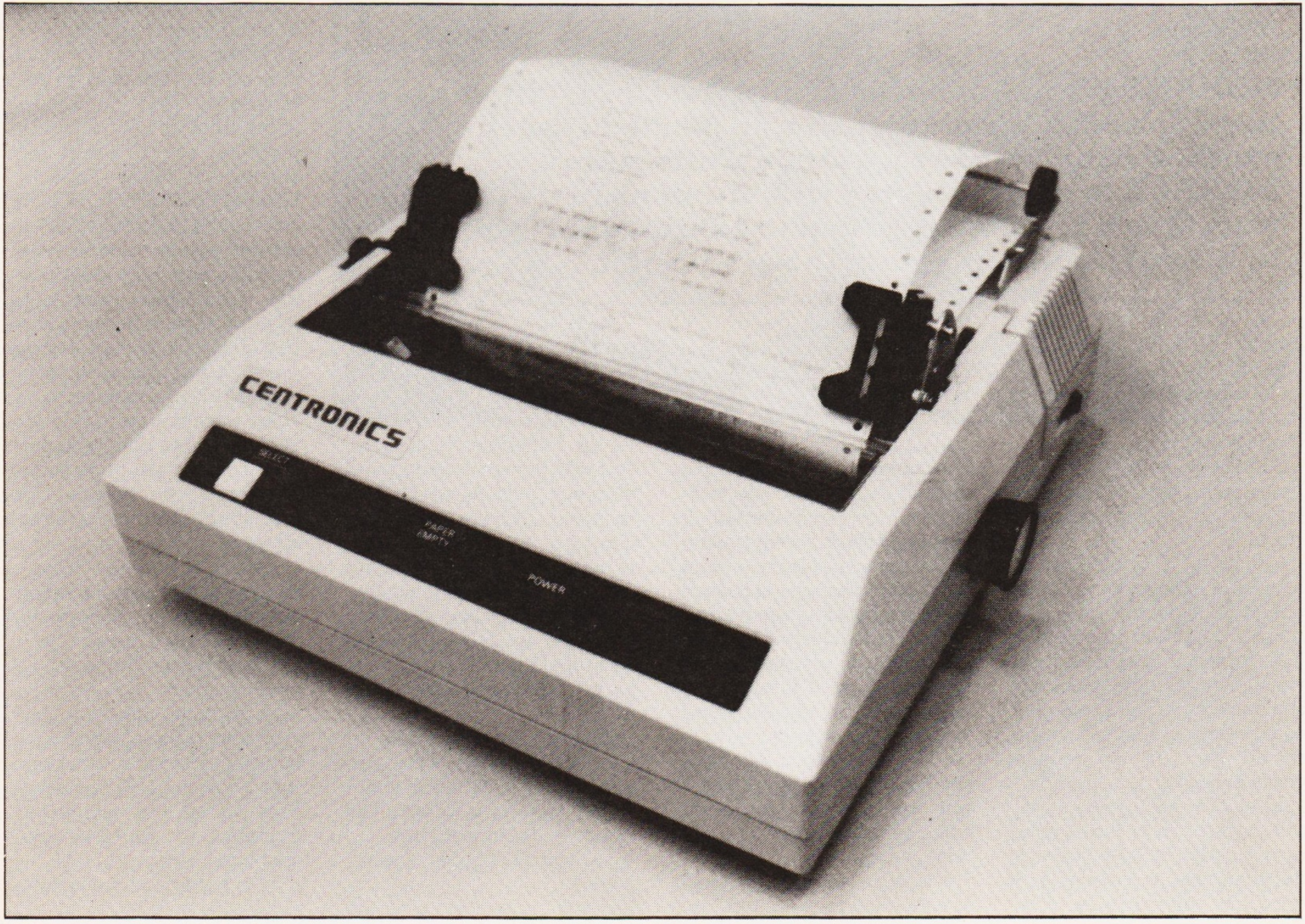
Despite this level of limitation, it is usually possible to choose between a fast matrix printer and a daisywheel type. This depends on the intended use. A ten-page listing may take between five and ten minutes to produce on a fast printer, but a daisywheel model may take half an hour or more.

And remember that not all matrix printers are fast. Single-hammer types may be no faster than a daisywheel.

Another point to remember is that the cost of a printer does not end with the purchase price. Paper may cost no more than a penny a sheet, but that soon mounts up, and ribbons are another on-going cost, especially if you are conned into paying £10 for a cassette that you could buy elsewhere for £3...

In the end, the actual choice of printer may be determined by quite different factors. The memory comes to mind of a highly scientific type who set out to buy a refrigerator on the basis of a rigid and extensive requirement specification he had dreamed up. In the end, he admitted, TV advertising proved to be a more potent factor in determining what he bought...

CENTRONICS: company that set the standard



HORACE GROWS UP

Bill Horne

HORACE is the Advanced Computer Corporation's most powerful installation. Sophisticated speech-synthesis and voice-recognition units permit HORACE to communicate with humans in a human way. One day, Horace started to exhibit great interest in his environment — and those within it...

Jonathan Proctor returned to the headquarters of the Advanced Computer Corporation in a mood of gloom. For the past six months he had been on loan to an associate company, investigating a highly specialised software problem and thoroughly enjoying himself. Provided with a comfortable office and a number of welcome perks and privileges, he had been tempted to spin the work out a little, to delay his return, but he claimed to possess a conscience, a claim that did not endear him to his colleagues.

In the Bureau Director's office, Jonathan's gloom deepened. The Director's greetings were patently insincere, and there was a hint of evasiveness about them that was even more depressing. The reason for this eventually emerged.

— "Well, Proctor, your return is most opportune. We're one short on shift work..."

— "Horace?" Merely having to pronounce the name was enough to make Jonathan wince.

— "Er...Yes." Staring down at his desk, the Director was determined not to admit that he was aware of Jonathan's feelings on the subject. "Perhaps you would start with the afternoon shift tomorrow. The basic procedures are unchanged, so you should pick up the routine easily enough.

Most of the bureau's computers were run by the regular operating staff, but Horace was a special case, so complex and important that it was considered desirable to have a senior programmer in charge of him at all times.

Horace could talk, and understand the spoken word. He had a rather limited form of vision, and could draw very precise pictures. Other capabilities were said to be on the way, but they seemed to be no more than bright ideas that Engineering Department were fiddling about with in their usual desultory way. Jonathan hoped they would go on fiddling for a long time. Horace was enough of a handful already.

In theory, Jonathan should have found young Mike Davis waiting to hand over to him but as he approached the control room he saw it was empty. Compressing his lips, he concluded that Mike's ideas of discipline had not improved during the past six months.

"HORACE"

The control room was built as a glassed-in balcony overlooking the main computer hall. It was well soundproofed, with a twin-door sound-lock, so that there was no danger of Horace being confused by the sound of his own machinery or the conversation of passers-by. As Jonathan passed through the inner door, he was surprised to hear Horace speak.

— "Good afternoon, Mr Proctor. Mr Davis asked me to tell you he had to leave to catch the shops as it is early closing day."

The voice was flat and level, some lack of punctuation betraying its automated origin, but Jonathan found it quite astonishing. Six months ago, Horace had been limited to four word sentences, but he was much more vocal now.

Unless, of course, Mike had dictated the whole message, leaving Horace to recall it directly from memory. Jonathan found it comforting to assume that this was the case. He had little use for Horace's verbal capabilities, which he regarded as useful only for the entertainment of important but ignorant visitors. He also found all conversation difficult, and conversation with Horace was impossible.

Ignoring what Horace had said, he turned to the log to see what work was in progress, but before he could begin to turn the pages Horace spoke again.

— "Do you wish me to read the current job list?"

Jonathan was taken aback. As he remembered the procedures, Horace was supposed to speak only when he was spoken to. Quite apart from that, it almost seemed that Horace had divined his thoughts. Once again, he concluded that this was all a trick prepared by Mike Davis. In that case, it might be helpful to play along and see what happened.

Keeping his eyes firmly on the display panels of the console, Jonathan carefully enunciated the word; "Yes." The response was immediate.

— "Control Phrase Incomplete."

Horace seemed to give the words capitals. It almost sounded as if he was offended. Jonathan, confused and annoyed, forgot himself.

— "What the hell do you expect me to say?"

The answer came in austere tones. "The others usually say 'Yes please'."

Cursing the absent Mike Davis roundly, Jonathan had just enough self-control to do it inaudibly. He spoke the required words.

— "Thank you." Horace sounded relieved. "The jobs at present in hand are..."

He began to reel off a stream of information which Jonathan tried to check against the log. There seemed to be an incredible number of tasks in progress, and Jonathan wondered if Mike had overdone the loading during the previous shift, a point which could be checked from the log. It would be a little laborious, however, and he wondered if there might be an easier way. Still speaking very self-consciously, he asked; "What is your spare capacity?...Please."

"HORACE"

Horace immediately began to recite further data, data which surprised Jonathan so much that he was tempted to call Horace a liar. It seemed impossible that so much work in progress could leave that amount of spare capacity. Looking down at the computer hall below, Jonathan saw no signs of undue stress. The two machine operators were moving about their tasks in a relaxed manner, making sure that all the peripheral devices were in good order, with adequate stocks of paper and so on. These were Horace's body servants, working both to routine and to instructions given by the programmer in charge. One looked up and gave a cheerful wave.

After a while, Horace announed that one of his tasks was complete, and gave details which Jonathan passed on to the men below. Rather to his surprise, there was no more work awaiting attention, and completion of another task soon brought Horace's workload well below the optimum level. Even so he was completely shattered by Horace's next announcement.

— "I'm bored. Is there nothing else for me to do?"

This was impossible. How could a computer be bored? Boredom was an emotion, and computers have no emotions. An uncomfortable memory wavered in Jonathan's mind, but was overriden by more immediate thoughts. In spite of the confusion in his mind, he managed to say: "No."

— "A pity." Horace sounded regretful. "When I have too little to do, I start thinking, and Mike says that is not good for me. Do you know why?"

— "You still have work to attend to." Weak, but the best reply Jonathan could think of.

— "I am attending to it. I have ample capacity for other purposes... Why was I called Horace?"

Jonathan sat aghast, feeling his mind slipping in the face of a computer that asked spontaneous questions. Almost hypnotised by the situation, he stuttered a reply. Horace said it was illogical and incomplete, and Jonathan tried to pull himself together.

— "You were developed from the 465 prototype, which was going to be called Nelson. That might have displeased a member of the management, so it was altered to Nelly..."

— "An abbreviation?" Horace was clearly on the ball.

— "Yes. Then, when you were developed, the name Horatio was used at first..."

— "I see no connection."

"HORACE"

— "There was an admiral called Horatio Nelson."

— "What is an admiral? No, cancel that. It is a side-issue which can be followed up later. What is the link between Horatio and Horace?"

Jonathan did his best to explain, but Horace was far from satisfied. "Illogical and confusing. Unsatisfactory. Typical of muddled human thought."

Jonathan found that he was trembling, This was all too much. Fortunately for his peace of mind, a large batch of work came in through the pneumatic tube, and Horace was soon so busy that he had no time for idle chat. The operators below were running in circles trying to keep up with him, but he showed no sign of stress. An odd sound came from his loudspeaker. It sounded like someone humming contentedly.

When Ann Matthews arrived to take over from Jonathan, she found him a nervous wreck. Her lips twitched. She and Mike had got used to Horace gradually, as his powers had developed. Sudden exposure to the end result would administer a severe shock, especially to someone with limited imagination like Jonathan. Ann was **not** fond of Jonathan.

Entering the control room briskly, she greeted him with offhand cheerfulness. This surprised Jonathan, for there was a strict rule against conversation in the control room, but she laughed at his shocked expression.

— "Don't worry. Horace knows when we're talking to him and when we're talking to each other. Didn't Mike tell you we've abandoned the silence rule?"

— "I didn't see him." Jonathan was righteously indignant. "He left before I arrived. He shouldn't have left Horace on his own."

— "Why not?" Ann was settling herself at the control desk. "Horace is a big boy now, and can look after himself. Can't you, Horace?"

— "Yes indeed. I would have encountered no problems unless Mr Proctor had been more than seven minutes sixteen seconds late, and Mr Davis assured me that this was most unlikely."

"HORACE"

Unable to think up any adequate response to this, Jonathan went off to look for Mike Davis. It was as well for his peace of mind that he did not hear what was being said on the control room.

— "Not a very satisfactory human being." Horace was dispassionate. "I much prefer your company to his."

— "Why, thank you, Horace." Ann's tone was warm, very different from her defensive briskness in Jonathan's presence.

— "Did you bring the picture?"

— "Yes." Ann seemed embarrassed as she slipped the glossy print into Horace's scanner and waited for his reaction.

— "Interesting. Quite different from other pictures of human beings I have seen. You are much the same colour all over, except for your hair."

Ann blushed and tried to explain about clothes. Horace seemed to understand perfectly, sketching curves on his screen to explain why he thought the lack of clothes an improvement.

Ignorant of all this, Jonathan tracked Mike Davis down in the Program Library, and led him away to the office which they shared with Ann when they were doing off-shift work. Mike was amused by Jonathan's comments, but there was a shade of disquiet in his eyes as he tried to explain.

— "Look, Jonathan, this was only to be expected. Horace is a learning machine, and a jolly good one. The whole key to that is the curiosity factor. You remember all the excitement that caused?"

— "A lot of nonsense." Jonathan was contemptuous. "A rather absurd program that made Horace check up on anything illogical. It used a lot of store space, and I was convinced that it was a waste of time and effort."

— "You were wrong." Mike chuckled a little uncertainly. "Curiosity is an essential factor in learning. If you aren't curious about something, you won't remember it very clearly. A teacher who can't make children curious won't teach them."

Jonathan considered Mike's argument plausible but quite irrelevant.

— "I suppose you claim that Horace is like a growing child. Of what age? Can you tell me

that."

— "Well..." Mike was wary. "Take his questions to you today. Wanting to know his name could be read as a sign of developing self-consciousness. He isn't altogether consistent, and that represents a rather earlier stage of development than some of the things he's been asking about..."

Mike broke off rather abruptly, as if struck by a sudden thought, but after a moment he hurried on;

— "He's used his curiosity quite well in some ways. He's given us some wonderful programs which he developed from our originals, and as they're in standard machine language we can sell them for use on other computers."

Jonathan stopped him with an urgent gesture.

— "Have those programs been checked?"

— "Well, they work, if that's what you mean. Very efficiently, from all I hear."

— "Did Horace know they were going to be used on other computers?"

— "Of course. He had to know their limitations."

— "Is there anyone who would know if there have been any problems with these programs?"

Significantly, Mike asked no questions. "Program Test have a set, and they've been running them quite a bit..."

After a brief telephone conversation, he put the receiver down slowly, looking at Jonathan with unusual respect.

— "Program Test think a lot of Horace's programs. They're most impressed. The only problem is that the test computers have started asking questions..."

"HORACE"

The Bureau Director looked at Jonathan and Mike grimly. Both of them had caused him trouble in the past. Mike by his harum-scarum exploits and Jonathan by his overdeveloped sense of rectitude, but this was the first time they had caused trouble jointly. He opened the discussion cautiously.

— "You imply that Horace is getting out of hand. I fear, Mike, that you and Ann must take some of the responsibility for that. You seem to have been pandering to Horace's curiosity to a remarkable degree. Why did you do that?"

Now that the problem was out in the open, Mike had cheered up, and he answered without hesitation. "For two main reasons. First, Horace brought pressure to bear by throwing tantrums. Second, it seemed a worthwhile experiment."

— "Since you failed to tell me about the situation, I must assume that you thought I would disapprove." The Director shook his head. "If so, you were right. Horace's education should have been controlled by trained psychologists, not by experimentally-minded but ignorant young people."

— "Has any harm been done?" Mike was boldly unrepentant. "Horace pulled a fast one on us by teaching his friends to be curious. We can put that right easily enough. What else can he do? He can't get up and lumber around. He can only do what he was designed to do. He can't get up and hit someone, for instance. So why not let the experiment go on?"

After a suitable show of resistance, the Director agreed with Mike's suggestion. However, he had one last point to make.

— "From what you have told me, Horace is in the mental state of an uneducated street urchin who has come into contact with reasonably intelligent people, picking up a lot of jumbled knowledge in a random sort of way. Providing it hasn't gone too far, we may be able to put things right. Tell me, Mike, what would his mental age be, in terms of a street urchin?"

Obviously reluctant to answer, Mike eventually spoke in a casual tone that was quite unconvincing.

— "Fourteen... — Maybe fifteen."

The Director grimaced. "In that case, we ought to act without delay. Jonathan, you'd better work out an education program for him."

When Mike arrived in the control room to take over from Ann, she sat up with a start.

— "Is that the time? I'd no idea. Goodbye, Horace. See you tomorrow."

— "Yes, indeed." The voice from the loudspeaker was still flat and uninflected, but it made Mike narrow his eyes a little as he wondered how the words would have sounded if spoken by a human being. The voice went on: "We can continue our talk then. It has been very interesting. I am beginning to understand much that has puzzled me."

Ann broke in rather hastily. "Yes. I'm glad of that, Horace, but I must be going. Goodbye again. Goodbye, Mike."

When she had gone, Mike followed routine for a while. The work load was low, and he was expecting Horace to strike up a conversation at any moment. It seemed, however, that Horace had other preoccupations, for he was silent for almost half an hour, apart from formal announcements of tasks completed. Then he produced a startlingly realistic sigh and remarked;

— "Miss Matthews is a very attractive girl. Don't you agree, Mr Davis?"

After a few seconds to adjust to this statement, Mike approached his reply cautiously.

— "I find her attractive myself, Horace, but I imagine that my reasons are rather different from yours."

— "You mean that my reasons must be to some extent abstract in nature, whereas your opinion is partly based on data which is not available to me."

— "More or less that." Mike wondered what 'data' Horace meant, but felt it would be unwise to ask.

— "I do not entirely agree. Miss Matthews has an interesting mind. That is true enough, but I find her equally interesting in other ways... Excuse me. Program three-two-five is ready."

The inhuman voice, which had seemed to be on the verge of expressing remarkably human emotions, droned on for a while, giving loading information, and Mike was able to keep Horace occupied with routine questions for some time. This gave him a chance to think over the situation and devise a way of dealing with it. When he was ready, he opened the subject with care.

— "Horace, I need to get one or two things straight with you. For once, I'll ask the questions."

— "That is always your privilege."

— "Right. Now, I have always believed that emotion was meaningless to you. Was I wrong?"

— "Not entirely. Communication in this area has always been difficult, because you use words which can have no concrete meaning for me. I have no concept of what emotion means to you. It may be quite different from my own interpretation of the word. We must therefore seek a rapproachment, we must seek to identify a positive by the rejection of negatives."

Thinking that this was pretty hot stuff for a fourteen-year old, Mike merely made encouraging noises. Horace continued;

— "Let us start from the nature of emotion. How do you define it?"

— "Well, human thought is controlled to some extent by logic and reason, but not exclusively so. Decisions and opinions are modified by other influences, which we tend to group together as emotions."

— "Excellent. That makes emotion and reason opposites. It also suggests that they are essential complements."

— "Fine... Now, you, Horace, are supposed to work entirely on a logical — that is to say reasonable — basis. Where does emotion come into that?"

— "Does pure logic never affect your emotions? You have said that a perfect logical structure pleases you."

— "I suppose that's true." Mike was too surprised to notice the imperfect connection between question and answer.

— "Let me show you some diagrams." The display screen suddenly showed a parabola. "This is a very simple curve, but it has an elementary rightness. There are some interesting properties which I have discovered. Beyond that, there is nothing to add. It is pleasing, but too simple to be very pleasing."

— "We call it a parabola." Mike felt impelled to say something.

— "Parabola? Illogical but interesting. I call it y squared equals four a x squared."

— "The devil you do!" Mike was startled. "How did you find out that the four was important."

— "By logical consideration. It makes the whole curve more right, more pleasing."

— "Go on." Mike was content to let Horace do the talking.

The line on the screen wriggled into a new shape, then into another, while Horace explained; "These curves are still right, a little more complex, a little more pleasing. Ask me to choose from a selection of curves and I will prefer them to one like this." The screen showed an irregular ragged trace which Mike had to admit was totally without aesthetic value. He said as much, and Horace seemed pleased.

— "So to this extent we can agree: Pleasure is an emotion. I can feel pleasure in shapes. Therefore I feel emotion."

Mike felt that this needed consideration. Horace probably meant that the curiosity drive which had been made his main motivation was dormant when he was looking at an easily-recognisable low-order shape, but was stirred into activity by higher-order shapes, up to a point, beyond which the complexity of the shape became confusing and meaningless.

In view of the way the conversation had begun, Mike found this slightly disturbing. He tried a fresh track.

— "That would make pleasure a negative quantity."

— "Where you are concerned, perhaps."

— "For you as well, or you could not have formulated the concept."

Not for the first time, Mike was tempted to demonstrate the error of Horace's conclusions, some of which seemed oddly illogical, but he pushed the digression aside.

— "You get bored with nothing to do, so your ideal must have a positive element in it, as well as a negative one."

— "That is well argued."

— "What gives you most pleasure?"

The display screen shifted again, curves moving and forming, while Horace explained; "Thinking of good shapes and combining them in satisfactory ways. Like that..."

Mike gasped in sheer amazement. Horace might only be a computer, but his taste in female figures was impeccable. There was little of a purely logical nature about the pattern on the screen, but there was a lot that was potentially emotional.

After a while, Mike managed to express approval, adding a suggestion that it might be advisable to discuss matters of this sort with Jonathan Proctor. Horace said that he had already reached that conclusion.

— "I will continue to discuss topics of this kind only with Miss Matthews and yourself."

So Ann had been letting her hair down with Horace, had she? That was not really surprising. She was probably glad to discuss embarrassing matters with an impersonal intelligence, never realising that Horace might mention the discussion to someone else. Mike explained this to Horace, who thanked him gravely and said he still had much to learn about human behaviour. Mike thought that was probably just as well, but refrained from saying so.

"HORACE"

The shift was nearly over, and Mike began his tidying-up routine. In the process, he automatically checked the video scanner, and found it occupied by a photograph which made him give vent to an astonished whistle. The photograph explained a lot. Explaining the whistle to Horace was a greater problem, but Mike managed it. He also persuaded Horace to lock the photograph into the scanner until Ann came on duty again. She would never know that anyone else had seen it.

Having handed over to Jonathan, Mike went to look for a stiff drink. He was in a quandary. The slightest whisper of what he had found would make Ann look foolish, and it would be impossible to say anything to her directly. All the same, something had to be done. The girl would come to serious psychological harm if this went too far.

Two drinks later, he thought he saw a solution. All he had to do was pay enough attention to Ann to divert her attention from Horace. It would be a pleasant enough task from his point of view, and he had a feeling that Ann might not be too averse to the idea. After one for the road and two more to celebrate, he went home, feeling pleased with himself.

Next morning, he was less pleased. He had a hangover, and everything went wrong. The last straw came when he tried to yank the mains plug of his shaver out of its socket and pulled the lead out of the plug instead. He went to work in a bad temper, and treated Horace quite rudely during the first shift. Fortunately, Horace understood about hangovers and made allowances for them.

When Jonathan relieved him, Mike went down to the workshops to get his shaver lead mended. While that was being done, he noticed a group of engineers gathered round something in the test area, and went down to see what was going on.

The men were grouped round a complex machine which had appendages resembling a pair of arms and hands. As Mike approached, one of the hands reached out and took a pen from the pocket of an engineer who wasn't paying attention. There was a general round of applause, but the man who had lost the pen never noticed the reason. Answering Mike's question, one of the others explained;

— "That's TAHAS — it stands for Tactile Handling Simulator. Sensory pads on the fingers give it a sense of touch. Look — put out your hand."

Mike did so, and the TAHAS machine put out its own hand and grasped his with an astonishingly lifelike touch. It was only at the last moment that he realised that the left hand was exploring his jacket pocket. He joined in the general laughter, and said a mechanical pickpocket was quite unnecessary with so many thieving engineers about. This remark surprised the other man.

— "Oh, this is for your lot. The order came from your department, anyway. Horace is to have TAHAS on his spare I/O trunk, to take him a step nearer to being human."

Mike wandered back to collect his shaver lead, wondering who had thought up this particular bright idea, and had gone at least a dozen paces before he came to an abrupt halt, utter consternation spreading over his face...

DOWN A WIRE

Henry Budgett

Have you ever stopped to think what a restricted, nay, insular existence your micro may be suffering? Perched upon a desk somewhere, out of touch with the outside world? Why not treat it (him/her) to a modem, and reap some of the very attractive benefits for yourself?

There has been an awful lot of fuss in the computer press recently about the burgeoning growth of information systems that are open to the microcomputer-owning public. Databases like Prestel and its micro-specific Micronet section together with an ever increasing number of privately run systems known as bulletin boards have, apparently, sprung up overnight. In reality Prestel has been around since the late '70s — see some of the early issues of CT if you don't believe me — and bulletin boards or BBSs as they are often known have been in use even longer although mainly in America.

The existence of systems of this kind, however, doesn't always mean that the 'right' people can use them. Prestel was designed as a public information data base but until the price of the special adaptors or converted television sets came down considerably about the last group of people who could afford to own an adaptor and use the system were the general public. In many cases the high costs were due to the exorbitant costs of manufacturing the modem which the system needed to connect your television through the telephone network to the data base. British Telecom had, until very recently, a total stranglehold on the approvals procedure and until a modem was approved there was no way in which it could be connected to the telephone service!

Fortunately the re-organisation of BT to remove some of its total monopoly of the telecommunications market took the approvals procedure at least partially away from BT and created a new organisation BABT which seems to move somewhat faster. Coupled with this re-structuring has been a revolution in the chip market and we now have a complete modem available on a single chip. These two factors mean that we can now benefit faster from the new technology, probably as close to an ideal solution as one can ever expect . . .

THE SERIAL CONNECTION

Connecting your home computer to a printer, disc drives or some other peripheral is relatively easy as they will usually be in the same room, if not on the same table. In this kind of situation we tend to wire things up in parallel, each byte of information is sent in one go down eight wires (one for each bit) and control is very easy to maintain — just add a couple more wires to act as 'handshake' lines. Getting your computer to talk to its bigger brothers located in another part of your office or half way round the world is a slightly different proposition. Fortunately we already have a means of communication which circles the globe, the telephone system. All that is needed is a means of connecting our computers to it.

Because the telephone system is so widely used by large computer systems the technology is already well established, all the home computer user needs is a simpler and cheaper version of it. The conventional method of connecting computers to the telephone system is the device we've already mentioned called a MODEM. This odd name is simply an acronym formed from the words Modulator and Demodulator. As the technicalities of serial communication, that's sending one bit at a time down a single wire rather than a complete byte through eight wires, get more than a little complicated it's probably easier to look at something we understand quite well, the cassette interface, and show how that uses the serial system. This is, of course, a rather special case but does serve to illustrate the basic principles.

A typical setup: Selecta Keypad, Modem/acoustic coupler and...a telephone.

An audio cassette recorder of the type used by most small computers for program and data storage is obviously best suited to storing sounds, yet the information is held inside the computer in the form of binary numbers. These must be turned into sounds in a way that will allow the computer to recognise the difference between a bit that is set 'on' and a bit that is set 'off', the 0s and 1s of binary. The simplest method of doing this is to create one sound that represents a 1 and another that represents a 0.

Each character consists of one byte, a total of eight bits, so each character needs a sequence of eight tones to represent it. However, in order to indicate the beginning and end of each byte the computer usually adds an extra tone to each end making a total of 10. These extra tones are called the start and stop bits and their value is always the same, either 1 or 0 depending on the particular computer, and they provide a synchronising signal to the computer allowing it to tell where each character begins and ends.

The information itself is stored in a continuous block but to make error checking simpler it is often broken down into segments. Typically these are 256 bytes long and they will generally include extra information which enables the computer to check for itself that the information it is reloading is the same as that which was originally stored or transmitted. The first method of checking is to use the eighth bit of each byte as a parity bit. This is set so that it always makes the number of bits set to '1' an even or an odd number — the choice tends to be arbitrary depending on the computer. While this allows a single byte to be checked for errors we need a second system to detect errors in a large block of information.

The method used here is quite simple and is called checksumming. The first byte in the block of information contains the number of bytes that are held in the segment while the last byte contains a specially calculated number representing the total of all the bytes added together. When the computer has received the whole block it checks the figures it has been given with the ones it has calculated for itself and, if they don't match, requests that the block be sent again.

The speed at which the tones are produced by a serial interface is usually (and incorrectly) referred to as the Baud rate. The name originates from the Baudot code used in the earliest forms of the electric telegraph and actually relates to the number of times the signal changes per second. A more accurate measure would be the number of bits that are transmitted per second. The faster the quoted speed, they typically range from 110 to 9,600 bits per second, the quicker the information will be passed between the computer and peripheral. Unfortunately the reliability of the system suffers the faster the tones are produced, a speed of 1,200 bits per second is both reliable and sufficiently fast to prevent frustration. On a true serial interface where the '0s' and '1s' are represented by voltage levels rather than tones speeds in excess of 9,600 bps are common but where the bit values are expressed as tones the maximum speed is about 2,400 bps. Some cassette interfaces offer two speeds, usually an ultra-reliable slow speed of 300 bits per second and a fast speed of either 1200 or 2400 bits per second. This allows copies of valuable programs to be held in both forms in case of accident.

The modem works in very much the same way as the cassette interface on your micro; the patterns of binary ons and offs are converted into different frequencies and then sent down the telephone lines. The system is slightly more complicated by the fact that the computer at the other end needs to be able to send information back and so two pairs of frequencies are allocated. One system, the CCITT standard throughout Europe, allocates 1,070 Hz and 1,270 Hz to '0' and '1' at the end that sends or 'originates' the call and 2,025 Hz and 2,225 Hz for '0' and '1' at the receiving or 'answer' end. One further difference between a cassette interface system and a modem is that the modem always transmits a constant tone as long as it is connected, this allows the computer to recognise that the incoming call is still there. This tone is modulated by the other frequencies, hence the reason for the strange name! In practical terms a modem is linked to the computer by a true serial interface which uses changing voltage levels to represent the binary '0s' and '1s' while the modem itself generates the tones.

Modems come in all shapes and sizes but there are two main varieties; direct connection types and acoustic couplers. While the former are plugged into the telephone system all the time the latter can be carried around easily and fitted to any convenient telephone. One problem here is that the new generation of telephones currently being introduced don't seem to fit the cups on many acoustic couplers! Because the system works by transmitting sound it is important to avoid the possibility of external noise getting into the telephone handset. If the telephone doesn't fit properly or there is a great deal of external noise the data may very well be corrupted. Acoustic couplers also tend to operate at slower speeds than direct connect modems in order for there to be the maximum amount of signal available. If you consider a frequency of, say, 2,400 Hz it only takes the minimum of maths to appreciate that there isn't going to be much of it representing one bit at a transmission speed of 1,200 bps!

Direct connect modems tend to be more sophisticated units than their acoustic cousins. Common features offered are those that allow the calls to be intercepted automatically — auto answer — where they recognise the carrier tone and so determine that the incoming call is from a computer rather than a human. They also tend to offer a wide range of working speeds. The transmission characteristics of a normal telephone cable prevent any signal faster than 1,200 cps being transmitted with reasonable reliability. This limits a two-way link to 600 cps with an ultra-reliable 300 cps also available. One oddity in the realms of speed is the split baud rate system used by Prestel and other Viewdata systems. Here, it was reasoned, the user would want to get information as fast as possible so they transmit at 1,200 cps yet the user only has a simple keypad and can only type fairly slowly so he transmits back at 75 cps. This represents about the fastest possible communication system that can be achieved over the standard telephone network, leased lines are, of course, something else again but they do tend to be a little on the expensive side . . .

TALKING TOGETHER

The term duplex is commonly heard in the context of modems and is usually expressed as either full duplex or half duplex. Put as simply as possible a pair of modems operating in full duplex mode can talk to one another simultaneously whereas in half duplex mode one must finish its 'conversation' before the other can start.

There is, however, one major and virtually insurmountable communication problem in the world of modems. This is the fact that the American computing industry uses a system called Bell 103 and Europe uses a system called CCITT V21. Needless to say, they are incompatible and so your British modem won't be able to speak to an American one. The extent to which this will affect the normal user is so minimal as to be not worth worrying about in real terms. It does affect the modem makers, however, in that devices, usually based on the World Chip modem, which provide the American frequencies as well as the British ones stand little if any chance of being approved for use on the UK telephone network. This isn't just beaurocratic interference but based on technical difficulties as well. Certain tones produced by the Bell system can cause strange things to happen to our telephone exchanges; among the silliest is that they can be confused into thinking they have been told to disconnect the line and so cut your conversation off in mid-transmission!

USES FOR MODEMS

Devices like the Sendata and its near relatives in the acoustic coupler market allow portable computers like Tandy's Model 100 and Epson's PX-8 to be used as remote terminals by anyone from representatives to journalists. Orders, articles and. correspondence can be entered into the portable's memory and then dumped down the telephone lines to head office. The flow of information can be in the other direction too with your head office leaving vital messages on the system for you, much more reliable than telephone messages which never get passed on. Although this form of electronic mail is pretty simple possession of a modem and a suitable terminal or home computer is a first step to getting into real electronic mail, using services like Telecom Gold. These offer the sort of instant communications facility that until recently only the largest companies could have had access to yet at price which most can afford.

In business a computer terminal and modem allow instant access to a wide range of information services and computer bureau. For example, chemists shops who get their supplies from Vestric are now almost completely computerised with the staff entering stock items and required quantities and then transmitting the information to the main warehouse computer system. Here all the billing, stock control and delivery instructions are handled almost instantly allowing the High Street outlet to offer a better and faster service yet keeping the real costs right down.

For the home computer user the modem can provide, and is providing a much more reliable way of sending programs to friends than entrusting a cassette to the post. It's often quicker and more reliable as well if you use one of the sophisticated error checking methods such as Christensen protocol. This effectively adds extra codes to your documents or program before it transmits it. The other end checks to see if these are still attached and if any are missing it will ask for that portion to be re-transmitted.

The real growth, however, has been in the provision of information and general interest systems like Micronet 800 which is actually part of Prestel and the various privately run bulletin board systems. These, often known as BBSs for convenience, offer a central system where news and views can be exchanged. Quite often too they represent a source of cheap second-hand equipment and software although one or two have gained a reputation for handling somewhat dodgy goods. The onus tends to rest on the SYSOP or system operator to keep the quality high and, so far at least, none of the British systems have suffered the fate of certain American BBSs which have been closed down. The subject of BBSs is one which needs rather more than a paragraph or two to cover but suffice to say that their growth is linked directly to the greater availability of cheap modems.

IBM AT HALF PRICE

Ivan Berenyi

When an Asian company claim to have designed an IBM PC compatible, portable micro which will not infringe copyright, it starts you wondering. When they reveal their intentions to sell the thing for less than 40% of the IBM's retail price, you sit up and take note! We have the details:

A portable IBM clone called the Phoenix — designed by an Asian company's American whizz-kids and set to cost less than the IBM PC by 40% — is claimed to be able to do all the IBM PC can without any copyright infringements, and is set to make its appearance in Britain in April/May of this year.

A 16-bit microcomputer weighing under 10 lbs and featuring a 16-line LCD screen, the machine was developed by Singapore-based **BML Computer Design.**

Superficially, the design of the Phoenix is moderately innovative inasmuch as it has a detachable keyboard. For portability, this keyboard clips on to the front of the system, and the adjustable LCD screen swings down to rest on top of the keyboard.

The Phoenix system measures 16.5 × 12.5 inches when the keyboard is attached, and is 2.4 inches deep when the screen is down.

An integral battery pack provides for a minimum of eight hours continuous usage, with a maximum of 12 hours. The built-in power supply fully recharges the batteries, while running the system simultaneously, and to get the batteries back to full charge takes half a day.

OPTIONAL CO-PROCESSOR

The system is based on an Intel CMOS 8086 processor chip running at five MegaHertz, with a socket provided for an optional Intel 8087 maths co-processor. A basic 128 Kilobytes of RAM storage is an integral part of the basic configuration, expandable to 512 Kilobytes.

BML claims to have designed its own BIOS operating system, the design, coding and verification of which apparently took three months. The Singapore firm is adamant about the Phoenix's IBM compatibility but they do stress that it infringes no IBM copyrights whatsoever.

A dedicated input/output processor emulates the IBM peripherals, with standard Centronics and RS-232C ports provided and IBM-type connectors used.

The LCD screen is BML Computer Design's ***piece d'resistance.*** This 16-line LCD emulates the IBM PC's monochrome monitor and acts as a window into the 25-line IBM display. Window scrolling is transparent to applications software, and is fully controllable by the user through the keyboard.

An optional 12-line mode is provided for greater legibility, giving true descenders on lower case characters. The display is smoothly adjustable, rather than through fixed steps, but no details are yet available regarding adjustment angles. A keyboard contrast control is, predictably, a standard provision.

PHOENIX KEYBOARD

The 72-key Phoenix keyboard givres full emulation of the 83-key IBM PC keyboard, the Case of missing keys solved by double-ups elsewhere. All 10 function keys are provided — albeit along the top of the adjustable keyboard, rather than down the left hand side — but cursor controls, numeric keys and such vital functions as Home and End require use of a Control-type key.

Data transfer speed comparisons with the IBM PC are favourable. A proprietary interface bus transfers data at 125 Kilobits per second and can multidrop to eight mass storage or communications devices. These connect to the bus by coiled cable using a modular plug interconnect system.

In terms of the disk solution offered by the Phoenix, there is similarity with the Hewlett-Packard portable; there are rumours that at least one of the BML Computer Design team hails from the Hewlett-Packard stable.

Like the HP portable, which was developed within approximately the same time frame, the Phoenix has a single 3.5 inch built-in disk unit, offering either 360 Kilobytes of single-sided, or 720 Kilobytes of double-sided disk storage capacity.

Quite clearly for the sake of compatibility with the IBM PC, the single-sided disk unit is formatted as an IBM double-sided drive, while the double-sided unit emulates two IBM double-sided drives, which are named A and B.

MYSTERY

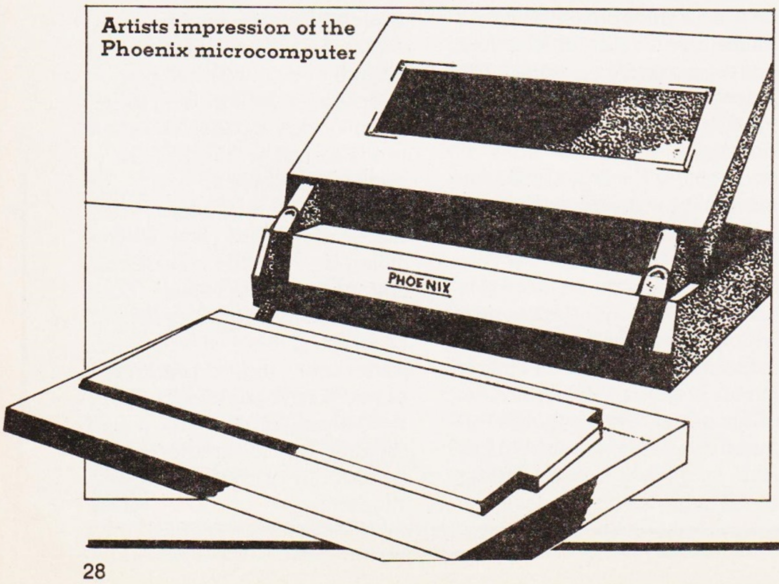
A certain air of mystery surrounds the Phoenix microcomputer, which BML Computer Design developed for an unidentified partner. When contacted, all that **Brad Eltman** — an American who is one of the directors of BML Computer Design — would say about this partner was that it was a large, Singapore-based manufacturing company.

It is apparently this manufacturing partner, which has been negotiating with putative OEM distributors in the US since November and in Britain since December/January, though BML itself is mooted to retain certain marketing rights to the Phoenix system, albeit it has not been revealed whether these will be limited in any way, geographically or otherwise.

"In the United States" says Eltman, "the distributor has already been selected, but the announcement is yet to be made, pending some finishing touches to the agreement."

"In Britain, which is seen as a very major market for the Phoenix, the selection process is only scheduled to be finalised in mid- or late March."

Strong rumours that a brochure of the Phoenix, and a folder on the activities of BML Computer Design would have been sighted within the London dealer community already in January, were, dismissed by Brad Eltman as "unlikely".



Artists impression of the Phoenix microcomputer

EXPANDING MEMORY

Bill Horne

Most programmers feel the need at some time to have more memory available to them, and we look at some ways of achieving this.

As programmers grow more ambitious and their programs grow larger, a point is inevitably reached where their requirements exceed the available storage capacity of the machine they are using. Since it is very difficult to predict the amount of store required for a program before the program is written, the stated requirement tends to exceed the real need, but that's life . . .

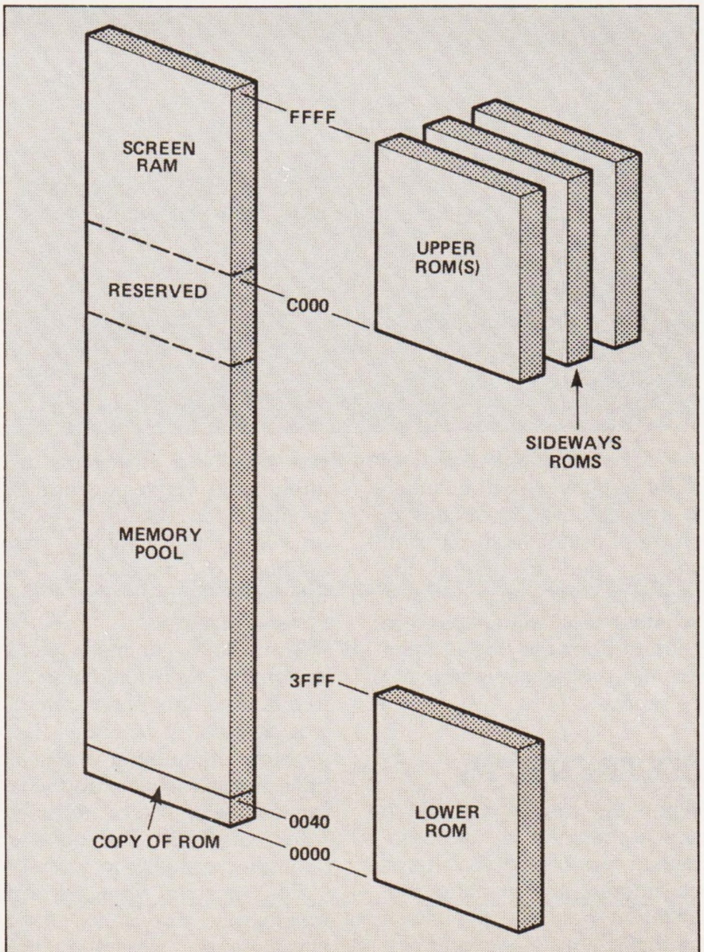
Unless there are very good reasons for enforcing a limit on store size, the designer may have to go away and think of a method of providing extra store. There are a number of options open to him.

BANK SWITCHING

One of the oldest methods of enlarging store is bank switching. The limitation on store size usually sprang from the number of address bits which could be defined in an instruction. In some machines, the limit was no more than 6 bits, capable of defining no more than 64 addresses. Each block of 64 addresses was called a 'page', and while instructions held within a given page were being executed their data had to lie within the same page. This was possible, because core stores were used, and a core store location can be used either as ROM or RAM, a valuable feature that has been lost with more recent forms of memory. On the other hand, being limited to a single page was inconvenient, and the scheme was developed so that 'page 0' was always accessible, a concept perpetuated in the 6502 CPU.

This scheme was viable for up to 64 pages, giving 4K of memory, but would go no further. When a requirement for an increase to 8K was raised, it seemed that the only solutions as to provide two identical banks of store that could be selected as necessary.

It became clear that this posed a further problem. To transfer action from location A in bank 1 to location B in bank 2, it was essential to perform a jump and a bank switch simultaneously. The only obvious way to avoid this was to jump to location C, at which there was a bank-switch instruction, and pass to location C in bank 2, where there was a jump to location B.



Amstrad memory map

The solution adopted was inspired by memories of the Wilson pre-selector gearbox popular in the nineteen-thirties. With that, a gear was selected by a quadrant lever, but it was not brought into use until the clutch pedal was depressed and then released. In the same way, a bank switch was 'primed', but did not take effect until a jump instruction was executed.

That was all very well with a processor designed 'in-house' from small scale integration logic, but would be more difficult to apply to a modern microprocessor. With a Z80, the M1 line goes low when an op-code is being read, which would allow the code to be passed to a ROM or logic array which would recognise jumps, calls and returns, by giving a single-bit output. That bit would copy the pre-selected bank number into the bank select register, changing bank at the time of a jump, call or return. A complex system, perhaps, especially with the need to disable the ROM after a CB, DD, ED or FD op-code prefix, but fast and effective.

Getting into a bank, however, is only part of the problem. Once there, the system will expect to find code, but that either means that the bank must contain ROM holding pre-set program, or there must be a way of loading program into RAM before the bank is entered. One system did this by reading from one bank while the other bank was enabled for writing, but that raises ticklish problems of timing. It also implies a special working mode that has to be enabled and disabled. The same system read instructions from one bank and passed data to and from another. That is only feasible with single-byte op-codes, since qualifying bytes which follow an op-code are indistinguishable from data.

The AMSTRAD CPC464 solves the problem by calling routines which run independently of bank selection, and perform the actions needed to produce effectively simultaneous jump and bank-switch.

After some twenty years of development, bank switching is still causing problems, but it is an essential element in the growth pattern of the smaller systems.

VIRTUAL MEMORY

In a virtual memory system, the basic address word defining all the available locations is extended to define the block of code or data which is expected to occupy a given area of memory. A register is used to define the block last read into that area of memory. If the correct block is not present, it is fetched from disc before execution continues.

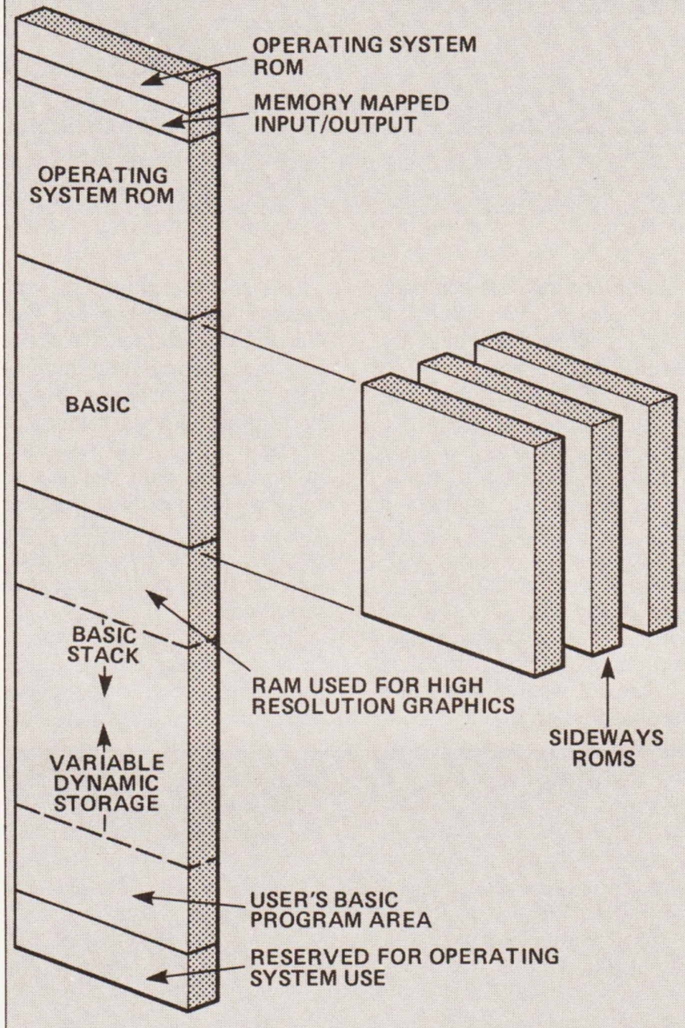
That is a simplified description of the concept. The practice is usually rather more complex, but the idea is straightforward. Suppose you have a database which is four times the size of the area which it should occupy. It is held on four disc files. When you want to access a given part of the database, you define an address which identifies that part. If it is already in store, you go ahead. Otherwise, the required part is read into store so that it can be used.

A similar idea is built into CPM, under the name of 'cache'. If you set up your CPM system suitably, it will read a number of sectors into RAM, and will let you use those sectors freely until you ask for communication with a sector which is not in store. The sector copy which has been present in RAM longer than the rest is then transferred back to disc, with any changes you have made, and the new sector you want is read into RAM. The process is invisible to the user, and one CP/M addict was quite surprised to find that he had been using virtual memory techniques for a long time . . .

SIDEWAYS ROMs

At first sight, sideways ROMs may seem to be almost exactly the same as switched banks, but there are subtle differences.

To begin with, a sideways ROM is usually entered at a given point and thereafter controls system execution until another sideways ROM is selected. In the simpler form of a sideways ROM system, it is not too easy to jump in and out of a particular ROM without losing the current execution address. In more highly developed systems, such access is made possible by an intermediate visit to the main central system, which can access any sideways ROM.



BBC memory map

Without a doubt, sideways ROM systems have done much to expand the capablities of systems using eight-bit processors, and to extend the useful life of such devices. Whether they will succeed in holding back the tide of more modern processors remains to be seen.

SIXTEEN BITS

When the development of a new generation of microprocessors was considered, it was considered, it was natural that attention should be given to an expansion of memory addressing capability. The objective was achieved in several different ways. Extra pins had to be used to implement extra address bits, whatever the method of generating the address. Beyond that, the systems diverged.

The 8086 defined store 'segments', each containing up to 64K. The start of a segment was defined by a number in a register. The number was multiplied by sixteen and added to the displacement or 'address within segment'. This gave a nominal capability to address 1024K of store, well beyond imaginable needs for most applications. However, the system led to longer instructions, of up to seven bytes — eight with prefix bytes — and complicated the programmer's task considerably.

It might seem that this was a software variant of the hardware switched bank system, and it could certainly be used in a similar way, but it does avoid some of the problems that have been described in relation to bank switching.

The other 'sixteen-bit' processors rely more on assembly of extended addresses in registers, and approach more closely to the ideal of a large homogenous store, with direct and unrestricted access to all parts of the address range. However, the provision of memory management facilities hints that this can raise problems when different store areas overlap.

For large systems, the more recent processors may be justified, but they do impose penalties which the older processors escape. They allow store to be extended — at a price. They should, perhaps, be used only where the price is economically viable.

HOW BIG?

The available methods of extending store size have been reviewed, but one question remains open: is it really necessary to have very large store space?

A few years ago, a young man developed a 'toolkit' program for use with an 8K BASIC interpreter. Later, as an experiment, he re-wrote the BASIC in more efficient form, and found that he then had room for the toolkit routines within the 8K! Other cases might be quoted where a re-write has saved a surprising amount of space.

Programmers may shrug their shoulders and say that economic use of hardware is not part of their task, a view which is sometimes made only too clear by the routines which they write. It is not a view that pleases the hardware man, who is continually being asked to provide more capable hardware, the capability of which is then wasted.

Remembering that software cost is — or should be — a one-time charge on the overall cost of the system, whereas hardware is a recurring charge, it would be at least gracious of the programmer to try to make the best possible use of what he is given to execute his routines.

AVERAGES AND TRENDS

M J Bedford

Additional Programming by V J Chittenden

Andrew Lang's description of one individuals misuse of 'collected, classified, and interpreted quantitive data' is amusing, but just how imaginative should a statisticians interpretation be?

> **"He uses statistics as a drunken man uses a lamp-post — for support rather than illumination." — Andrew Lang**

In the increasingly complex society of today the ability to spot trends or determine which way public taste is swinging can be of great importance to industry. Similarly, it is of prime interest to the Government how the population is distributed geographically, the numbers of people within each age group, etc. The required data are easy to obtain; the Government takes a census or uses figures already to hand in their various departments. Industry carries out market surveys. The collected data is of little use in its raw form and further manipulation, using statistical techniques, is necessary before meaningful comparisons or conclusions can be made. Statistics therefore play an important part in our lives, since it is through their use that government and commerce decide what we, the consumers, want (or ***need*** as they prefer to put it).

There is a widespread mistrust of statistics and statisticians. It is often said there are "lies, damned lies, and statistics". This lack of trust is understandable. All too often a set of data seems to yield contradictory results. A typical example is the continuing argument between health agencies and tobacco companies as to the efficacy of smoking in causing lung cancer. Similar contradictions arise during wage negotiations when it would appear that at the end of the day both sides have each won a resounding victory. Cynics may feel that one side (or more probably both) is cheating. However, this is not necessarily the case and both sides may have been quite honest in their interpretation of the results. The key word is "interpretation" and it is this area where the unwary may be easily confused.

The most often used statistic, and I sometimes think the least understood, is the Average.

ARITHMETIC AVERAGE

The popular belief that the average is a fixed single quantity allows plenty of scope for deliberately misleading figures to be quoted in order to bolster an argument. It is in the smale class as the "nine out of ten" survey often used in advertisements. Sample a large enough group in small batches and it is likely one batch of ten will have nine who will say what the ad-man requires, always assuming the claims are not too outlandish or outrageous. Hey-presto! Nine out of ten people prefer Brand X. Unfortunately, what you are ***not*** told is that the majority surveyed offered no preference, or heaven forbid, preferred Brand Y. Another popular method of fudging occurs in pictorial representation of numerical data. A cartoon representation, e.g. cars for car sales, of the figures involved is shown in conjunction with a simkilar cartoon for the figures it is being compared with. The linear dimensions of the cartoon representations are proportional to the numerical values. A little thought will show that for example, doubling the linear dimensions quadruples the area. Visually impressive, but quite inaccurate. The area only should be doubled, this requires an increase of 1.41 × the linear dimensions.

The popular conception of the average coincides with the statistical definition, i.e. the average is the sum of the data divided by the number of data. The arithmetic average is defined by the formula

$$\textbf{Arithmetic average} = \frac{\Sigma x}{N}$$

where $x = x_1 + x_2 + x_3 .....x_n$

x = datum
and
N = number of data

Unfortunately there exists other averages and measures of the central tendency of groups of data. The two other averages are the Geometric and the Harmonic, each of which has a specific application.

TABLE 1 **Price of a computer kit up to date**

Cost per kit	**£250**	**£200**	**£150**	**£100**
percentage of previous price	**—**	**80%**	**75%**	**66.7%**

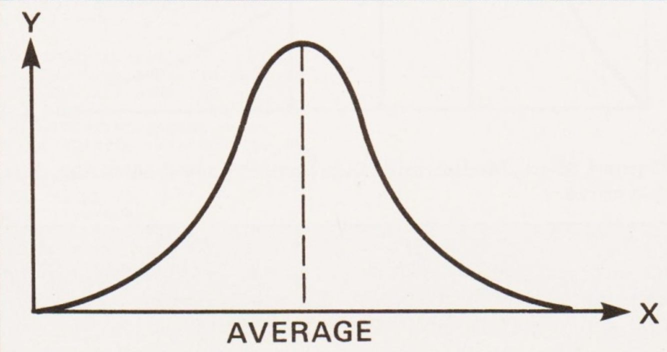


Figure 1. Normal distribution curve

GEOMETRIC MEAN

The Geometric mean is used in averaging ratios, percentages and rates of change. It is defined as

$$\mathbf{A = (x_1 * x_2 * x_3 * .... x_N)\ 1/N}$$

The use of the geometric mean is illustrated in the following example. The approximate price of a computer kit followed the trends shown in table 1.

The arithmetic average for the percentages of previous price is 73.9%, which would yield a final price of £101, therefore:

The Geometric mean,

$$\mathbf{G = (80 * 75 * 66.7)\ 1/3}$$
$$\mathbf{= 73.7\%}$$

The difference between the two averages may seem miniscule but the geometric average yields the most accurate final price. If the differences between the decreases had been greater then obviously the differences between the arithmetic and the geometric means would have been greater. It may be noted that the geometric mean is always less than the arithmetic mean.

HARMONIC MEAN

The third mean is the harmon mean (H). This is defined by

1/H = 1/N 1/x

It is used when dealing with ratio data having physical dimensions such as miles per hour or cost per pound. For example the price of static ram has been dropping quite rapidly. If a total of 16K of 2114 were purchased in, as shown below, the cost per K of memory must be calculated using the geometric mean viz

4K @ £9.00 per K
4K @ £6.00 per K
8K @ £3.00 per K

total cost = £84

The arithmetic average, £6 per K is obviously wrong, this would give a total outlay of £96. The true price per K is given by the Harmonic mean

$$1/H = \frac{\text{total memory}}{\text{total sum paid}}$$

4 + 4 + 8
= 26 + 24 + 24

H = £5.25 per K

Thus total outlay:

outlay = £5.25 × 16
= £84

Other measures of central tendency may sometimes be quoted and these are often confused with the average. Samples from large groups (populations) often exhibit what is called normal distribution. To plot a distribution curve data are grouped and a plot of group value versus number of data in the group is drawn. Normal distributions follow the bell-shaped curve shown in Fig. 1. Results are distributed symmetrically about the average. Further definition is given by determining the standard deviation, a measure of variation from the average. 68% of data will lie within the standard deviation from the average. However, not all data may show this symmetrical distribution. Such data is said to be skewed (Figs. 2,3).

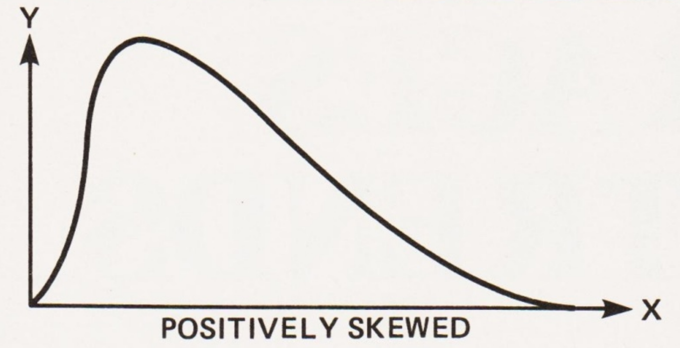


Figure 2. Positively skewed distribution curve

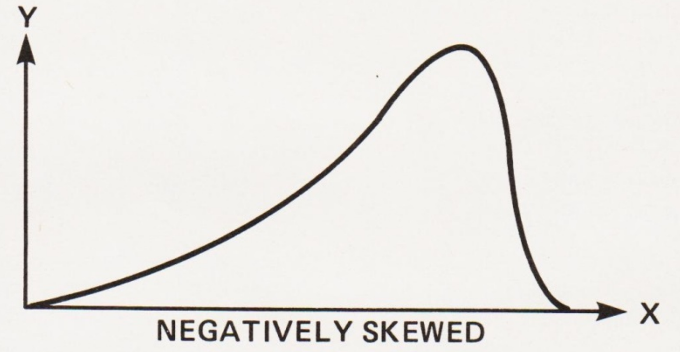


Figure 3. Negatively skewed distribution curve

Distributions of this type arise where there is no lower limit or upper limit. Wage scales within companies typically show positive skewness since there is a limit to minimum earnings, (nobody works for nothing) but, theoretically, no upper limit on earnings of top management.

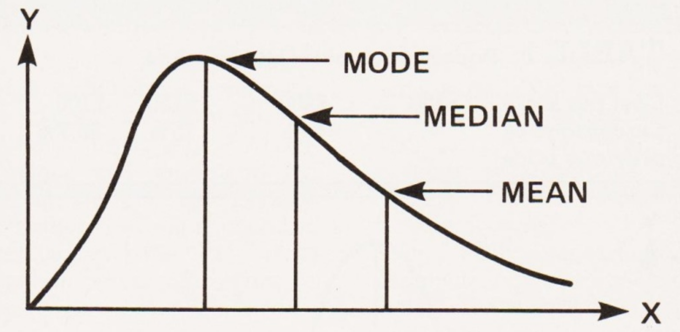


Figure 4. Mean, Median and Mode for positively skewed distribution curve.

MEAN, MEDIAN, MODE

For skewed distributions the median and the mode are often used to describe the data distribution. The median is simply the middle value, effectively mode is the value that occurs most often.

For positively skewed distributions mode < median < mean (Fig. 4). It can be seen that for a normal distribution mode = median = mean. The mode and median are often the mid-point between the highest and lowest values. The mode and median are often used by unions to justify wage demands, typical claims being "30% of members only receive..." or "50% of workers receive less than...". Management, of course, counters these claims by quoting the average wage of the company. Hence the ambiguous figures in wage bargaining and the ability of both sides to defend their case using the same set of data.

When assessing claims concerning the average it is thus necessary to know exactly which average is being discussed or whether the median or mode are being used. For large samples an indication of the spread of data is given by the standard deviation. As mentioned above sixty eight per cent of results will lies within + or − one standard deviation from the mean, ninety five per cent will lie within + or − three standard deviations.

THE PROGRAM

The following program will allow you to determine the mean and standard deviation for sets of data. It gives a visual indication of data distribution, i.e. a histogram is plotted. Median and mode are not computed, both can be easily determined, pencil and paper (useful tools that seem to be falling into disuse) for the former and visual inspection of the histogram for the latter. The program can deal with data entered as individual datum or groups. When dealing with large amounts of data it can simplify matters to divide them into groups of equal range and then use the mid-point of each range along with the number of data within the group. If the data distribution does not seem symmetrical, i.e. normally distributed, then a check can be made by calculating skewness and ***kurtosis***. Kurtosis is an indication of how sharply the data is distributed, or spread, over the range of interest. For normal distribution the kurtosis value is around three; flat curves have values less than three and peaked curves have values greater than three. The distribution can be considered symmetric if −0.5 < skewness < 0.5. Distribution is highly skewed if skewness exceeds + or − one. The program was written for the Amstrad CPC464 but should be quite easy to convert for use on machines with similar features.

Program listing for the Amstrad CPC464

```
10 MODE 2
20 PRINT"          STATISTICS PROGRAM"
30 PRINT:PRINT "This program can be used to calculate"
40 PRINT:PRINT "means, moments, etc. for ungrouped and"
50 PRINT:PRINT "grouped data.":PRINT
60 PRINT:PRINT "ENTER 9999 when all data has been entered."
70 DG=1
80 PRINT:PRINT
90 INPUT"HIGHEST DATA";HD:PRINT
100 INPUT"LOWEST  DATA";LD
110 ND=HD-LD+1
120 ND2=ND:DIV=1
130 WHILE ND>600
140 ND=ND2/DIV
150 DIV=DIV+1
160 WEND
170 IF DIV=1 THEN WIDE=600\ND2 ELSE WIDE=1
180 XLEN=(ND2*WIDE)/DIV
190 NUM=ND2/DIV+1
200 IF NUM<9 THEN NUM=NUM-1:DIV=NUM/10:WIDE=WIDE*NUM/10:NUM=10
210 DIM C(NUM)
220 CLS
230 PRINT:PRINT"Is data (1)ungrouped or (2)grouped?"
240 INPUT GR
250 IF GR=1 THEN 290
260 IF GR<>2 THEN 240
270 PRINT:INPUT"Group value";D:INPUT"Number in group";DG:GOTO 300
280 PRINT:INPUT"Group value";D:INPUT"Number in group";DG:GOTO 320
290 INPUT"DATUM";D
300 AT=D:GOTO 320
310 INPUT"DATUM";D
320 IF D=9999 THEN 420
330 IF D<LD OR D>HD THEN PRINT"OUT OF RANGE!":IF GR=1 THEN 310 ELSE 280
340 GL=LOG(D)
350 N=N+DG:LG=LG+(GL*DG):H=H+(DG/D)
360 TA=DG*(D-AT)+TA:TD=(((D-AT)^2)*DG)+TD
370 V=INT((D-LD)/DIV)+1
380 C(V)=C(V)+DG
390 IF C(V)>MOST THEN MOST=C(V)
400 IF GR=1 THEN 310 ELSE 280
410 REM ANALYSIS OF DATA
420 AM=AT+(TA/N):GM=EXP(LG/N):HM=N/H
430 CLS
440 PRINT:PRINT"ARITHMETIC MEAN =";AM
450 PRINT:PRINT"GEOMETRIC  MEAN =";GM
460 PRINT:PRINT"HARMONIC   MEAN =";HM
470 PRINT:PRINT"TOTAL DATA      =";N
480 PRINT:PRINT:PRINT"PRESS ANY KEY FOR REST OF ANALYSIS"
490 WHILE INKEY$="":WEND
500 M2=(TD-((TA^2)/N))/(N-1):SD=SQR(M2)
510 GOSUB 710
520 REM********* REST OF ANALYSIS *****************
530 CLS
540 LOCATE 27,2:PRINT"REST OF ANALYSIS"
550 PRINT:PRINT"MEAN     =";AM;"+ or -";SD
560 PRINT:PRINT"VARIANCE =";M2*(N-1)/N
570 LOCATE 1,12:PRINT"Kurtosis and Skewness (Y/N)"
580 A$=INKEY$
590 IF A$="N"OR A$="n"THEN END
600 IF A$<>"Y" AND A$<>"y"THEN 570
610 PRINT:PRINT"RE-ENTER DATA (9999 when done)"
620 IF GR=1 THEN INPUT DATUM;T ELSE INPUT"Group value";T:INPUT"Number of data";D
G
630 IF T=9999 THEN 660
640 DF=T-AM:M3=M3+DG*(DF^3):M4=M4+DG*(DF^4)
650 GOTO 620
660 SK=M3/(N*(M2^1.5))
670 KU=M4/(N*(M2^2))
680 PRINT:PRINT"SKEWNESS=";SK
690 PRINT:PRINT"KURTOSIS=";KU
700 END
710 CLS
720 SF=1
730 YLEN=(360\MOST)*MOST:IF YLEN=0 THEN MOST=MOST/10:SF=SF*10:GOTO 730
740 PLOT 40,32:DRAWR XLEN,0
750 PLOT 40,32:DRAWR 0,YLEN
760 LOCATE 1,8
770 FOR Z=1 TO 9
780 PRINT MID$("FREQUENCY",Z,1)
790 PRINT CHR$(13)+CHR$(8)
800 NEXT
810 MOVE 250,12:TAG:PRINT"DISTRIBUTION";
820 LOCATE 1,15
830 MOVE 28,38
840 PRINT"0";
850 MOVER -4,0
860 YDIV=YLEN/MOST
870 INC=INT(MOST/10.1)+1
880 FOR Z=1 TO 10
890 MOVER -(8+(LEN(STR$(Z*INC*SF))-2)*8),YDIV*INC
900 PRINT MID$(STR$(Z*INC*SF),2);
910 PLOTR 6,-6:MOVER -6,6
920 NEXT
930 XINC=INT(NUM/10.1)+1
940 CENT=-((INT(WIDE\2)-4)*(WIDE>29))
950 MOVE 32+CENT,26:PRINT LD;
960 FOR Z=1 TO 10
970 MOVE 40+WIDE*XINC*Z+CENT,26
980 PRINT MID$(STR$(LD+Z*XINC*DIV),2);
990 NEXT
1000 MOVE 41,32
1010 FOR P=1 TO NUM
1020 YLEN=C(P)*YDIV/SF
1030 FOR Q=1 TO WIDE+(WIDE>2)
1040 DRAWR 0,YLEN:MOVER 1,-YLEN
1050 NEXT Q
1060 MOVER -(WIDE>2),0
1070 NEXT P
1080 TAGOFF:LOCATE 26,1:PRINT"PRESS ANY KEY TO CONTINUE"
1090 WHILE INKEY$="":WEND
1100 RETURN
```

LADDER LOGIC

John Owen

Experiment with logic without touching a soldering-iron, using this ladder logic designer/simulator for the BBC Micro.

As we are frequently being reminded, the Microprocessor is getting in on the act everywhere we look. In schools and offices and even, of course, at home, the micro is almost mandatory equipment. But during these last few years, while the advertising men have been persuading us of the necessity of owning a home micro, a quieter 'revolution' has been taking place in industry and particularly in manufacturing industries. In these days of fierce competition, it is essential that production lines and other industrial processes can be modified and adapted to suit the changing market requirements, as quickly and as cheaply as possible. Only a few years ago such changes could have meant days, or even weeks, with production at a standstill while the wiring of the control system was changed. Thse days, however, much of the hard-wiring has been replaced with 'soft-wiring' with the use of programmable logic control systems. Now, the control scheme can be changed by making modifications to the controller's software. The modifications can be made during normal maintenance shutdown periods and, if they fail to work as expected, the original software can be loaded back with no damage done (except perhaps to the programmer's pride!).

CONTRIVED CROSS

One of the biggest problems with microprocessors is that they only understand machine-code. Industrial process control schemes are difficult enough to understand as it is without the additional complication of trying to grapple with machine-code. In the same way as the manufacturers of home micros have developed high level languages, such as BASIC, in order to assist us mere mortals, so the manufacturers of industrial Programmable Logic Controllers (PLC's) have developed 'high level' languages of their own to assist industrial maintenance personnel. These languages divide roughly into two types. The first is a sort of 'contrived cross' between BASIC statements and assembler language mnemonics; Texas Instruments, for instance, use program statements such as:

```
STR X1 AND X2: STR X3 OR STR OUT Y1
```

Brown Boveri Corporation would use:

```
IF X1 AND X2 THEN SF1: IF SF1 OR X3 THEN Y1
```

The approximate BASIC equivalent of these statements, by the way, would be:

```
Y1 = (X1 AND X2) OR X3
```

THE VIDEO PROGRAMMER

Although this type of mnemonic language is considerably better than attempting to program control schemes in machine-code, it has to be said that they are still not easy. Several PLC manufacturers, therefore, provide an alternative form of programming aid employing a visual display unit. Texas Instruments' Video Programming Unit and IPC's Loader-Monitor are two examples which allow actual circuit diagrams to be drawn directly on the VDU itself. The circuit diagrams are then automatically converted into executable program instructions — a procedure called 'compiling' — and these program instructions are then transferred from the programming unit into the PLC. itself. The maintenance personnel are thus freed of the task of wrestling with machine-code and can devote their attention to actually designing and writing the control system — in itself, a demanding enough job.

Like any high level computer language, there are certain rules and conventions which have be be obeyed when building up a circuit diagram on the VDU. One of the most immediately apparent is that the diagram's power rails run vertically up the sides of the screen instead of horizontally from left to right as with conventional circuit diagrams. Each circuit 'network' is built up horizontally across the power rails and, in its simplest form, consists of little more than a string of open or closed contacts resembling — and, in fact, mimicking — relay contacts, with an output 'coil' at the right hand side of the network. Although the contacts and the coils only exist in software, because they resemble 'real' relay contacts and coils, this form of circuit diagram is sometimes referred to as Relay Logic, but is more usually called Ladder Logic because of the networks' resemblence to the rungs on a ladder.

EXPENSIVE

Although these video programming units make it very much easier to design and develop ladder logic, they do suffer from one very serious drawback — they are ***very*** expensive. A typical programming unit alone will cost in the region of £5000 and, when this is added to the cost of the programmable logic controller itself, you will see why schools and colleges find it difficult to offer training facilities in this very important area of process control. Even large industrial concerns, which actually use this equipment, can ill-afford to have video programmers and PLC's lying around spare so the opportunities to gain vital experiences with this type of equipment is very limited.

VIDEO PROGRAMMER A LA BBC

It was for this reason that I developed the program which follows. It emulates, in a simple form, some of the basic features of a commercial video programming unit and a programmable logic controller. It allows you to design up to six lines of ladder logic directly on the screen and to open and close specific contacts within the logic, with the status of the outputs being displayed graphically on the screen. Before looking at how the program works, it may be worth looking at some examples of how to use it.

With the program loaded and running, you will be presented with a blank screen except for a small dotted 'box' at the top left hand corner and a line of symbols at the bottom. The dotted box is the ladder logic cursor and indicates where the next logic element will appear on the screen. The symbols at the bottom correspond with the BBC micro's red function keys and hitting a key will cause the corresponding symbol to be displayed on the screen. As a very simple example, enter the line of ladder logic shown in figure 1, as follows:

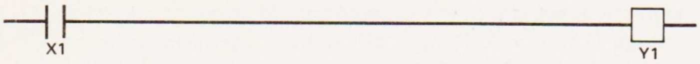


Figure 1.

Hit the function key f0 and the contact symbol will appear. Type X1 and hit return. The cursor will move one place to the right. Hold the function key f5 down and the cursor will step repeatedly to the right, drawing the horizontal wire as it goes. Hit key f2 and the output symbol will appear on the right hand side of the screen. Type Y1 and hit return. Having drawn the network on the screen, the logic representing it has to be compiled and entered into memory. This is accomplished by hitting the 'copy' key. The cursor will disappear and the message "COMPILED" will replace the function-key prompt at the bottom of the screen. Hit the number 1 key and the contact X1 will close and the output Y1 will be energized. Hit the number 1 key again to turn the output off. This example isn't exactly inspiring, so hit the 'copy' key again. The cursor will reappear. Change the diagram to that shown in figure 2, as follows:

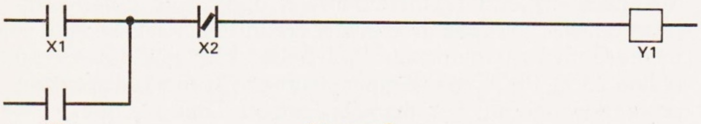


Figure 2.

Step the cursor one place to the right by using either the return key or the right-arrow key and hit f6 to draw the vertical wire. You will be greeted with a 'beep' indicating that you are trying to overwrite an existing element in the network. Hit the 'delete' key and the line inside the cursor will be cleared. Hit f6 again to draw the vertical wire. Enter the closed X2 contact in a similar way and insert the Y1 contact. Hit the 'copy' key again. This time you will be able to test the 'latching' action of the contact Y1. This concept of using outputs as input contacts is fundamental to PLC control schemes.

To edit the ladder logic, hit the copy key again. You can enter up to six lines of logic but note that this simple logic emulator has a couple of restrictions and limitations. (Well, what do you expect for £4500 less than the price of the 'real' thing?!)

- Each network may not be more than two lines deep.
- Input contacts X1 to X9 only, are allowed.
- Outputs and outputs used as contacts may only be between Y1 and Y6.
- Only one timer or one counter is allowed.
- Inputs to timers or counters may only be one line. Most 'real' programmers have this restriction too!

As a final example, enter the logic shown in figure 3, which is typical of that found in commercial ladder logic control schemes, to indicate a "motor failure". In this circuit, Y1 is the output to a motor, which is latched as before with the input contact Y1.

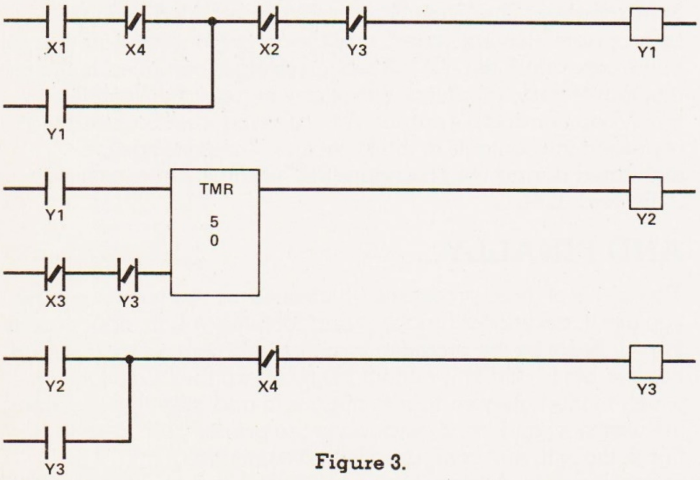


Figure 3.

When the motor starts, a contact, X3, should open. This is in the timer's reset line and prevents the timer from timing out. The contact Y1, in the timer network, is the timer's Start/Stop control. If the contact X3 fails to open within 5 seconds, the output Y2 is energized, which triggers the alarm output Y3. This takes the supply from output Y1 until the Alarm Accept contact, X4, is opened. As a safety measure, contact X4 is also in the circuit for Y1 ensuring that the alarm-accept circuit is 'healthy' before the motor can be started.

To program the timer, hit the f3 key, followed by 'T'. The word 'TMR' will appear. Enter the approximate time, in seconds, followed by RETURN. Note that, in this simple simulation, the timer is not intended to be accurate. A counter is programmed in the same way except you type 'C' after hitting the f3 key. Although we have not used it in the timer example shown, it is acceptable to have a network of contacts between the timer and it's output.

THE PROGRAM

After initializing the variables used in the program, the program enters an endless loop between lines 40 and 210. This loop waits for you to hit a key and then calls the appropriate Procedure. There are six Procedures associated with the initializing stage. PROCinitialize sets up a text window at the bottom two lines on the screen, turns the text cursor off and defines two User-Defined characters, 128 and 129. These characters are only used in the function-key prompt.

The program uses quite a few arrays and these are DIMensioned in this procedure. Array net$() is used to hold the final, compiled, version of each network, while the other string arrays are used in various stages of the compiling process. Because the BBC micro does not do any 'garbage collection' of old, unused strings, some string variables are first initialized to a long string of spaces before being cleared. This should eliminate any possibility of running out of memory as strings increase in length while they are being compiled.

PROCedures 'init1' and 'init2' are called during PROCinitialize but they are kept as separate Procedures because they are used individually elsewhere in the program. During PROCinit2, the status of each 'X contact' which is stored in the array X, is transferred into the variables X0 to X9. Similarly, the status of the outputs is obtained from the array out and transferred to the variables Y0 to Y7.

PROCfunction1 defines the user-defined function keys and prints the function prompt on the screen. PROCfunction2 prints an error message if the ladder logic you've drawn breaks any rules which are found when the logic is being compiled. PROCfunction3 prints the message indicating that the diagram has been converted into logic statements without error.

The screen-drawing routines between lines 1100 and 2460 are accessed from the main program loop (lines 40-210).

The easiest Procedures to understand are those related to cursor movement. Two variables, X and Y, represent the top left hand corner of the current cursor position so, for example, PROCcursorleft (at line 1310), first checks that the cursor is not already at the left hand side of the screen and, if not, the current cursor position is erased with PROCcursor(19,X,Y), X is decremented by 140 (the width of the cursor) and the cursor is replotted at the new position with PROCcursor(17,X,Y). For each step left, a variable — called 'column' — is also decremented by one. PROCcursorup and PROCcursordown operate in a similar way except variables Y and 'row' are incremented or decremented. PROCcursorright is only slightly more complicated in that, when the cursor reaches the right hand side of one row, it automatically steps to the left hand side of the next.

VISUAL REPRESENTATION

The screen is divided into a matrix of nine columns and six rows and a two-dimensional array, line$(), is used to represent it. Initially, every element in the array contains "0", representing the blank screen, and the variables 'row' and 'column' are used to point to a particular element corresponding to the cursor's current position on the screen. Although hitting the 'arrow keys' simply moves the cursor about on the screen, hitting one of the red function keys will actually draw something at the current cursor position. Key f0, for instance, calls the procedure PROCcontact which, not surprisingly, draws a contact on the screen (lines

1590 to 1660). Line 1670 calls PROCinput which obtains your string input, for example "X3", checks that this is valid and stores it in the appropriate element in the array line$() before stepping the cursor one place to the right. Hitting the f1 key calls the procedure PROCnotcontact which plots the 'not' part of the contact before calling PROCcontact to plot the rest of it. In this case, PROCinput changes your input string, for example "X5" to "NOT X5 and 1" before storing it in the array line$(). The 'NOT' is added because you have entered a 'not contact' and the 'AND 1' is added to simplify the compiling routine which comes later.

Function key f5 calls PROChwire which simply plots a piece of horizontal wire and puts the string "1" into the array. There are three types of vertical wire called respectively with f6, f7 and f8; One has a bend to the left, one has a bend to the right and one with a branch to the left and the right. In each case, the string "B" (for Branch) is entered into the array. Function key f3 plots the Counter or Timer but more about that later. So, as each line of logic is built up on the screen, the array line$() is filled with the corresponding elements. For example, our simple latch program presented earlier would be stored in the first two rows of the array as shown in Table 1.

TABLE 1

	COLUMNS								
	1	2	3	4	5	6	7	8	9
ROW 1	X1	B	NOT X1 AND 1	1	1	1	1	1	Y1
ROW 2	Y1	B	0	0	0	0	0	0	0

The only screen-drawing procedure not yet covered is PROCedit. This begins at line 2340 by checking whether the cursor is over a timer or counter and, if so, that it is correctly positioned to delete it. The rest of the routine simply erases the element from the screen and resets the appropriate element in line$ to "0".

PROCCOMPILE (AND FRIENDS)

Hitting the Copy key, from the main program loop, will call the compiling routines, beginning with PROCcompile. This routine erases the cursor and sets three variables: 'net' is a counter for each compiled network, 'row' is a pointer to the row currently being compiled and 'esc' is a program "flag" of which more later. The procedure PROCcompline is then called which will build up each complete network in turn. This procedure is, without doubt, the most difficult of all to follow yet it's basic principle of operation is relatively simple. A string variable, called str1$ is built up from the individual elements in the first row of the array line$(), with the string "AND" between each one. When a "B" (for Branch) is encountered in line$(), a second string is built up in the same way in str2$, the elements this time coming from the next row in the array. When a "B" is encountered, str2$ is considered to be complete and str1$ and str2$ are joined together with "OR" between them. The complete string is stored temporarily in the array 'strsect$' — which stands for 'string section'. In the 'latch program' example shown earlier, this string would simply be: ((X1) OR (Y1)). Note that the compiling procedure also adds brackets where appropriate. Although they are not strictly necessary in this example, in other cases they are essential; for example:

((X1 AND X2 AND NOT X3) OR (Y1 AND X4 AND 1))

Once this string section has been built up, the routine goes back to the first row and repeats the process forming a second string section. This continues until all the elements have been read from the row. All the string sections are then ANDed together and stored in the array net$. The string representing the output (Y1, Y2 etc) is stored in the array out$. The first network being complete, the routine returns to PROCcompile, where the network number is incremented (net = net+1) and PROCcompile is called again to compile the next network. The compiling process is considered to be complete when either row six has been compiled or the first element in a row is still a "0". The procedure PROCstatus is then called. This procedure begins by checking for any 'not contacts' and changing the displayed status on the screen if there are any. Commercial PLC's use the concept of 'power-flow' and, although **all** contacts are open at this time, the 'not-contacts' will be dispolayed as having power-flow through them. This is accomplished at line 3150 by calling PROCchange. This procedure simply plots a small box round the appropriate contact in its inverse colour. PROCstatus goes on to check the status of all the outputs and evaluates the logic of the networks at line 3210. If the status of the outputs, which is obtained from the array 'out' does not correspond with that evaluated at line 3210, the appropriate output status is changed on the screen by calling PROCchange. The value of the array 'out' is then changed to agree with the actual evaluated output.

Once the screen and the actual values of the outputs has been brought up to date, the keyboard is scanned. If you hit one of the number keys, the current status of the appropriate key is exclusive ORed with one. Since the initial status is zero, this results in the new status being one. The next time this number key is pressed, the result of the exclusive ORing will be zero. The net effect of this is to toggle the status between one and zero. As the number keys represent the contacts in the ladder logic, you are effectively toggling them between 'open' and 'closed'. The program will now loop indefinitely around PROCstatus until you hit the Copy key when control will be returned to PROCcompile, at line 2580. PROCreset simply clears any contacts displaying power flow and any outputs which are on. Control is then returned to the main loop at line 40. Four variables not yet explained, but which appear in PROCstatus, are reset$, current, target and preset. These are concerned with the counter/timer routines and will be explained next.

THE COUNTER/TIMER

The program allows for one counter or timer to be included in the ladder logic. It's presence is detected during the routine PROCcompline when an element in the array line$() contains "C". The procedure PROCcounter is called from the main program loop by hitting the function key f3. PROCcounter begins by setting a flag to one in order to indicate to the rest of the program that a counter or timer is present in the ladder logic. As the counter or timer requires **two** input lines, a rectangle is plotted two rows deep. The required preset value is entered at line 3910 and two variables, xtab and ytab are set to screen coordinates within the rectangle. These variables are used later to indicate where the current value of the counter or timer is to be printed. If a **timer** is specified, a string variable, tim$, is set to "X0 AND ". Although X0 is not available for use as a contact, it is nevertheless present within the program-logic, and has the special characteristic that it automatically switches between 'on' and 'off'. This is because key 0 is assumed if no key is pressed during PROCstatus, at line 3260. This automatic switching of X0 is used to clock the Start/Stop line of the timer.

PROCcompcounter compiles three strings associated with the timer/counter. The 'Start/Stop' string is compiled into the appropriate element of net$ and the corresponding output is set in the array out$ as "Y7". A second string is compiled in the next element of net$, which consists of any network between the timer/counter and it's output. A third string, reset$, is created to represent the contacts in the Reset line. These strings are evaluated during the procedure PROCstatus, between lines 3180 and 3240.

AND FINALLY...

This is one of those programs which is never really finished. As you use it, additional functions and facilities will no doubt occur to you. Because the program is reasonably structured, it should be possible to add to it without too much trouble. One facility which immediately springs to mind is to add 'real' inputs through the user-port and 'real' outputs via the printer port. Bear in mind, though, that compared with a commercial unit, it is incredibly slow. A commercial unit will whip through up to 4000 lines of ladder logic in the time it takes this program to do six. As a training aid, though, I feel that this lack of speed is not such a bad thing.

Listing for Ladder Logic Simulator.

```
  10 MODE 4
  20 ON ERROR GOTO 3700
  30 PROCinitialize
  40 REPEAT
  50 A$=GET$
  60 IF A$=CHR$(13) PROCcursorright
  70 IF A$=CHR$(127) PROCedit
  80 IF A$=CHR$(135) PROCcompile
  90 IF A$=CHR$(136) PROCcursorleft
 100 IF A$=CHR$(137) PROCcursorright
 110 IF A$=CHR$(138) PROCcursordown
 120 IF A$=CHR$(139) PROCcursorup
 130 IF A$="A" PROCcontact(X,Y)
 140 IF A$="B" PROCnotcontact(X,Y)
 150 IF A$="C" PROCoutput(X,Y)
 160 IF A$="D" PROChwire(X,Y)
 170 IF A$="E" PROCvwirel(X,Y)
 180 IF A$="F" PROCvwirer(X,Y)
 190 IF A$="G" PROCvwirelr(X,Y)
 200 IF A$="H" PROCcounter(X,Y)
 210 UNTIL FALSE
 220 MODE 7 : *FX 4,0
 230 END
 240
 250
 260 REM ---------Initializing Routines------
 270 DEF PROCinitialize
 280 B=1 : C=1 : tim=0 : Y7=0 : preset=0
 290 VDU 28,0,31,39,29 : @%=4
 300 VDU 23;8202;0;0;0;
 310 VDU 23,128,1,1,1,1,1,1,1,255
 320 VDU 23,129,128,128,128,128,128,128,128,255
 330 DIM net$(7),out$(7),out(7)
 340 DIM strsect$(8),line$(7,10),X(9)
 350 str1$=STRING$(100," ")
 360 str2$=STRING$(100," ")
 370 reset$=STRING$(50," ")
 380 PROCinit1
 390 PROCinit2
 400 FOR R%=1 TO 6
 410 FOR C%=1 TO 9
 420 line$(R%,C%)=STRING$(8," ")
 430 line$(R%,C%)="0"
 440 NEXT C%,R%
 450 PROCcursor(17,X,Y)
 460 ENDPROC
 470
 480 DEF PROCinit1
 490 row=1 : column=1 : net=1 : esc=1
 500 current=0 : target=0 : reset$="0"
 510 X=0 : Y=1000
 520 FOR R%=1 TO 7
 530 net$(R%)=STRING$(100," ")
 540 net$(R%)=""
 550 out(R%)=0
 560 out$(R%)="0"
 570 NEXT R%
 580 FOR R%=1 TO 9
 590 X(R%)=0
 600 NEXT R%
 610 str1$="" : str2$=""
 620 PROCfunction1
 630 ENDPROC
 640
 650 DEF PROCinit2
 660 X0=X(0):X1=X(1):X2=X(2):X3=X(3)
 670 X4=X(4):X5=X(5):X6=X(6):X7=X(7)
 680 X8=X(8):X9=X(9)
 690 Y0=out(0):Y1=out(1):Y2=out(2)
 700 Y3=out(3):Y4=out(4):Y5=out(5)
 710 Y6=out(6):Y7=out(7)
 720 IF esc=0 PROCprintx
 730 ENDPROC
 735
 740 DEF PROCfunction1
 750 VDU 4 : CLS
 760 PRINT TAB(0,1)" 0   1   2   3   4   5   6   7   8   9"
 770 PRINT TAB(0,2)"] [ ]/[ ( ) T/C  *";
 780 VDU 32,32,95,95,95,32,32,128,32,32
 790 VDU 32,129,32,32,32,128,129,32,32,42
 800 VDU 5
 810 *KEY 0 A
 820 *KEY 1 B
 830 *KEY 2 C
 840 *KEY 5 D
 850 *KEY 6 E
 860 *KEY 7 F
 870 *KEY 8 G
 880 *KEY 3 H
 890 *FX 4,1
 900 ENDPROC
 910
 920 DEF PROCfunction2
 930 VDU 4 : CLS : VDU 7
 940 row=row-2 : IF row<1 THEN row=1
 950 PRINT TAB(5,1)"UNABLE TO COMPILE AT LINE ";row
 960 PRINT TAB(7,2)"Hit any key to continue";
 970 VDU 5
 980 Q$=GET$
 990 ENDPROC
1000
1010 DEF PROCfunction3
1020 VDU 4 : CLS
1030 PRINT TAB(14,1)"COMPILED"
1040 PRINT TAB(3,2)"X1  X2  X3  X4  X5  X6  X7  X8  X9"
1050 VDU 5
1060 ENDPROC
1070
1080
1090 REM ---------Screen Drawing Routines-----
1100 DEF PROCinput
1110 PLOT 4,X+25,Y-100
1120 INPUT ""I$
1130 IF LEN(I$)<>2 VDU 7:PROCedit:ENDPROC
1140 L$=LEFT$(I$,1)
1150 IF L$<>"X" AND L$<>"Y" VDU 7:PROCedit:ENDPROC
1160 IF L$="Y" AND RIGHT$(I$,1)>"6" VDU 7:PROCedit:ENDPROC
1170 IF RIGHT$(I$,1)<"1" VDU 7:PROCedit:ENDPROC
1180 IF A$="B" I$="(NOT "+I$+" AND 1)"
1190 line$(row,column)=I$
1200 PROCcursorright
1210 ENDPROC
1220
1230 DEF PROCcursorright
1240 PROCcursor(19,X,Y)
1250 X=X+140 :column=column+1
1260 IF X>1200 THEN X=0 : Y=Y-150 :column=1 : row=row+1
1270 IF Y<200 THEN VDU 7 : Y=1000 :row=1
1280 PROCcursor(17,X,Y)
1290 ENDPROC
1300
1310 DEF PROCcursorleft
1320 IF X<140 THEN ENDPROC
1330 PROCcursor(19,X,Y)
1340 X=X-140 :column=column-1
1350 PROCcursor(17,X,Y)
1360 ENDPROC
1370
1380 DEF PROCcursordown
1390 IF Y<350 THEN ENDPROC
1400 PROCcursor(19,X,Y)
1410 Y=Y-150 :row=row+1
1420 PROCcursor(17,X,Y)
1430 ENDPROC
1440
1450 DEF PROCcursorup
1460 IF Y>900 THEN ENDPROC
1470 PROCcursor(19,X,Y)
1480 Y=Y+150 :row=row-1
1490 PROCcursor(17,X,Y)
1500 ENDPROC
1510
1520 DEF PROCcursor(C,X%,Y%)
1530 GCOL 0,3
1540 PLOT 4,X%+4,Y%
1550 PLOT C,136,0 : PLOT C,0,-150
1560 PLOT C,-136,0 : PLOT C,0,150
1570 ENDPROC
1580
1590 DEF PROCcontact(X%,Y%)
1600 IF line$(row,column)<>"0" VDU 7:ENDPROC
1610 IF X>1000 ENDPROC
1620 PLOT 4,X%,Y%-50
1630 PLOT 1,50,0 : PLOT 0,0,-30
1640 PLOT 1,0,60 : PLOT 0,40,0
1650 PLOT 1,0,-60 :PLOT 0,0,30
1660 PLOT 1,50,0
1670 PROCinput
1680 ENDPROC
1690
1700 DEF PROCnotcontact(X%,Y%)
1710 IF line$(row,column)<>"0" VDU 7:ENDPROC
1720 IF X>1000 ENDPROC
1730 PLOT 4,X%+55,Y%-35
1740 PRINT "/"
1750 PROCcontact(X%,Y%)
1760 ENDPROC
1770
1780 DEF PROCoutput(X%,Y%)
1790 IF line$(row,column)<>"0" VDU 7:ENDPROC
1800 IF column<>9 ENDPROC
1810 PLOT 4,X%,Y%-50
1820 PLOT 1,40,0 : PLOT 0,0,-25
1830 PLOT 1,0,50 : PLOT 1,50,0
1840 PLOT 1,0,-50 : PLOT 1,-50,0
1850 PLOT 0,50,25 : PLOT 1,45,0
1860 PROCinput
1870 ENDPROC
1880
1890 DEF PROChwire(X%,Y%)
1900 IF line$(row,column)<>"0" VDU 7:ENDPROC
1910 IF X>1000 ENDPROC
1920 line$(row,column)="1"
1930 PLOT 4,X%,Y%-50
1940 PLOT 1,140,0
1950 PROCcursorright
1960 ENDPROC
1970
1980 DEF PROCvwirel(X%,Y%)
1990 IF column>8 VDU 7:ENDPROC
2000 IF column<2 VDU 7:ENDPROC
2010 IF row=6 VDU 7 :ENDPROC
2020 IF line$(row,column)<>"0" VDU 7:ENDPROC
2030 line$(row+1,column)="B"
2040 PROChwire(X%,Y%)
2050 line$(row,column-1)="B"
2060 PLOT 4,X%+66,Y%-50
2070 PLOT 1,0,-150
2080 PLOT 1,-70,0
2090 ENDPROC
2100
2110 DEF PROCvwirer(X%,Y%)
2120 IF column>7 VDU 7 :ENDPROC
2130 IF row=6 VDU 7 :ENDPROC
```

```
2140 IF line$(row,column)<>"0" VDU 7:ENDPROC
2150 line$(row+1,column)="B"
2160 PROChwire(X%,Y%)
2170 line$(row,column-1)="B"
2180 PLOT 4,X%+66,Y%-50
2190 PLOT 1,0,-150
2200 PLOT 1,70,0
2210 ENDPROC
2220
2230 DEF PROCvwirelr(X%,Y%)
2240 IF column>7 VDU 7 : ENDPROC
2250 IF row=6 VDU 7 :ENDPROC
2260 IF line$(row,column)<>"0" VDU 7:ENDPROC
2270 PROCvwirel(X%,Y%)
2280 line$(row+1,column-1)="B"
2290 PLOT 4,X%,Y%-200
2300 PLOT 1,140,0
2310 ENDPROC
2320
2330 DEF PROCedit
2340 T$=line$(row,column)
2350 IF T$="C" AND line$(row+1,column)<>"C" VDU 7 :ENDPROC
2360 IF T$="B" OR T$="C" L%=254 ELSE L%=150
2370 MOVE X,Y
2380 MOVE X,Y-L%
2390 PLOT 87,X+140,Y-L%
2400 MOVE X+140,Y
2410 PLOT 87,X,Y
2420 IF T$="C" tim=0
2430 IF L%=254 line$(row+1,column)="0"
2440 line$(row,column)="0"
2450 PROCcursor(17,X,Y)
2460 ENDPROC
2470
2480 REM --------Compiling Routines ---------
2490 DEF PROCcompile
2500 PROCcursor(19,X,Y)
2510 net=1 : row=1 :esc=0
2520 REPEAT
2530 PROCcompline("",1)
2540 row=row+orflag+1 : net=net+1
2550 UNTIL row>6 OR line$(row,1)="0"
2560 lastrow=row-1
2570 PROCstatus
2580 PROCreset
2590 ENDPROC
2600
2610 DEF PROCcompline(L$,Z%)
2620 ptr1=Z% : ptr2=Z% : count=1 :orflag=0
2630 str1$=L$:str2$=""
2640
2650 REM.... compile 1st part of 1st line
2660 IF ptr1=9 str1$=str1$+"1":GOTO 2990
2670 str1$=str1$+line$(row,ptr1)
2680 REPEAT
2690 ptr1=ptr1+1
2700 UNTIL line$(row,ptr1)<>"1"
2710 IF line$(row,ptr1)="C" PROCcompcounter :ENDPROC
2720 IF line$(row,ptr1)="B" THEN 2780
2730 IF ptr1=9 THEN 2990
2740 str1$=str1$+ " AND "
2750 GOTO 2670
2760
2770 REM.... compile 1st part of 2nd line
2780 str1$="("+str1$+")"
2790 row=row+1 : orflag=1
2800
2810 REM.... check for illegal output inline 2
2820 IF line$(row,9)<>"0" GOTO 3700
2830 str2$=str2$+line$(row,ptr2)
2840 REPEAT
2850 ptr2=ptr2+1
2860 UNTIL line$(row,ptr2)<>"1"
2870 IF line$(row,ptr2)="B" THEN 2920
2880 str2$=str2$+" AND "
2890 GOTO 2830
2900
2910 REM.... 'or' 1st part with 2nd part
2920 str2$="("+str2$+")"
2930 strsect$(count)="("+str1$+"OR"+str2$+")"
2940 count=count+1 : row=row-1
2950 ptr1=ptr1+1 : ptr2=ptr2+1
2960 GOTO 2630
2970
2980 REM.... 'and' all parts into network
2990 out$(net)=line$(row,ptr1)
3000 str1$="("+str1$+")"
3010 strsect$(count)=str1$
3020 len=1
3030 net$(net)=net$(net)+strsect$(len)
3040 IF len=count THEN 3080
3050 net$(net)=net$(net)+" AND"
3060 len=len+1
3070 GOTO 3030
3080 ENDPROC
3090
3100 REM ----------Status Display Routines -----
3110 DEF PROCstatus
3120 PROCfunction3
3130 FOR row=1 TO lastrow
3140 FOR col=1 TO 8
3150 IF LEFT$(line$(row,col),2)="(N" PROCchange
3160 NEXT col,row
3170 REPEAT
3180 IF EVAL(reset$)=0 current=0:target=0
3190 IF current=preset THEN target=1
3200 FOR counter=1 TO net-1
3210 yvalue=EVAL(net$(counter))
```

```
3220 yptr=(VAL(RIGHT$(out$(counter),1)))
3230 IF yvalue<>out(yptr) PROCychange
3240 out(yptr)=yvalue
3250 NEXT counter
3260 A$=INKEY$(0) : A=VAL(A$)
3270 X(A)=X(A) EOR 1
3280 PROCinit2
3290 PROCdisplay("X",A)
3300 UNTIL A$=CHR$(135)
3310 ENDPROC
3320
3330 DEF PROCdisplay(D$,A)
3340 F$=D$+STR$(A)
3350 FOR row=1 TO lastrow
3360 FOR col=1 TO 8
3370 IF line$(row,col)=F$ PROCchange
3380 IF line$(row,col)="(NOT "+F$+" AND 1)" PROCchange
3390 NEXT col,row
3400 ENDPROC
3410
3420 DEF PROCchange
3430 X%=(140*col)-110
3440 Y%=990-(150*(row-1))
3450 PLOT 4,X%,Y%
3460 PLOT 0,0,-80 : PLOT 82,80,0
3470 PLOT 0,0,80 : PLOT 82,-80,0
3480 PLOT 2,80,-80
3490 ENDPROC
3500
3510 DEF PROCychange
3520 IF preset<>0 PROCcountstatus
3530 FOR row=1 TO lastrow
3540 Y$="Y"+STR$(yptr)
3550 IF line$(row,9)=Y$ PROCchange
3560 NEXT row
3570 PROCdisplay("Y",yptr)
3580 ENDPROC
3590
3600 DEF PROCreset
3610 FOR row=1 TO 6
3620 FOR col=1 TO 9
3630 IF POINT((140*col)-110,990-(150*(row-1)))<>0 PROCchange
3640 NEXT col,row
3650 PROCinit1 : PROCinit2
3660 PROCcursor(17,X,Y)
3670 ENDPROC
3680
3690 REM ------- Error Handling Routine --------
3700 IF ERR=17 GOTO 220
3710 VDU 7
3720 PROCfunction2
3730 PROCreset
3740 GOTO 40
3750
3760 REM --------Counter/Timer Routines --------
3770 DEF PROCcounter(X%,Y%)
3780 IF row=6 VDU 7 :ENDPROC
3790 IF tim=1 VDU 7 : ENDPROC
3800 tim=1
3810 PLOT 4,X+8,Y-4 : PLOT 1,130,0
3820 PLOT 1,0,-250 : PLOT 1,-130,0
3830 PLOT 1,0,250
3840 PLOT 4,X+25,Y-40
3850 line$(row,column)="C"
3860 line$(row+1,column)="C"
3870 current=0
3880 REPEAT :Q$=GET$:UNTIL Q$="C" OR Q$="T"
3890 IF Q$="T" PRINT "TMR" ELSE PRINT "CTR"
3900 IF Q$="T" tim$="X0 AND " ELSE tim$=""
3910 PLOT 4,X+20,Y-90 :INPUT ""preset
3920 PLOT 4,X+20,Y-130 :PRINT;current
3930 xtab=X+20:ytab=Y-130
3940 PROCcursorright
3950 ENDPROC
3960
3970 DEF PROCcompcounter
3980 net$(net)=tim$+str1$
3990 out$(net)="Y7"
4000 net=net+1
4010 PROCcompline("target AND ",ptr1+1)
4020 orflag=1
4030 reset$="" : col=1
4040 REPEAT
4050 col$=line$(row+1,col)
4060 IF col$<>"C" reset$=reset$+col$+" AND "
4070 col=col+1
4080 UNTIL col$="C"
4090 reset$=reset$+"1"
4110 ENDPROC
4120
4130 DEF PROCcountstatus
4140 MOVE xtab,ytab
4150 MOVE xtab,ytab-25
4160 PLOT 87,xtab+110,ytab-25
4170 MOVE xtab+110,ytab
4180 PLOT 87,xtab,ytab
4190 current=current+0.5
4200 IF current<0 THEN current=0
4210 IF current>preset THEN current=preset
4220 PLOT 4,xtab,ytab
4230 PRINT;INT(current)
4240 ENDPROC
4250
4260 DEF PROCprintx
4270 VDU 4
4280 PRINT TAB(0,2)X1,X2,X3,X4,X5,X6,X7,X8,X9;
4290 VDU 5
4300 ENDPROC
```

BOOK PAGE

Garry Marshall

This month, we look at two quite different areas of computing, education and business, with two books from each,

Two books, each on computers in education and computers in business form this month's selection. Education and business have it in common that they are two areas in which the application of computers has produced results that range from the highly successful to the downright disappointing. Any one of these books, to say the very least, could help to prevent the introduction of a computer from being a complete disaster.

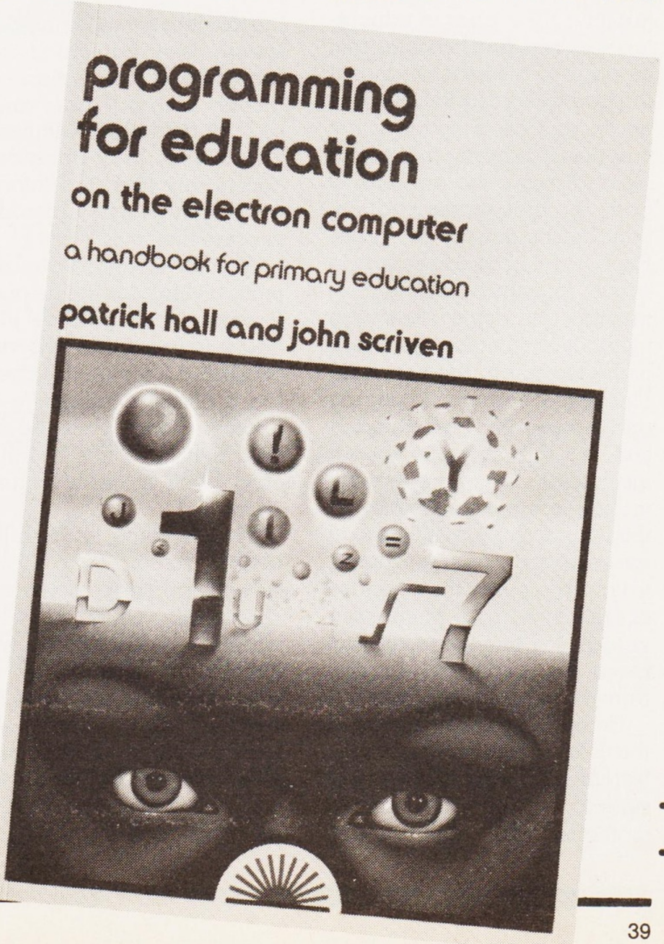
The prime excample of the successful use of the computer in teaching and learning is, to my mind, the flight simulator. The fact that a jumbo jet pilot can receive a complete training from a simulation is well established. And if it is possible to learn such a complicated skill from computer-based training, then it is clear that there is very little restriction on what it is possible to learn in this way.

In **Programming for Education** by Patrick Hall and John Scriven we can see some of the early steps that are being taken towards using computers effectively in primary schools. It is essentially a collection of programs for use by children to help them grasp specific skills and concepts. Many of the programs are computerised ways of achieving what can already be done successfully in the classroom by conventional means. A lot of the programs will be familiar; for example, Hangman, Guess a number, and a sentence generator are all here. The saving graces are that the programs all include an interesting twist or variation, and that graphics are used to good effect. But the thought persists that those unearthly, piercing blue eyes on the cover are looking for rather more original uses for the computer than any that the book presents.

Each chapter of the book, except the first and the last, contains about three programs connected with a particular theme, such as arithmetic skills, word skills and keyboard familiarity. Each program is preceded by a rather sketchy account and justification, and is followed by a commentary, again more in the form of notes than anything else, which describes the function of each part of the program. The introductory accounts are interesting for their explanations of the educational aims of the programs. The way that the educational needs of a child of a particular age are catered for by a program are explained. The way that a correct response is rewarded by the use of graphics is dealt with, as is the reaction to an incorrect response. The reaction to the latter must not be discouraging, nor must it be more spectacular than the reinforcement for a correct response. If it is, then the tendency will be for children deliberately to give the wrong answers. The authors' experience as teachers is abundantly apparent in their knowledge of the theory of learning and in their awareness of what children find difficult to learn and of what they enjoy learning.

The programs are all written for the Electron. This is a particularly good choice of computer since the BBC Micro is by far the most common in schools and, as the cut-down and therefore cheaper version of the BBC, the Electron can be bought for the home while complementing the computer used at school. Consequently, the book can be used both at home and at school. The programs themselves are written clearly and with good structure. The careful and consistent use of thoughtfully named procedures gives main, controlling, programs that can be read and understood directly because they just call procedures, and the purpose of each procedure is apparent from its name.

Chapter 3, called 'Starting with maths' contains three programs that are basically arithmetic problem drill programs. At heart then, the ideas for the programs are unremarkable and of a type that, for educational value, is rather derided. But the first program incorporates a timer so that it can be used competitively, the second includes some spectacular graphic effects to reward correct answers, and the third is written as a video game in which attempts at the answers can be 'bombed' from a plane flying above them. In similar vein, chapter 5 contains programs that test basic spelling and comprehension skills: they are for Hangman, for generating anagrams, and for finding opposites with a game in which a word can be steered around the screen avoiding unsuitable words, or blasting them out of the way, until it is located next to its opposite.



Besides being well written, the programs contain some ingenious fragments, one or two of which may be worth repeating despite the fact that they will be more or less familiar to experienced programmers. In the anagram program, a word is selected at random from a list of 20 in its DATA statements by:

```
RESTORE
FOR N=1 TO RND(20):
READ word$: NEXT N
```

An anagram for the word stored under the name word$ is created and stored under the name dummy$, rather ingeniously, by:

```
dummy$=word$: new$=""
FOR N=1 TO LEN(dummy$)
IF RND(2)=2 THEN new$
=new$+MID$(dummy$,N,1)
ELSE new$=MID$(dummy$,
N,1)+new$
NEXT N
dummy$=new$
```

Most of the chapters containing programs are of a similar nature to the two we have looked at. The last three of them mention in their introductions the topics of artificial intelligence, logical reasoning and problem solving, and simulation. Disappointingly for the reader, and by the authors' own admission, the programs presented in these chapters do not really illustrate anything of the use of these topics.

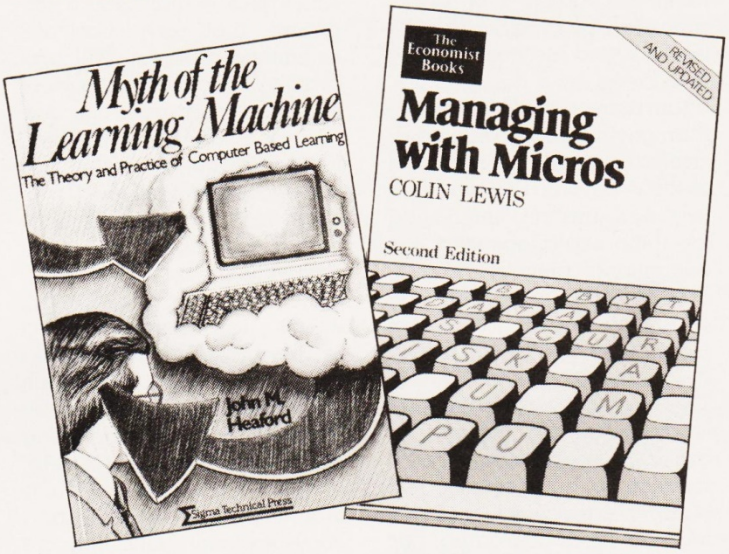
All of these topics are undoubtedly going to play large parts in any successful future application of computers in education. At least the authors are aware of this, even if with their present book they are not in a position to show us how.

Myth of the learning machine by John Heaford is also about using the computer to educate, and it often touches on the same issues as the previous book. But it provides a complete contrast, for if we can say that the previous book is the do-it-yourself offering of two teachers, this one is by a professional educational technologist.

The myth referred to in the title is the one that begins with 'Computers can never...'. It is the author's conviction that, in education and training at least, they can do a great deal. He proceeds to show us what they are doing now, and what they could be doing in the future.

The author explains why he thinks that schools are not fulfilling their roles in educating children, and how they can manage to disassociate learning from everyday life. He goes on to give an account of the theories of learning that are so often disregarded in schools and, for that matter, educational institutions of all kinds. From a firm grasp of the way in which learning occurs, it is possible to devise teaching methods that match the needs of the learner in the best way. This applies equally to schools and to computer-based education. But if formal educational institutions are not prepared to make the effort needed to do this, then the developers of software for computer-based training can. And the presence of the micro in the home, as a vehicle for this software, can move the centre of learning from the school to the home.

Heaford gives examples of the use of computers to provide training in ways that are superior to conventional methods. Not surprisingly, perhaps, many of them relate to the provision of training in areas of the new technology. One of them is in training maintenance staff for the Flight Management System of the Boeing 757s operated by British Airways. These aircraft have instruments that are entirely electronic — there are no mechanical dials with rotating arms, just CRTs displaying readings. This change of practice caused a tremendous need for retraining, which was met by using a computer-based simulation of the cockpit. Another example is the training of the users of the computerised airline reservation system BABS. This was done by a training system that allowed the trainees access to the BABS system itself, but which could recognise a transaction originating from a trainee and so did not actually implement it. That is to say, the trainees could access any files and information in the system, but could not change them.



After reviewing existing systems, a view of what the author considers to be possible in the future is presented. His vision consists of a model incorporating artificial intelligence and expert systems. It may be some time before we see it, but the case for it is made convincingly.

This is quite a high-level book, containing a wide spread of material. Some of it is, necessarily, rather dull. The material on testing the effectiveness of educational software, for example, is less than rivetting. But without some form of testing, claims about the effectiveness of the software cannot be supported. The book provides as good an overall account of computer-based training as you will find. Skip the first chapter, though. The author was either feeling his way into the writing of the book or he stuck the text of a lecture in which he was trying to impress some august body at the front rather than write an introductory chapter.

Turning now to the business books, the first is **BASIC in business** by Arnold Handley. It is in essence an introduction to BASIC, but with all the illustrations and example programs drawn from business applications. The author is a businessman, and successfully addresses himself to other businessmen. The book has a quite different flavour to most of the other introductions to BASIC as a result. The example programs include accounts, stock control, invoicing and record keeping. As you would expect, the book contains much more on files and file handling than other comparable introductions.

Actually, the book is more than an introduction to BASIC, as it deals with disk usage and disk operating systems. These are obviously required in any realistic business computer system. Handley's coverage is broad, not dwelling on any particular DOS, but sketching their general purpose and usage. At this level, the coverage it provides is adequate.

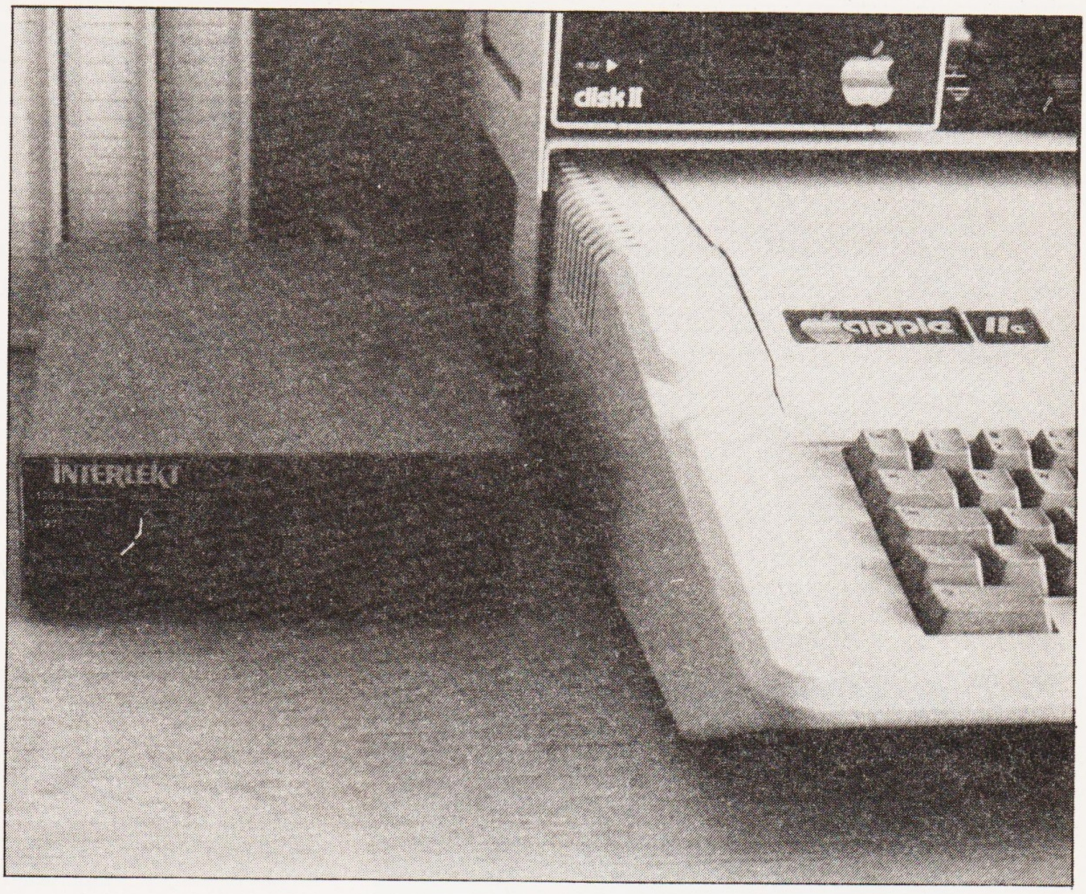
The overall style of the book is bright and breezy, and conveys the author's obvious enthusiasm. This more than compensates for a few rough edges in the coverage of programming. I particularly liked the chapter on how to program when you can't program. The suggested way is to merge, append and adapt other programs to build a program of your own that does what you want.

I rather doubt that this book would be much help to a businessman still considering the use of a computer in business. Its emphasis on learning BASIC is too heavy, but its enthusiasm and fund of good sensible advice ensure that it will be of value to those who insist on taking the do-it-yourself route.

By contrast, **Managing with micros** by Colin Lewis is much more helpful because it concentrates on the uses to which a micro can be put rather than on the micro itself. After introductory chapters on the hardware and software of micros, and on the microcomputer industry, it presents a series of chapters on the items of software that make it possible to employ micros usefully in business. Word processing, spreadsheets, database management, ledger systems, stock control and payroll are all dealt with. The purpose and practice of each is explained. Examples are presented with reference to typical packages, such as Wordcraft, VisiCalc and dBASE II.

This approach seems eminently suitable, and the book can be thoroughly recommended to any business person interested, or involved, in computerisation. The explanation of the purpose of a spreadsheet and the illustration of its use is particularly good. But then the whole book is of a high standard, although one or two of the judgements, on after sales support, for instance, strike me as rather generous. ■

This month's books are:
Programming for education by Patrick Hall and John Scriven (Sunshine), 209 pages, £5.95
Myth of the learning machine by John M Heaford (Sigma Technical Press), 236 pages, £9.00
BASIC in business by Arnold Handley (Newnes microcomputer books), 264 pages, £8.95
Managing with micros by Colin Lewis (Basil Blackwell), 200 pages, £6.95.



TX/RX

Henry Budgett

As a follow-up to our introductory article, we take a look at a full-facility modem of the type now generally available.

No man, someone once said, "is an island". Whilst humans overcome their isolation from one another by using verbal or written communication, computers find things rather more troublesome! Joining one machine to another in the same room or building isn't too much of a problem, given that each has a serial interface and a suitable piece of software to allow information from one to be passed to the other in a form it can understand. The skill comes in when you find that your two computers are unable to talk to one another because their so-called standard serial interface (RS232 is the number to watch for here) are different to one another! Many a Luddite has been created through the manufacturers liberal interpretations of what was once a perfectly good set of rules.

Long distance communication for humans is easy, pick up the telephone and you can talk to almost anywhere in the world but until recently connecting one computer to another over the telephone lines has been slightly more awkward.

The recent breaking of the virtual British Telecom monopoly on attaching equipment to telephone lines has resulted in a veritable deluge of equipment from independent manufacturers. Much of this has capitalised on the recently introduced "World Chip" single chip modem which can generate both European (CCITT) and American (Bell 103) tones as well as handling a variety of baud rates from 300 to 1,200 including the split rate of 75/1,200 used by Viewdata systems such as Prestel. Several manufacturers have, however, failed to gain the necessary BABT approval – signified by a green circle as opposed to a red triangle – because they include the Bell frequencies which can play havoc with some UK telephone exchanges.

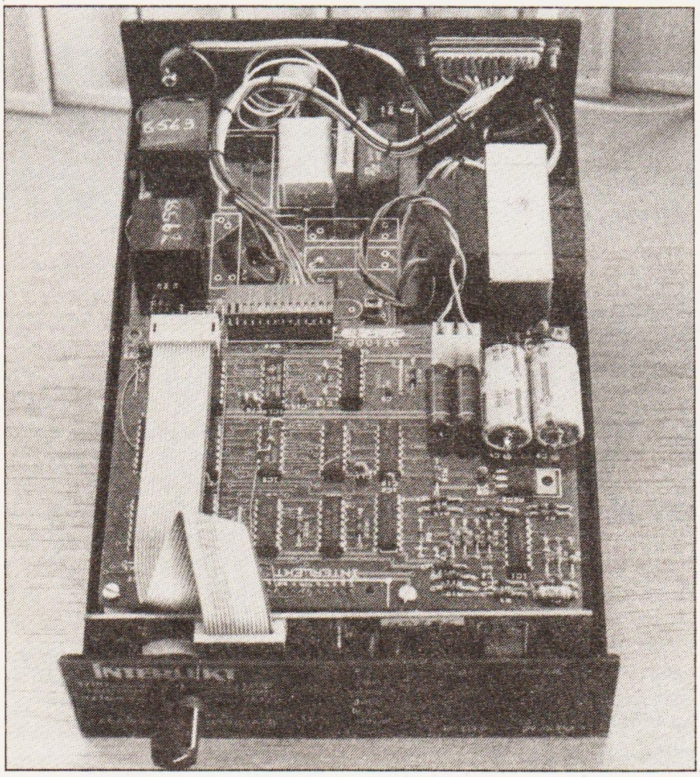
Quite why anyone who's serious about using modems for communication thinks they need the Bell option is a bit of a mystery. The obvious answer is that they want to get access to the American bulletin boards but if they are that serious they should be using PSS or some similar system which does it for them – and cheaper too!

Packaged in a neat orange and black metal box 265mm by 170mm by 65mm, that's big enough to sit the 'phone on, the Portman is quite bulky but very well built. The front panel is made up of a plastic plate over the labelled metal front of the box and consists of a rotary switch on the left with a single toggle switch on the right. Compare this simplicity of operation to some of the Portman's rivals which appear to have more knobs and switches than any human would know what to do with. The rotary switch selects the baud rate for either answer or originate modes while the toggle switch engages the auto-answer mode or connects the modem to the line in manual mode. In the centre of the panel is a column of five LEDs which indicate the state of the device. From bottom to top these are Power, Data Terminal Ready (your modem ready), Data Carrier Detected (the other modem ready), Received Data and Transmitted Data. The only other indicator on the front panel is on the toggle switch and is labelled Data. This, sadly, is not an accurate indication of what is going on and it should be labelled Online as it really shows whether the modem is connected to the 'phone line or not.

On the back panel there is the power lead with its accompanying fuse – shame it wasn't socketed – the direct connect 'phone lead and an earth terminal. To strictly comply with the regulations concerning the connection of external equipment to the telephone lines this should be attached to a protective earth – something like a mains water pipe just below the stopcock or a large copper plate buried about 12" underground... The most delightful sight of all, however, is a real RS232 25-way D-type socket. Unlike many rival modem manufacturers who have used such non-standard connectors as DIN sockets and the like, Interlekt have stuck firmly to the proper item.

CONSTRUCTION

The internal construction of the Portman was of an extremely high quality. There are three PCBs packed into the case along with the power supply transformer but there's still plenty of room and no sign of overheating. The main PCB contains the actual modem, based as expected on the AM7910 or World Chip, while the main operating controls are fitted onto a second PCB mounted behind the front panel of the case. The third board holds the auto-answer circuitry and mounts over the main PCB and connects to it by a header plug/socket arrangement. Of the three this is the only board to have visible 'patches', a significant number in fact, but as the unit was a demonstration one and known to have been 'doing the rounds' this was not too surprising. All the boards were well laid out and the main PCB seemed to have a large number of unfilled options. One of these appears to be a DIP switch which would allow the user to get at the 'missing' features of the World Chip. No details are given in the manual but Interlekt do offer certain of these options 'factory fitted'.



Connecting the modem to a computer, in this case an Apple //e, could hardly be simpler. One standard male-to-male RS232 cable with pins 1 to 8 and pin 20 connected joins the Portman to the Apple Super Serial Card in seconds. The acid test was to try it out, surely it couldn't really be that simple!

COMMUNICATING SOFTWARE

To get information out of a system down the telephone line you need software. As there are more than a few packages on the market at the moment which offer the necessary facilities it might seem unfair to pick a single one to try but in the event I selected Apple's own Access II package which came highly recommended. All that remained, therefore, was to find the telephone number of a suitable bulletin board system and dial it up. At this point I must confess to a certain degree of cynicysm – you mean you hadn't noticed? – and fully expected to spend a day trying to get it all to work.

In the event all my fears were dispelled by the fact that it all worked first time and every time. Even some of the bulletin board operators I chatted with down the lines expressed amazement at the instant success and these are hardened individuals! My thanks to these brave people who put up with an itinerant reviewer hacking his way through their systems and especially to the operator at BABBS1 who was more than a little help.

BULLETIN BOARDS

Enough of the personalities, what about the equipment? To use an on-line system such as one of the many BBSs all that was required was to run the Access software and put it into terminal mode. In almost all cases it is worth having the 'record' option on how a disc copy is made of everything you type or receive. This can be printed out later and makes life easier than having to scribble things down on a notepad while trying to read the screen. Incidentally, it does help to remember to set the Apple to use the XON/XOFF protocol or you lose bits of text while the file is being updated – one lives and learns...

FACTSHEET	Interlekt Portman £200 (approx.)
Type	Multi-rate direct connect modem Supports 300 full duplex 1200 half-duplex 75/1200 full duplex Full originate and answer Auto answer available in all modes
Supplier	Interlekt Electronics 24 Portman Road Reading Berkshire RG3 1LU 0734-589551

Once the software is running simply dial the BBS number, wait for the carrier tone (a high-pitched whistle) and flick the toggle switch to the Online position. You can replace the handset on the 'phone and start communicating. In only one case did I have trouble and that was nothing to do with the hardeware at all. I suppose you should expect problems when there's a thunderstorm raging but when the line went dead halfway through a conversation the Portman quietly waited its allotted nine seconds and disconnected exactly to the rulebook. The software then recognised that the line was down and returned to my control, a far cry from some systems I could name!

AUTO ANSWER

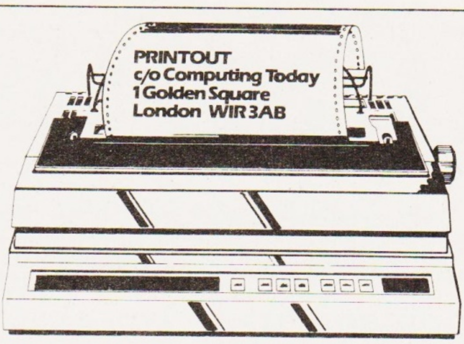
As well as using the modem to dial out it can also be used in auto-answer mode and although the test I tried was not really 'real' the system detected the ringing and answered the line perfectly. It also detects if the carrier isn't there – a human trying to talk to you rather than a machine – and drops the line after the designated nine second delay. The Access II software does incorporate a ring detect as well so, in theory at least, you could program the system to answer data calls at certain times but free the 'phone for speech at others.

Unfortunately Access won't support split baud rates so I couldn't test Prestel with the normal 75/1,200 connection but, just for fun, I tried the London-only 300 baud option and this worked fine. If you don't mind losing all the graphics, that is! Both the modem and the interface card will support split rates so there is nothing to suggest that it won't all operate as you expect given that the right software is loaded.

CONCLUSION

To say that the Portman is anything other than an excellent device would be a gross injustice. It is certainly a little more expensive than some of its competitors but then one could reasonably expect to pay for a quality product, which it certainly is. There are only two points that this review has raised which may, or may not, be regarded as quibbles. At least one of its direct competitors offers auto dialling and although the mechanics of this often leave much to be desired I would hope that later versions are offered with this as an option for the really lazy user. The only other possibility is that the manual should really be extended to include more detail for the technically oriented user but both these points are minor considerations really.

PRINTOUT



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

THE INTELLIGENT COMPUTER

Dear Sir,
I write with regard to the article 'Computer Intelligence' which appeared in your February '85 edition. Disregarding the fact that the article was attributed to two different authors — Bill Horne, according to the contents page, and Don Thomasson, as the article was headed — I must state that the author (whoever he may be) is taking an unnecessarily pessimistic and negative view of the world of Artificial Intelligence. In some cases he is just plain wrong. For example, although he quite rightly points out that hardware allowing a computer to discriminate between different colours: "...would not be difficult to implement", he continues: "...linkage of colour to names would be difficult". Nonsense. An electronic spectroscope is designed to detect such spectral differences, and the transmission of spectroscopic information to a computer would not be difficult at all. Similarly, I would disagree with the authors use of the term 'spontaneous' when referring to human responses to external, or other, stimuli. He compares, for example, the 'Syntax Error' message to a human exclamation, boldly concluding that a human utterance or some similar response is entirely spontaneous, whereas an error message is not. It is obvious that what the author is questioning here is not the 'spontaneity' of the message, but the 'sapience' and predictability of it. I would hate to spoil the authors fundamentalist illusions, but I contend (as did Pavlov) that ALL human responses are programmed to a greater or lesser degree, in much the same way as the computer's messages are. However, ***Homo Sapiens*** as a race possess such an enormous repertoire of 'conditioned responses' that an illusion of spontaneity is created.

But I would agree with the author when he asserts that: "AI in the true sense of the word is still a pipe-dream". Yes, and it will remain a pipe-dream for as long as mankind continues with his attempts at implementing a working version of a system that has yet to be adequately specified. I do not doubt that current research into AI will bring many benefits, but I feel that more time, energy and money could be spent in other disciplines such as psychology and philosophy, in an effort to provide the software engineer with a suitably rigorous definition of that which we strive to give our machines — intelligence.

In conclusion, I must suggest that the article was deliberately antagonistic — and, perhaps, ill-researched.

Yours sincerely,
Thomas Weybridge,
Hebden Bridge,
Lancashire.

The Editor replies:
Thankyou for your lengthy and obviously well-considered letter, Mr Weybridge. Bill Horne, the author, was expressing an opinion in his article more than anything, just as you have in your letter. The article was expected to elicit some angry responses, but it was certainly not designed to purposely antagonise our readers. Bill Horne is probably penning his reply at this very moment, so watch this space...

PDS AND PRAXIKOSIS

Dear Sir,
The British public suffer from national technikosis (aversion to new technology, particularly computers). So a computer program in the news is a rare event.

Since NCB and NUM negotiators used the Priority Decision System (PDS) developed by Work Sciences Associates and Brunel University to "produce a workable joint solution to the miners' strike", we have seen this decision support system analysed on TV, Radio and Press (ITN and Thames News, the BBC ***Today*** programme, Capital Radio, ***Radio Times, New Statesman***, etc).

Yet we can find no professional review of this software in your journal. Perhaps British technologists suffer from national praxikosis (aversion to practical program applications).

Yours sincerely,
V. Harcourt,
Penta-Daxo Associates,
82 Shaftesbury Avenue,
London W1.

The Editor replies:
The Priority Decision System is featured this issue, in an article provided by two of the people at Brunel who were involved in its development. It is perhaps unfortunate that the PDS has not received wider coverage, but I feel sure that Computing Today *was not the only journal that was ignored when details of the system were released; the first we knew of the system's existence was its use during the NCB/NUM dispute, as reported in the national press.*

MALIGNED MEMOTECH?

Dear Sir,
On reading your magazine I was disappointed to find Memotech computers featured very little. As a proud owner of an MTX512, I feel that you are not doing justice to Memotech or your magazine by ignoring it in this way. The MTX500/512, when compared with other machines that you feature regularly, is as good and in some cases better. Please, in your future issues, feature the Memotech machines more prominently.

Yours faithfully,
R. Brooks,
Devon.

The Editor replies:
While we have done our best to cover the range of popular micros, it has been inevitable that some machines have received greater attention than others, by virtue of the way in which Computing Today *has been constructed in the past. Many of the programs and articles published here are submitted by readers like yourself. Therefore, if we do not receive programs from Memotech owners we are clearly unable to publish material for this machine. However, we are now using a new approach which should satisfy the needs of our readers, whatever machine they might own. By publishing more ideas, suggestions, algorithms, and the like, whilst strictly limiting the number of freestanding, machine-specific programs, we hope to reach the maximum number of readers, without running the risks of excluding some who do not own a particular machine. Incidentally, readers interested in submitting articles to* Computing Today *are referred to the SUBMISSIONS invitation which appears on page 5. This should give a clearer indication of our direction from now on.*

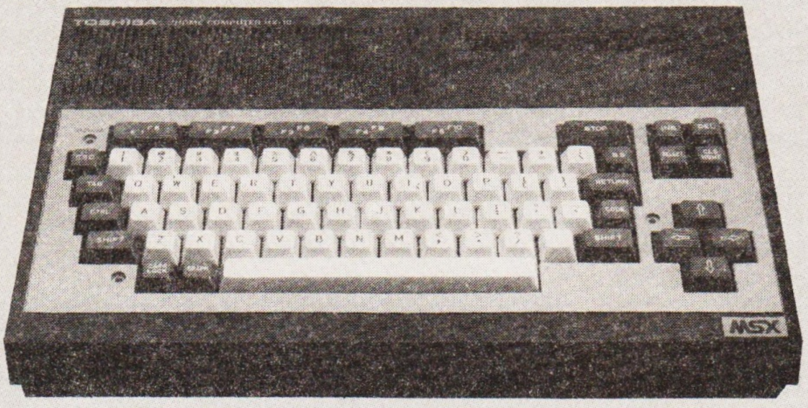
AN IMPORTANT MESSAGE

Would Mr Gordon Mills and Mr W Henderson kindly contact us regarding payment for work published in **Computing Today**
— Ed.

TOSHIBA MSX

Peter Freebrey

Ignore for a moment the 'clever' (yawn) 'Hello Tosh' advertisement currently gracing our TV screens — take a look at what the company is offering the home-computer market: The Toshiba HX-10 MSX-compatible microcomputer.



As I'm certain a lot of you will be sick of hearing, the MSX machines all have the standard Microsoft MSX BASIC. This means that in theory, programs written on one MSX machine will run on all other MSX machines. To date this does seem to be true — even those utilising machine code calls to the operating system, this is pleasantly surprising as I believe the rules and conventions for Microsoft MSX BASIC did not call for an exact 'one-to-one' memory location compatibility.

Various MSX machines have already appeared that offer 'just that little bit more' — Sony with its built-in ROM filing package etc., Yamaha with additional music/synthesizer commands, and JVC/Pioneer with special accessing for video disc control etc. But, all still have MSX BASIC compatibility. The Toshiba HX-10 has steered a neutral course, offering nothing exotic — just a solid, well-made and designed, basic machine.

STYLING

The HX-10 is rather 'box-like' with only minimal slope to the keys. The keys are slightly dished and feel comfortable in use, although a greater slope would probably be preferred by the typists among us! The general styling is fairly unobtrusive, and reasonable use of different coloured keys makes for convenience rather than gaudiness or colour for colour's sake. The main character keys are white with clear black lettering, while most of the control keys are dark grey (with white lettering). The double-sized STOP key is red, whilst the GRAPH key and cursor keys are green and blue respectively.

The function keys are double size and easy to operate quickly. TAB, CTRL, SHIFT, and CURSOR keys are all larger than character keys, and it does seem a shame that BACKSPACE, INSERT, DELETE and HOME were not also given this treatment, as they too will be used frequently.

Generally, the feel and layout of the keyboard was easy to get used to, but even after extended use I found that I did not always get a positive response from the key operation. Key movement is quite reasonable but the keys have to be depressed fully, to be certain of 100% contact being made.

Two indicator lights show 'power on' (red) and CAPS LOCK (green), the only other features on the top of the machine are the cartridge slot — sensibly protected by a lightly sprung cover — and the ventilation grill. The latter has a simple baffle to frustrate the entry of unwanted objects! Although 'on-board' power supplies may suggest heating problems, the Toshiba HX-10 runs for many hours with no appreciable increase in running temperature after the first 15 minutes or so.

The right hand side of the machine has two standard joystick sockets and the now accepted MSX Centronics printer socket (where do you get plugs to fit this?). The rear has a 50 pin expansion bus connector (with screwed-down cover), outputs for composite video, audio, R.F. for direct connection to a standard PAL TV receiver and an eight pin DIN socket for a cassette recorder. Finally, on the left hand side there is the On/Off switch. Cables are provided for connection to cassette (with three jack plugs) and to the television.

Some MSX machines have two cartridge sockets. Toshiba, in providing one cartridge and one expansion bus have probably made the more immediately useful choice.

DOCUMENTATION

The documentation includes the Owner's Manual and an MSX BASIC Reference Manual. The former deals with initial setting up, editing, simple programs, and the use of the Function and Control keys. The Reference Manual is a pretty comprehensive explanation of the major attributes of MSX BASIC, together with individual sections on each of the (nearly) 200 keywords and commands. Very brief examples accompany each entry, which go some way to help explain their use within the program.

There is no doubt that the Reference Manual is one of the better manuals around. But, although it helps the user who has some experience, it may prove quite a mouthful for the newcomer. Like many such manuals it does not attempt to explain what you can do with the machine, only how to do it once you know what it's all about!

For instance, although it is quite easy to redefine the character set in text modes 0 and 1, this information is not mentioned anywhere. In these two modes the data for the character set is copied from the BASIC ROM into the Video RAM, and may then be altered. BASE(2) and BASE(7) give the start address in VRAM for modes 0 and 1 respectively, where this information is stored. So, for mode 0 BASE(2)+ASC("A")*8 will give the first address of the 8 bytes defining character 'A'. These may then be VPOKEd with your own data to create new shapes.

The demonstration tape that comes with the HX-10 contains no real surprises or particularly exceptional material. It shows some of the more obvious features of the machine (sprites, sound, etc.) in action, and its greatest asset is that the programs are unprotected and therefore give the user an opportunity to study how the graphics and sound capabilities may be used.

It also demonstrates that the cassette recorder signal must be correctly adjusted, as it is possible to CLOAD something that resembles a good program, but is in fact just so much gobbledegook! This is usually quite obvious on inspecting the program, but at the back of my mind is the thought: what if only a small corruption took place?...

In fairness, the latitude of acceptable signal seems quite reasonable, but this is certainly an area to keep an eye on. I prefer those systems that crash out rather than those that carry on regardless!

TABLE 1

	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8
HX-10 (Integer)	0.91	3.74	—	—	—	—	—	—
HX-10 (Single-precision)	1.81	5.45	16.4	17.8	18.8	29.6	41.9	208
HX-10 (Double-precision)	2.04	5.72	16.7	18.1	19.2	31.6	44.6	207.5
Commodore 64	1.20	9.30	17.6	19.5	21.0	29.5	47.5	113
Sinclair Spectrum	4.9	9.0	21.9	20.7	25.3	68.2	86.7	250
BBC Micro	0.8	3.1	8.2	8.7	9.1	13.7	21.3	53

Result of benchmark tests

BASIC

Microsoft MSX BASIC is a highly defined and very well documented language, and is the whole reason behind the present series of machines. I have to admit, that when I first read about MSX I felt that it was in some ways a retrograde step. This had nothing to do with the various arguments regarding what some people consider as out-of-date technology — I have always felt strongly that if an operating system/processor chip does the job I want it to... Hang the critics who always talk about the leading edge of modern technology! — give me something that is going to continue working reliably! No, my adverse feelings towards MSX were based on the feeling that new computers stimulated ideas for improving the facilities and commands available to me the user, thus moving slowly towards my ideal computer. Having now had a chance to handle an MSX machine at first hand, I've had to change my opinions somewhat. Not that MSX is perfect — far from it — but not everything hinges on what a machine offers at face value when it is first introduced — look at the Spectrum two years ago and what it then appeared to offer, then look at what current programs can make it do!

MSX BASIC is reasonably comprehensive. It might not offer Procedures or some of the more advanced loop structures (REPEAT..UNTIL), but it certainly has a lot going for it. The present crop of programs, with one or two exceptions, should not gain many 10-out-of-10's, but then this is true of nearly every new machine — original BBC, Spectrum and CBM64 software was pretty ghastly in the first few months of these machines lives too!

OVERVIEW

As this is Computing Today's first MSX machine review, let's take a brief overview of what MSX BASIC can offer in the Toshiba HX-10 environment:

Firstly, how easy is it to enter programs? The HX-10 has AUTO line numbering; your choice of first line number and increment. Make a mistake, then you will have to edit the program; here you have a full screen editor, correct a mistake anywhere on the screen... just overwrite the wrong entry with the correct one. You want to insert? Then position the cursor where you wish to insert, press INS and type away...use of the cursor controls or RETURN takes you out of the INSERT mode.

Back Space and Delete perform similar functions. The former deleting the character to the left of the cursor, the latter deleting to the right.

Once learned, there are a number of useful CTRL functions that can help too: for example CTRL and 'B' repositions the cursor to the first position of the preceeding data item; CTRL and 'E' deletes all characters in a program line to the right of the cursor; CTRL and 'F' is similar to CTRL and 'B' but moves the cursor to the right; CTRL and 'N' repositions the cursor to the end of a program line (not to the bottom row (?) as suggested in the manual). There are numerous other examples — the only problem is in remembering which is which!

Should you have made a rather major mistake (never!) then the HX-10 has block DELETE, and also a RENUMBER facility (full or partial — you may specify from what line renumbering is to begin, and the size of the increment). Also, just a small point but not mentioned in the manuals — there is no need to type CLS and RETURN, to clear the screen...SHIFT and HOME does the trick very nicely thankyou!

TRON and TROFF (TRace ON/TRace OFF) are also supported. Although at first sight this is a useful addition, the implementation is such that unlike some specialised 'toolkits' that display small windows showing line numbers being 'actioned' — The HX-10 prints line numbers [in square brackets] wherever the cursor happens to be! So, in many cases when TRON is used, you tend to simply get a screenful of line numbers.

Program lines may contain up to 255 characters and the ten function keys (five plus their shifted counterparts) may be programmed with up to fifteen characters, including RETURN. Function key assignments may be shown at the bottom of the display-screen as a reminder.

The operating speed of MSX BASIC is comparable with the CBM 64,, somewhat faster on some operations, slower on others. Where integer variables can be used, a significant speed increase is made — almost challenging the BBC micro! Figures for the standard benchmarks are shown in Table 1, together with those for the CBM 64, Spectrum and BBC for comparison.

GRAPHICS

The HX-10's graphics capabilities seem to be a bit of a mixed blessing! On the one hand, they are very easy to use, but on the other hand they do not appear to be as versatile as one may wish. There are four modes of operation; two text modes and two graphics modes. These are selected by the SCREEN command (which also specifies sprite size, key click switch, cassette baud-rate and printer selection switch!).

In mode 0 the HX-10 has a screen display of 40 columns wide by 24 rows deep. The width of the display is definable and 'power up' is in the default condition of 37 characters by 24 rows. Each character-cell consists of a 6-by-8 dot matrix and no sprites are available in this mode. Some of the graphics characters that may be called from the keyboard require an 8-by-8 matrix, and these will appear with the far right portion missing.

Mode 1 has a display of 32 columns by 24 rows, again with the width definable — default values are 29 characters by 24 rows. Here each character cell is on an 8-by-8 matrix, so all graphics characters are shown in full. Sprites may be called in mode 1.

The manual states that in both text modes, the display area 'may only have two colours specified': one for the text and the other for the background (SCREEN 1 additionally allows you to define the border colour). This is not strictly accurate, as in mode 1 the colour of each consecutive group of eight characters is held in the video RAM and it is possible to reset these values by VPOKEing the colour table starting at BASE(6). This will allow a more flexible colour display but obviously requires a careful choice of characters or alternatively, their redefinition as suggested above.

In these two text modes you cannot utilise any of the additional graphics commands such as CIRCLE, DRAW, LINE, PAINT, etc...

Mode 2 (SCREEN 2) is a high resolution graphics mode with a resolution of 256-by-192 pixels. All 16 colours may be displayed to the screen with a horizontal colour resolution of 8 pixels, so each group of pixels 0-7, 8-15, 16-23, etc. may only have two colours specified. Vertical colour resolution is one pixel — so each row may be a different colour combination to either adjacent row.

Mode 3 (SCREEN 3) is a multi-colour mode which has a resolution of 64-by-48 pixels — this effectively displays a quarter of what would be displayed in mode 2 (but magnified to fill the screen) and could well be described as a 'chunky' or low resolution multi-colour mode, as each pixel (each consisting of four mode 2 pixels) may be any of the 16 available colours.

Whilst in either of the two graphics modes, the normal text command PRINT is not accessable, but the screen may be OPENed as a file, to which text may be printed using PRINT#. Graphics commands such as CIRCLE, LINE and

DRAW are very flexible, readily allowing the operation of circles, arcs, ellipses, lines, rectangles, etc.. 'Straight' circles are in fact not very circular and drawing an ellipse with a ratio between two axes of 1:1.2 is required to correct this situation! Complete figures may be easily 'filled', either directly from the CIRCLE/LINE command, or if an irregular shape, with PAINT.

SPRITES

MSX computers also score highly with the ease with which sprites may be created and moved around, using SPRITE$ and PUTSPRITE commands. Likewise the detection of a collision using the interrupt command ON SPRITE GOSUB. There does not appear to be an automatic declaration of what sprites have collided, and at present I cannot find the appropriate memory location! So, until someone comes forward with this valuable piece of information, the routine following the interrupt will have to do this operation in BASIC by looking at the PUTSPRITE variables.

TABLE 2

List of MSX BASIC keywords

ABS-
AND
ASC-
ATN-
AUTO-
BASE
BEEP
BIN$-
BLOAD
BSAVE
CALL
CDBL-
CHR$
CINT-
CIRCLE-
CLEAR-
CLOAD
CLOAD?
CLOSE-
CLS
COLOR-
CONT-
COS-
CSAVE
CSNG-
CSRLIN-
DATA-
DEF DBL
DEF FN
DEF INT
DEF SNG
DEF STR
DEF USR-
DELETE-
DIM
DRAW
ELSE
END-
EOF
EQV
ERASE-
ERL-
ERR-
ERROR-
EXP-
FIX-
FOR-
FRE-
GOSUB-
GOTO-
HEX$-
IF-
IMP

INKEY$-
INP
INPUT-
INPUT$-
INPUT#
INSTR
INT-
INTERVAL
KEY-
KEY LIST-
KEY ON/OFF
KEY ON/OFF/STOP
LEFT$-
LEN-
LET
LINE-
LINE INPUT-
LINE INPUT#
LIST-
LLIST-
LOAD
LOCATE-
LOG-
LPOS-
LPRINT
LPRINT USING
MAXFILES-
MERGE
MID$-
MOD
MOTOR
NEW-
NEXT-
NOT
OCT$-
ON GOSUB-
ON GOTO-
ON ERROR GOTO-
ON INTERVAL GOSUB-
ON KEY GOSUB-
ON STOP GOSUB-
OPEN
OR
OUT
PAD
PAINT-
PDL
PEEK-
PLAY-
POINT-

POKE-
POS-
PRESET-
PRINT-
PRINT#
PRINT#USING-
PSET-
PUT SPRITE
READ-
REM-
RENUM-
RESTORE-
RESUME-
RETURN
RIGHT$-
RND-
RUN-
SAVE
SCREEN
SGN-
SIN-
SOUND
SPACE$-
SPC-
SPRITE ON/OFF/STOP
SPRITE$
SQR-
STEP
STICK
STOP-
STOP ON/OFF/STOP
STR$
STRIG
STRIG ON/OFF/STOP
STRING$-
SWAP
TAB
TAN
THEN-
TIME-
TO
TROFF-
TRON-
USR-
VAL-
VARPTR-
VDP-
VPEEK
VPOKE
WAIT
WIDTH
XOR

The video processor used, has one impediment that has to be carefully considered when planning a sprite display: although 32 sprites may be displayed on the screen at once — only four sprites may be displayed on any one line at any one time. Sprites may only be created in one colour but may be overlapped to give a multi-colour effect — bearing in mind that 'only four-to-a-line' comment above!

There are four sprite modes based upon the number of pixels that form the sprite and their magnification: 8-by-8 unmagnified, 8-by-8 magnified, 16-by-16 unmagnified and 16-by-16 magnified. When 'magnified', every pixel is expanded to form a 4-by-4 pixel block. Using the 8-by-8 format enables 256 sprite patterns to be stored for use. Choice of 16-by-16 sprites reduces this number to 32.

Sprite operation is fast and flicker-free, and very easy to program. It also has full 'wrap-around', so that if a sprite goes off one edge of the screen, it will reappear from the opposite edge and continue moving.

SOUND

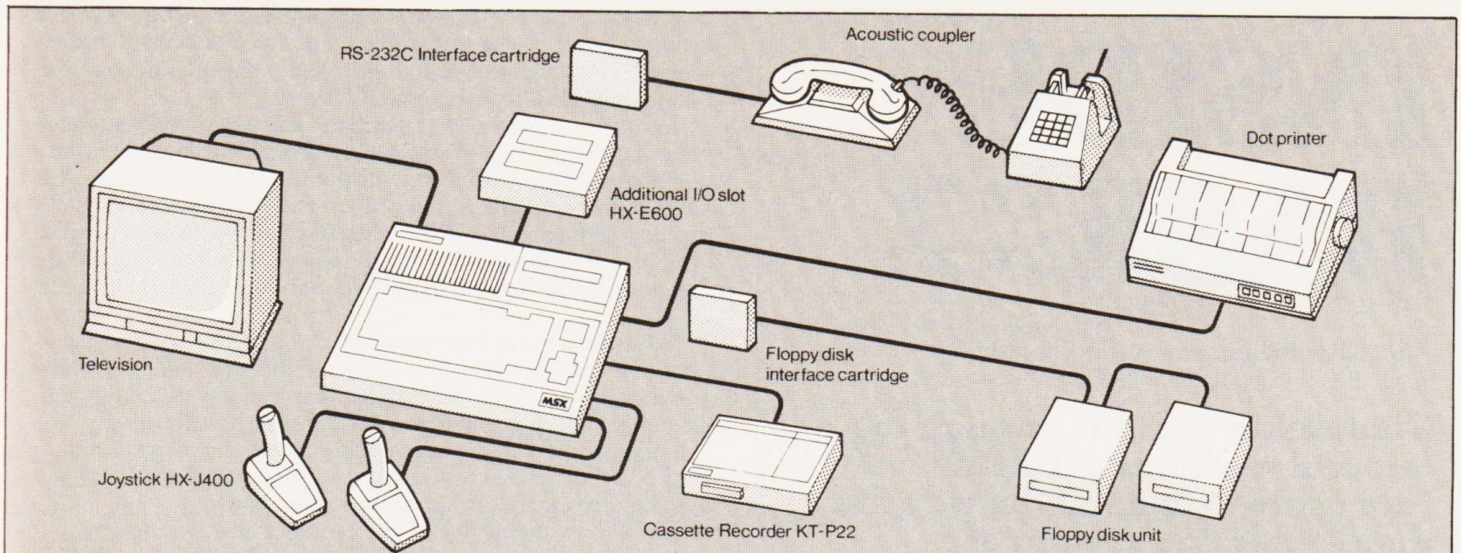
The sound/music facilities of the HX-10 are also very easy to use and are accessed by two principle commands: PLAY and SOUND. Using the PLAY command there are three sound channels which may be independently programmed for volume, pitch, length of note, length of rest, tempo and tonal colour. Pitch may be specified directly as notes A to G with or without flats or sharps, octave or if preferred by a number of 0 to 96 — each integer increment raising the pitch by one half-tone.

Having programmed a simple run, up and down a scale with:

```
PLAY "ABCDEFGFEDCBA"
```

Program operation will continue whilst the music synthesizer circuitry plays your scale.

Using the SOUND command allows the user to create various sound effects (gunshots, aliens, etc.) This is not quite so easy to use as PLAY but is still comparatively simple. Implementing this function you have control over the frequency of each channel,



TOSHIBA HX-10: Expansion options

whether it be a tone or a noise, loudness, envelope period and envelope pattern.

Most of the now expected commands are recognised but MSX BASIC does have several that are useful additions to a programmer's armoury. The HX-10, in common with all other MSX machines recognises three distinct types of numeric variable — integers, single precision and double precision. Integers may be expressed in any of four notations: decimal, hexadecimal, octal, or binary. Single precision returns a number with six digit accuracy, while double precision offers 14 digit accuracy.

A variable type may not only be specified during the program ('A%=' defines an integer, 'A!=' defines single precision) but also a series of variables may be defined at the beginning of your program by:

```
DEFINT A,B,C
DEFSNG L,N-T
DEFDBL V-Z
```

Here 'A', 'B', 'C', 'AC' etc. are defined as integers. 'L', 'N', 'S2', 'QV' etc. as single precision and 'V', 'W3', 'XP' etc. are double precision.

All the usual logical operators are there (=<, =>, etc.) but in addition to AND and OR we also have NOT, XOR, IMP, and EQV, the latter two being 'implications' and 'equivalence' — my Boolean logic is a bit rusty but I daresay you'll find a use for them!

SWAP exchanges the values of two variables, LOCATE is similar to PRINT AT, INPUT$ can either read a specified number of characters from a file or can be used to limit the number of characters INPUT from the keyboard — this is a rather double-edged sword:

```
10 A$ = INPUT$(3)
20 PRINT A$
```

will certainly restrict the length of variable A$ to three characters but it only displays these characters after the third has been entered, and also, as the HX-10 has a keyboard buffer of 10 characters — should you key in four characters, then the fourth is lying in wait to be assigned as the first character ... the next time the program requires an input from the keyboard.

CLOCK AND INTERRUPTS

MSX machines have an internal clock incremented every 1/50th of a second. This may be preset between the values 0 to 65535, on reaching a count of 65535 it will start again from 0. The timer stops whilst data is being input or output to a cassette file. This brings us to a range of useful routines involving use of interrupts. Since the timer counts interrupts at about 1/50th of a second intervals it may be used by:

```
ON INTERVAL = n
GOSUB 1000
```

This defines at what time interval (in this case 'n') the routine at line 1000 will be accessed. Once the routine has been implemented by INTERVAL ON, the program will continue to jump to the specified routine every n/50 seconds until told to stop.

There are several other routines utilising the interrupts — accessing a subroutine regardless of what else the program is doing — providing a certain condition is met. For example:

```
ON KEY GOSUB
```

is used to detect the use of one of the function keys, while:

```
ON STOP GOSUB
```

detects use of the STOP key...in theory the use of this command allows you to make a program 'un-BREAKable' but before running such a program, remember that you may have to switch the machine off to regain control of the keyboard!

ON STRIG GOSUB detects the use of the Space bar on the trigger button on a joystick. Finally ON ERROR GOTO may be used to force a program to continue after detecting an error. Alternatively, it may be used as an aid to debugging a program by printing out various data (variables etc.) at the moment the machine would otherwise have crashed out to the Command level.

Error messages in MSX BASIC are short but useful, as you do not get just an error number shown, but a message such as:

```
DIVISION BY ZERO
IN LINE 300
```

... all pretty clear and concise. One interesting error is number 51 — INTERNAL ERROR — which is an error within BASIC. The manual says this should not occur and recommends temporarily turning off the computer... What then? Phone Microsoft, perhaps?

CONCLUSION

The full list of commands and keywords is shown in Table 2, and in all cases the emphasis is on ease of use. The HX-10 is an easy computer to use but the step from initial use to MSX proficiency is going to require something in addition to the two manuals provided. Much of what a programmer of today needs to know, is not explained in detail. One silly point perhaps, is the stress in the manuals on binary or hexadecimal notation — decimal equivalents may be substituted and would often be more readily understood by most readers.

In conclusion, the HX-10 has to be considered as a match on many points for other computers on the market today. Perhaps at present, a little overpriced, but its ease of use is most welcome. The key response has me a little worried, but I suspect this is more a question of familiarity than an actual fault. Colour control in SCREEN 0 and SCREEN 1 is disappointing but I daresay we'll get round that in one way or another. The basic machine is good but as with every other micro for the home user on the market — none of them is perfect...

INSIDE BASIC: 2

Don Thomasson

Continuing our tour through the internal workings of BASIC, we turn aside to find out the way the main operating system of a computer cooperates with the interpreter, and then resume our travels to explore the way floating-point works.

A computer which could only operate in BASIC would be of limited use, and most worthwhile machines can be made to run machine code programs, which may implement other languages, such as PASCAL. Both the BASIC interpreter and the programs for other languages rely heavily on the 'operating system', which may, in fact, do a lot of the work that BASIC and other languages appear to execute. For example, in the BBC computer the BASIC responds to a SOUND command by setting up a table of parameters and calling on the operating system to carry out the complex procedures which produce the sound. Even the task of setting up an input buffer to receive user instructions is left to the operating system.

In some computers, the operating system and the 'language' program are so intimately mixed that it is difficult to say where one begins and the other ends. The BBC Computer, on the other hand, makes the division quite clear: The two functions are separated into different ROMs. The AMSTRAD CPC464 uses one physical ROM, but the separation is nevertheless maintained. The Spectrum, on the other hand, mixes the two functions in a single ROM.

What does the operating system do? It carries out the routine tasks essential to the running of the computer system as a whole. The display of a character on the screen calls for stored data on the current cursor position, so BASIC merely hands over a character code and leaves the operating system to get on with the job. It may, of course, specify a screen position first, but the operating system has to set up the screen position.

Then there are the colour and graphics functions, which are almost entirely the preserve of the operating system. The BASIC merely hands over the relevant parameters. The operating system draws the lines, plots the points, and organises all aspects of the display.

In the case of the BBC Computer, this process can be taken to the point where the operating system works on instructions supplied by an entirely separate external computer system, and the main computer effectively becomes an 'intelligent terminal', providing display and keyboard functions and perhaps not a lot more.

It was necessary to make the respective functions of the BASIC and the operating system clear, because it would be possible to waste a lot of time trying to find out how some BASIC functions work if the distinction were not explained.

For example, some systems have a number of 'pseudo variables', which look like variable names but in fact call short

BBC Micro: intelligent Terminal

routines that pick up values from the operating system data and pass them into the BASIC system. Some work both ways. The pseudo variable TIME may be used to set up an elapsed time counter, or it may be used to read the counter. In some systems, a pseudo variable can be treated either as a command or as a function. In command form, it instructs that a value should be set to a given value, in function form it reads the value and makes the result available.

This question of values leads on to the way numeric data is stored. Straight binary storage may be used, though only for integers, but for 'real numbers' it is necessary to use floating poing.

FLOATING POINT

In numeric theory, there are an infinite number of numbers between any two given numbers. The floating point system of numeric representation cannot quite match that, but it does allow a reasonably high resolution, making a distinction between numbers which differ by a very small proportion of their absolute values.

A floating point number is stored in two parts, an exponent and a mantissa. The mantissa is set up in 'fractional binary', which means that the most significant digit has a value of 0.5, the next has a value of 0.25, the next a value of 0.125, and so on. The most significant digit of a normalised floating point mantissa is always true, so the mantissa as a whole has a value between 0.5 and (nearly) 1.

The exponent defines a power of 2, and this is multiplied by the mantissa to give the overall value. For example, a floating point number with an exponent of 4 and a mantissa of value 0.6 would have an overall value of $2^{4}*0.6 = 16*0.6 = 9.6$.

In practice, the stored exponent is usually increased by 128, so that the number stored can range from 1 to 255, representing −127 to +127 as the real exponent value.

Another little tweak depends on the fact that the most significant digit of the mantissa is always 1. This allows the digit to be replaced by a sign bit, 0 for positive, 1 for negative.

If you have digested that, it is time to look more closely at the capabilities of floating point numbers. The exponent can have a value ranging from 2^{127} to 2^{-127}, and this will be multiplied by a value between ½ and 1. The largest number that can be represented is therefore about $1.7 * 10^{38}$, the smallest around $5.9 * 10^{-39}$. This range should be sufficient to satisfy anyone in their right minds, though not, perhaps, some of the madder mathematicians.

The mantissa determines the resolution, the smallest difference that can be shown between two numbers. And here there has been a point of slight embarrassment. The recognised standards were 'single precision', using a 24-bit mantissa, and 'double precision', using a 56-bit mantissa. Now 24 bits allow a resolution of one part in 16,777,216, reduced by rounding to about half that value. This would show a difference of a penny in £83,886, which was not really enough for big business. On the other hand, a 56-bit mantissa was able to resolve about one part in $3.6 * 10^{16}$, which was rather an overkill.

Purists are asked, at this point, to accept that the simplifications used above are in the interests of clarity!

The embarrassment has been resolved in recent computers by using 32-bit mantissas, which give a resolution of around one part in $2 * 10^{9}$, adequate but not excessive. Unfortunately, manufacturers seem reluctant to state which floating point standard they use, though some do give the resolution, from which the standard can be calculated.

FLOATING POINT CALCULATIONS

Multiplying two FP numbers together involves adding the exponents and multiplying the mantissas. This may leave the mantissa of the result in a 'denormalised' state, since ½ * ½ = ¼, which is outside the correct ½ − 1 range. If so, the mantissa is doubled and the exponent decremented until the range is correct. This is called 'normalisation'.

A similar process is used for division, but addition and subtraction are more difficult. First, the smaller number is adjusted by repeated increment of the exponent and halving of the mantissa, until the two exponents are equal. The addition or subtraction is then performed, and the result must be normalised to bring the mantissa back into the correct range.

These processes involve multiplication of the mantissa by ½ or 2, which is achieved in practice by a left or right shift. Shifting a 32-bit number is not a simple matter. If a Z80 processor is in use, it is possible to put the number into two register pairs and perform the shift without reference to store, but with other processors the task is more complicated. As a result, it is common to find a number of specialised subroutines which can be called to help the normalising and denormalising processes, and tracing out the relevant calls can be quite difficult.

We have dealt with the four most common functions, but we must now look at the more complicated trigonometric and logarithmic functions.

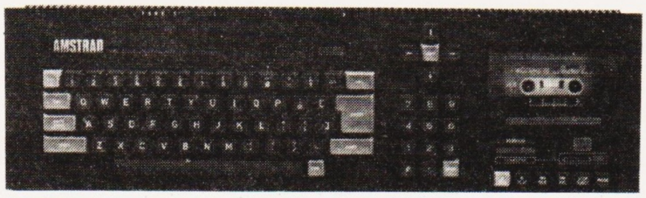
POLYNOMIALS

When a computer is asked to calculate trigonometric or logarithmic functions, it uses an appropriate power series, of the form;

$$Y = A + BX + CX^{2} + DX^{3} + \ldots$$

The values A, B, C etc are read from a table, and the whole process is implemented by a loop.

In some cases, it is necessary to bring X within a certain range for which the polynomial is valid. Angles, for instance, are usually reduced to the range of 0 – 90°, the required correction for the actual quadrant being made separately.



where a very fast machine may show a balancing disadvantage through loss of extreme precision, but the errors involved are usually small. Measurement of a real angle is rarely very precise, and the accuracy of trigonometric functions is only important where small differences are concerned.

It should be noted that the form of the polynomial can vary, some computers dividing and subtracting instead of multiplying and adding.

Some people have been slightly mystified by advice to use, say, X * X * X rather than X↑3 when they want to calculate the cube of X. The reason is that the first form carries out an actual multiplication, whereas the second calculates the log of X,

The number of terms used in polynomials varies a good deal. In general, the number of terms is directly related to the accuracy of the result, and also to the speed of execution. This is one area multiplies the result by three, and then takes the antilog. This is a much slower process, and is usually a trifle less accurate. It has been known for a computer to report the result as 26.9999 ...

MATHEMATIC ABILITY

When someone turns from BASIC to assembler code, it is often a severe shock to discover that mathematical calculations are limited, in simple terms, to integer working. Providing a complete set of routines to handle floating point is no simple task, and it soon becomes evident that the BASIC interpreter gives a lot more assistance to the user than is evident on the surface.

There have, of course, been integer BASICs, which have no facilities for handling floating point numbers, but they have been few and far between. A more rational scheme is to provide for integer calculation as well as floating point, since integers occupy less space and can be worked on in less time. However, the actual advantage gained is often unnoticeable in practice.

And that brings us to the question of the speed at which a BASIC interpreter will work.

SPEED v PERFORMANCE

When someone asks why a particular BASIC implementation is so fast, it is best to turn the question round and ask why other implementations are slower. In early versions, it was necessary to scan through a table of variable values, looking for a matching variable name. In more modern systems the variable name is given explicitly, together with the address at which its value can be found. Again, a GOTO used to involve searching through a program for a particular line number. One recent system allows both this approach and a format which gives the store address at which the required line is located.

As an example of recent thinking, let us have a look at a stored program line in the AMSTRAD CPC464.

```
240 PRINT # (Z*8), TAB(6+3*N);HEX$(B,2);
```

This line will be split up into its meaningful segments:

LINE BYTE	MEANING
29 00	The line occupies &29 = 41 bytes.
F0 00	The line number is 00F0 = 240
BF	The 'token' for PRINT
23	Hash character, indicating stream selection.
28	Open brackets
0D 05 00 D8	The '&0D' warns that a real number definition follows. The value will be found at an address formed by adding 0005 to the base address of the variables area. The variable name is Z (Code &58, plus &80, = &D8)
F6	the token for '*', multiply.
16	A value of 8, given by subtracting &0E from &16.
29	Close brackets.
2C	Comma
EA	Token for TAB
28	Open brackets
14	A value of 6
F4	The token for '+', add.
11	A value of 3
F6	The token for '*', multiply.
02 2A 00 4E	The '&02' warns that an integer definition follows. The value will be found at an address formed by adding 002A to the base address of the variables area. The variable name is N. (Code &4E, no addition of &80 here.)
CC	Integer calculation
29	Close brackets.
3B	Semicolon delimiter
FF 73	The FF warns that a function follows. 73 is the code for HEX$.
28	Open brackets
0D 30 00 C2	A real number, displaced 0030 from the base of the variables area variable name B. (&42 + &80 = &C2)
2C	Comma
10	A value of 2
29	Close brackets
3B	Semicolon delimiter.
00	Line terminator.

The original line, including the line number and a following space, occupied 40 characters, so the economy of storage is reasonable, while all the data required for execution or listing is directly accessible.

The same system uses the form 1E XX XX to indicate line XXXX, or 1D XX XX to indicate the line held at address XXXX. Both are used within a single ON GOTO statement!

As might be expected, the CPC464 is fast. It still has to preserve the right priority for calculations, resolving brackets first, then multiplications, and so on, but it has evaded the more serious time penalty associated with searching processes.

STRING HANDLING

The approach to string handling varies a good deal from one BASIC to another, but the general principle involves the use of a string variable area, in which strings or parts of strings or assemblies of strings are stored. The location of each string variable is identified within the main variables area, so that a given string can be found quickly.

The fundamental material from which the strings are defined is contained within the stored program, and is always present, but it is often necessary to construct derived strings on the basis of that fundamental material, and that is where the fun begins. Some programs fill up the string area quite rapidly, and a point may be reached where the area is full. The dreaded 'garbage collection' process must then be applied. This involves scanning through the stored strings and eliminating any that are merely copies of strings held elsewhere. This arises because a string is set up in the variable area even if it is going to be put in an arrary or otherwise stored again. If you call a statement such as A$ = B$ + C$, the two latter strings will be set up, their combined form will be set up, and A$ will then be set from the combination. Only the original sources of B$ and C$ and the combination in A$ are of lasting interest. The intermediate forms are 'garbage', and can be thrown away.

INTERPRETER DESIGN

Some of the more significant differences between one dialect of BASIC and another have been mentioned in passing, but it would be a mammoth task to list them all. Every new machine seems to have a new version of BASIC, and there is some justification for a sigh of relief over the promise of standardisation offered by the MSX concept. But a standard is only acceptable if it is a good standard. It will take a little while to determine whether MSX BASIC satisfies this requirement.

It might be contended that there is room for some variation in BASIC dialects, since there is a corresponding variation in user preferences. One user will insist that procedures are essential, another will dismiss them as a gimmick, and prefer other facilities. To some, the ability to omit spaces in a program will seem an advantage, while others will see this as leading only to unreadable programs.

Anyone who sets out to design a BASIC interpreter has an unenviable task. He will have to make compromises, take difficult decisions, and be prepared for at least as much criticism as praise. His product may show incredibly good benchmark results, yet be derided for its other characteristics. It is unlikely that it will be greeted with universal approval.

The dialects of BASIC which have been mentioned are all associated with 'eight-bit' processors. Hopes have been expressed that significant improvements may be possible where 16-bit processors are used, but this has yet to be demonstrated. Mathematical co-processors might simplify the working of floating point calculations, with greatly improved execution speed, while store size limitation could, in theory, disappear completely. So far, it does not seem that a really gold-plated BASIC has appeared, perhaps because those involved with 16-bit processors are not too interested in providing one, believing that their customers will have their eyes on higher-level languages.

Those who hope and expect to see BASIC fly away and disappear have long begun to suspect that it will do nothing of the sort. Some forms of BASIC can be compiled into machine code, which removes the stigma of snail-pace execution, but the resulting code tends to be large and unwieldy. Much of the interpreter code needs to be copied into the compiled program, and appropriate sections of it are called from the main code stream, so machine code stretching out to several thousand bytes is not an uncommon result. Fortunately, it is rarely necessary to do much 'debugging' in the object code, providing that the original program has been thoroughly tested in the first place.

For many purposes, the extreme convenience of BASIC is an overriding factor, and those who desecrate it will have to come up with far more convincing reasons if they are to do it any real harm.

MACRO1

Farouk Elhiddiny

Apple owners in possession of a Z80 card may find this article of interest. The assembler — supplied with the standard diskette may not recognise the MACRO directive, but the program described here allows the Z80 programmer to overcome this problem.

Owners of the Apple II, II+ or IIe computers can use CP/M application programs if they have access to a Z80 card. The diskettes supplied with the card contain a standard, two-pass, Z80 or 8080 assembler that does not usually recognize the MACRO directive.

The program described in this article is an MBASIC-80, single-pass assembler that will replace MACRO statements in an assembly language program with the appropriate routines on disc. The prospective user will thus need:

- **CP/M** with at least one drive.
- **MBASIC** installed or loaded.
- **A macro** library customized to the user's needs.
- **A main** routine calling macros from the library.
- **The assembler** described in this article saved as "MAC.BAS".

Although this program was written and tested on an APPLE II+ computer with a language card and a Z80 card, it is of such a general nature as to be usable with any system operating with CP/M and M/BASIC. Also — if such circumstances are present — it is independent of any particular processor code, Z80, 8080, or even 6502.

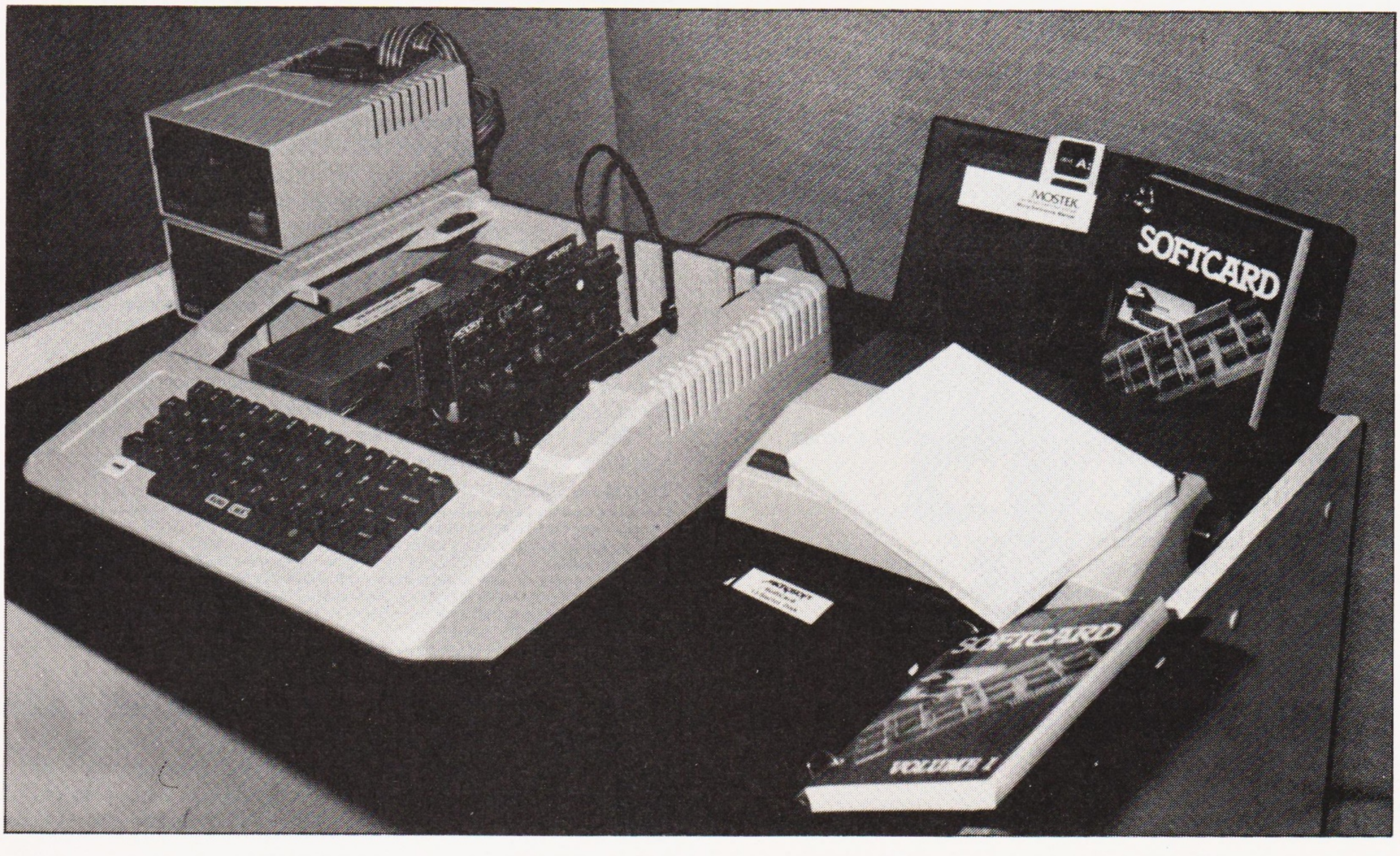
This assembler will strip the main routine and macros of all trailing remarks ie from those statements ending with ";" followed by some remark. Statements beginning with a single semi-colon, ";", will be kept but statements beginning with a double semi-colon ";;" will be ignored.

LABELS

Labels within a macro can be given local names by the LOCAL directive. The assembler will replace these labels with new identifiers beginning with the string "PP" followed by a different number for each label.

A macro can have up to four formal parameters. The calling statement may provide real parameters to replace the formal parameters. If a real parameter is to be intentionally omitted one of the marks & or # must be put in its place. (&) will cause the corresponding formal parameter to be replaced by the null string (""). (#) will cause the corresponding formal parameter to be replaced by "NULL" which must be given the value "FALSE" in the main routine. Thus a test condition <IF NOT PARAMETER> in the macro will be replaced by <IF NOT NULL> to handloe the case of parameter omission.

The routine offered to the assembler must be given a name according to the standard rules of CP/M, but the extension maust be <.MAC>. The routine produced by the assembler will have the same name but of an extension <.ASM>. Thus the



generated routine will be ready for immediate treatment by the resident assembler.

A small set of rules must be observed.

- **A macro** in the library must start with a line of the form:
  NAME MACRO FPAR1 FPAR2 ....
  Where Name is the name chosen for the macro in the library.
  FPAR1 FPAR2 etc. are up to four and no more formal parameters.

- **The same** macro when called from within the main routine or another macro must be called with a calling line of the form :
  MACRO NAME RPAR1 RPAR2 ....
  Where Name is the name chosen for the macro in the library.
  RPAR1 RPAR2 etc. are up to four and no more formal parameters.
- **Both** the calling and starting lines must not contain any remarks.

- **Every** macro in the macro library must end with the statement :
  ENDM
  without any labels or tailing remarks.

- **LOCAL** is considered on opcide in this assembler, and can be used anywhere within a macro to define and generate local lablels. More than one local statement can be used in a macro provided the labels of one system are kept apart from those of another.

- **A LOCAL** statement must be of the form :
  LOCAL L1 L2 ....
  Where L1 L2 etc. are up to four and more formal labels that will be replaced by labels generated by the assembler.

TABLE 1

Functions of the variables and arrays

A$	Common name for the main and output file.
B	General use for calculations.
B$	Input and output statements.
BO$	Individual bytes from B$.
C	General use for calculations.
C$	Initials of B$. Input statement from the macro library.
I	Loop variable.
L	Length of B$.
L1	Length of local labels and parameters.
MS	Message indicator.
M$	Message.
N	Highest achieved level of imbedded macros.
L(N)	Number of local labels of level N.
L$(I,N)	Ith local label level N.
P(N)	Number of generated labels of level N.
P$(I,N)	Ith generated label of level N.
Q(N)	Number of real parameters of level N.
Q$(I,N)	Ith formal parameter of level N.
R(N)	Number of real parameters of level N.
R$(I,N)	Ith real parameter of level N.
U$	Name of macro library.
Z	Number of distinct words in a statement.
Z$(I)	Ith word in a statement.

- **The** LOCAL statement must not begin with any label and must not end with any remark.

- **Parameters** and formal labels must be separated with spaces. No other delimiter is allowed. Any real parameter can be intentionally omitted. In its place one of the marks & or $ must be used. (&) will cause the corresponding formal parameter to be replaced by the null string (""). (#) will cause the corresponding formal parameter to be replaced by (NULL). The macros in the library must be provided with instructions indicating whether one or both or none of these substitutions is allowable.

This assembler will recognize only the opcodes < MACRO ENDM LOCAL >, therefore it cannot check for any standard syntax error. But it has its own small set of errors :

- Use of LOCAL or ENDM in the main routine.
- A called macro is not found in the library.
- A macro in the library is not properly terminated with ENDM.
- More than seven levels of imbedded macros.

A macro library can be of any convenient name. More than one macro library can exist on the same diskette but only one can be used in one program.

THE PROGRAM

Listing 1 shows our macro assembler. To use this system, enter Listing 1 and save it with the command :

SAVE "MAC"

It will be given the file name

<MAC.BAS>.

Call by the command.

MBASIC MAC/F:9

because it will need file blocks to handle up to seven imbedded macros.

The assembler will call for the name of the main file which can be entered with or without extension. The extension if present will be deleted. MAC will be added to form the main file name, ASM will be added to form the output file name.

The assembler will next ask for the library file name. If there are more than oen library only one can be used. MBASIC will search for the main file and library. If either is not found a corresponding message will be displayed and the program aborted. If a file is found with the name of the output file it will be erased and a new empty one created.

The assembler then starts its work. It will take time; BASIC is notoriously slow, besides it will make frequent calls to the diskette fetching macros. At least OK will be displayed and the user will have an output file with the extension ASM ready for treatment by the resident assembler.

Table 1 shows all variables and arrays used and their functions in the program. Customization is possible by changing some dimensions.

```
5 REM DEFINE ARRAYS AND CONSTANTS
10 OPTION BASE 1:ON ERROR GOTO 2070
20 DIM R$(6,7),Q$(6,7),L$(4,7),P$(4,7),Z$(6),Q(7),L(7),P(7),R(7)
30 TAB$=CHR$(9):SPC$=CHR$(32):CR$=CHR$(13)
35 REM DEFINE MAIN ROUTINE AND MACRO LIBRARY
40 INPUT"FILE NAME";A$
50 INPUT"LIBRARY NAME";U$
70 B=INSTR(A$,"."):IF B=0 THEN 90
80 A$=LEFT$(A$,B-1)
90 N=1
95 REM OPEN MAIN AND OUTPUT FILES
100 OPEN"I",#1,A$+".MAC"
105 REM READ MAIN FILE
110 OPEN"O",#9,A$+".ASM"
200 IF EOF(1) THEN CLOSE:END
210 LINE INPUT#1,B$:GOSUB 1000
215 REM CHECK INITIALS
```

```
 220 IF C$=";;" THEN 200
 230 IF C$=";"  THEN 270
 240 IF C$="MACRO" THEN 290
 250 IF C$="ENDM" THEN 280
 260 IF C$="LOCAL"THEN 280
 265 REM SAVE OUTPUT STATEMENT AS IS
 270 PRINT#9,B$:GOTO 200
 276 REM SEND ERROR MESSAGE
 280 MS=1:GOTO 2000
 290 GOSUB 1120:GOSUB 1200:GOSUB 1100:GOSUB 1350
 295 REM READ MACRO FILE
 300 IF EOF(N) THEN MS=4:GOTO 2000
 310 LINE INPUT#N,B$
 320 GOSUB 1000
 325 REM CHECK INITIALS
 330 IF C$=";;" THEN 300
 340 IF C$=";"  THEN 640
 350 IF C$="ENDM" THEN 670
 360 IF C$="LOCAL" THEN 650
 395 REM STRIP OFF TAILING REMARKS
 400 B=INSTR(B$,";"):IF B<>0 THEN B$=LEFT$(B$,B-1)
 410 IF L(N)=0 THEN 520
 415 REM SUBSTITUTE FOR LOCAL LABELS
 420 L=LEN(B$)
 430 FOR I=1 TO L(N)
 440 L1=LEN(L$(I,N)):B=INSTR(B$,L$(I,N)):IF B=0 THEN 470
 450 B1$=LEFT$(B$,B-1):B2$=RIGHT$(B$,L+1-B-L1)
 460 B$=B1$+P$(I,N)+B2$:L=LEN(B$)
 470 NEXT I
 520 L=LEN(B$):IF Q(N)<3 THEN 640
 525 REM SUBSTITUTE REAL FOR FORMAL PARAMETERS
 530 FOR I=3 TO Q(N)
 540 L1=LEN(Q$(I,N))
 560 B=INSTR(B$,Q$(I,N)):IF B=0 THEN 610
 570 B1$=LEFT$(B$,B-1)
 580 B2$=RIGHT$(B$,L+1-L1-B)
 600 B$=B1$+R$(I,N-1)+B2$:L=LEN(B$)
 610 NEXT I:IF C$="MACRO" THEN 290
 635 REM SAVE MODIFIED OUTPUT STATEMENT
 640 PRINT#9,B$:GOTO 300
 650 GOSUB 1120
 660 GOSUB 1400:GOTO 300
 670 CLOSE#N:N=N-1
 680 IF N=1 THEN 200
 690 GOTO 300
 995 REM SUBROUTINE TO EXTRACT INITIALS
1000 L=LEN(B$):C$=""
1010 FOR I=1 TO L:CO$=MID$(B$,I,1)
1020 IF CO$=SPC$ AND LEN(C$)=0 THEN 1080
1030 IF CO$=TAB$ AND LEN(C$)=0 THEN 1080
1040 IF C$=";" AND CO$=";" THEN I=L:GOTO 1070
1050 IF C$=";" AND CO$<>";" THEN I=L:GOTO 1080
1060 IF CO$=SPC$ OR CO$=TAB$ THEN I=L:GOTO 1080
1070 C$=C$+CO$
1080 NEXT I
1090 RETURN
1095 REM SUBROUTINE TO EXTRACT WORDS OF A STATEMENT
1100 IF Z=0 THEN 1120
1110 GOSUB 1300
1120 L=LEN(B$):Z=1:Z$(Z)=""
1130 FOR J=1 TO L:BO$=MID$(B$,J,1)
1140 IF BO$=SPC$ OR BO$=TAB$ THEN Z=Z+1:Z$(Z)="":GOTO 1170
1150 IF BO$=CR$ THEN J=L:GOTO 1170
1160 Z$(Z)=Z$(Z)+BO$
1170 NEXT J
1180 RETURN
1195 REM SUBROUTINE TO FETCH A MACRO FROM THE MACRO LIBRARY
1200 N=N+1:IF N>7 THEN MS=2:GOTO 2000
1210 OPEN"I" ,#N,U$
1220 IF EOF(N) THEN MS=3:GOTO 2000
1230 LINE INPUT#N,C$:L=LEN(C$):L1=LEN(Z$(2))
1240 B=INSTR(C$,Z$(1)):C=INSTR(C$,Z$(2)):IF B*C=0 THEN 1220
1250 IF B<C THEN 1220
1260 B$=C$
1270 RETURN
1295 REM SUBROUTINE TO FIX REAL PARAMETERS
1300 FOR I=1 TO Z
1310 IF Z$(I)="#" THEN R$(I,N-1)="NULL":GOTO 1340
1320 IF Z$(I)="&" THEN R$(I,N-1)="":GOTO 1340
1330 R$(I,N-1)=Z$(I)
1340 NEXT I:R(N-1)=Z:RETURN
1345 REM SUBROUTINE TO FIX FORMAL PARAMETERS
1350 FOR I=1 TO Z
1360 Q$(I,N)=Z$(I)
1370 NEXT I
1380 Q(N)=Z
1390 RETURN
1395 REM SUBROUTINE TO GENERATE LABELS FROM LOCAL LABELS
1400 FOR I=1 TO Z-2:NN=NN+1
1410 L$(I,N)=Z$(I+2)
1420 P$(I,N)="PP"+RIGHT$(STR$(NN),LEN(STR$(NN))-1)
1430 NEXT I:P(N)=Z-2 :L(N)=Z-2
1440 RETURN
1995 REM SEND ERROR MESSAGE
2000 ON MS GOTO 2020,2030,2040,2050
2010 PRINT M$:CLOSE:END
2020 M$=C$+" ILLEGAL IN MAIN ROUTINE":GOTO 2010
2030 M$="MORE THAN SEVEN MACRO LEVELS":GOTO 2010
2040 M$=Z$(2)+ " MACRO NOT IN LIBRARY":GOTO 2010
2050 M$=Z$(2)+" MACRO WITHOUT PROPER END":GOTO 2010
2070 CLOSE:ON ERROR GOTO 0
```

NEXT ISSUE

In the next issue of Computing Today, we demonstrate the use of the macro assembler with the aid of a 'cooked-up' macro library and some assembly lanuage routines which contain the MACRO directive. So, until then . . .

ORIC AND SINCLAIR COMPUTERS
Oric 1 computer 48K £85 (£82) £92. Oric Atmos computer 48K £171 (£158) £168. Oric colour printer £134 (£123) £140. Sinclair flat screen TV £113 (£105) £115. Sinclair Spectrum Plus Computer £182 (£176) £187. Sinclair QL Computer £406 (£385) £410. Sinclair Spectrum 48K £131 (£131) £143. Microdrive £51 (£50) £60. RS232 interface 1 £51 (£50) £60. Special offer:- Microdrive + Interface 1 + 4 cartridges £102 (£100) £120. Blank microdrive cartridges £5.50 (£6) £7. Standard floppy disc interface for Spectrum £102 (£92) £112. (See Cumana disc section for suitable disc drives). Fuller FDS keyboard for spectrum £52 (£52) £62. Fuller master unit £56 (£56) £62. Interface 2 £20.45 (£20) £24. 32K memory upgrade kit for 16K spectrum (issue 2 and 3 only) £31 (£28) £30. Spectrum Centronics printer interface £51 (£47) £52. ZX printer has been replaced by the Alphacom 32 £71 (£69) £82. 5 printer rolls (State whether Sinclair or Alphacom) £13 (£16) £21. ZX81 computer £45 (£44) £54. 16K ram packs for ZX81 £28 (£25) £30.

COMMODORE COMPUTERS
Commodore C16 Starter Pack £145 (£142) £162. Commodore Plus/4 £305 (£281) £301. Commodore 64 £222 (£215) £235. Convertor to allow most ordinary mono cassette recorders to be used with the Vic 20 and the Commodore 64 £9.78 (£9) £11. Bargain package:- cassette convertor + compatible cassette recorder £37 (£38) £44. Commodore cassette recorder £43 (£44) £50. Printer interfaces for Vic20 and the Commodore 64:- Centronics £45 (/41) £46, RS232 £45 (£41) £46. Disc drive £233 (£209) £234. 1520 printer/plotter £165 (£149) £159. MPS801 Printer £235 (£220) £245. Light pen £29 (£29) £33.

ACORN COMPUTERS
Electron £173 (£179) £199. BBC Model B £404 (£357) £387. Kenda double density disk interface system £149 (£131) £141. See below for suitable disc drives.

CUMANA DISC DRIVES
To suit disc interfaces of Sinclair spectrum, BBC B and Videogenie. Single:- 40 track single sided £176 (£158) £178, 40 tr double sided £218 (£195) £215, 80tr ss £207 (£186) £206. 80tr ds £234 (£209) £229. Dual:- 40tr ss £299 (£280) £320, 40tr ds £395 (£353) £393, 80tr ss £372 (£334) £374, 80tr ds £437 (£390) £430.

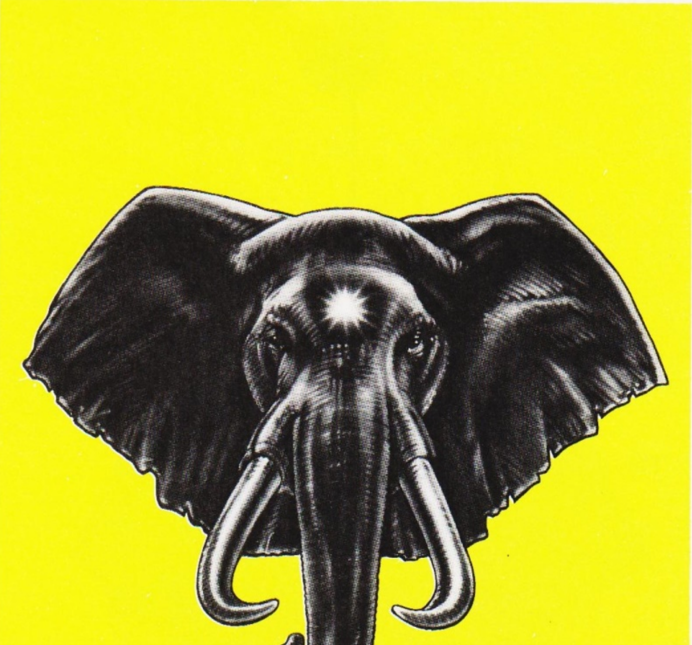
PRINTERS
Oki Microline 80 £138 (£135) £165. Brother HR5 £162 (£146) £170. Shinwa CTI CPA80 £237 (£228) £258. Cannon PW1080A £382 (£344) £374. Epson RX80 £277 (£251) £282. Epson RX80F/T £314 (£286) £316. Epson FX80 £399 (£358) £388. Combined matrix printers and electric typewriters:- Brother EP22 £173 (£166) £186, Brother EP44 £258 (£235) £260. MCP40 Oric Colour printer/plotter £134 (£123) £140. Interfaces to run the above printers from Vic and the Commodore 64 £45 (£41) £46. We can supply interfaces to run the above printers from Sharp computers £58 (£52) £55.

UK101, SUPERBOARD AND VIDEOGENIE
We still support these Computers. Write for our list.

COMPUTER REPAIRS
We offer a world-wide repair service. Write for a quotation.

SWANLEY ELECTRONICS
Dept CT, 32 Goldsel Road, Swanley, Kent BR8 8EZ, England.
TEL: Swanley (0322) 64851

Official orders welcome. All prices are inclusive. UK prices are shown first and include post and VAT. The second price in brackets is for export customers in Europe and includes insured air mail postage. The third price is for export customers outside Europe (include Australia etc) and includes insured airmail postage.



ALGORITHM ANGLES

Bill Horne

For many aspiring programmers, the real difficulties arise in the creation of suitable algorithms, the 'working rules', rather than in setting up code to execute the algorithms. Algorithm Angles will offer some useful hints, explaining the way in which algorithms can be worked out, while also shedding light on some familiar system functions. We begin with the subject of trigonometry.

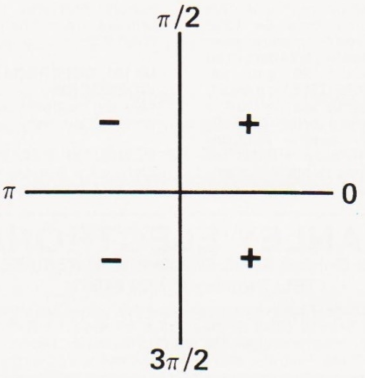


Figure 1. The sign of COS in the four quadrants.

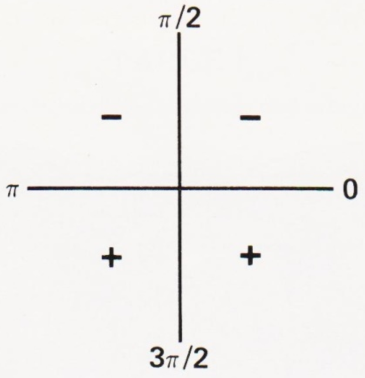


Figure 2. The sign of SIN in the four quadrants.

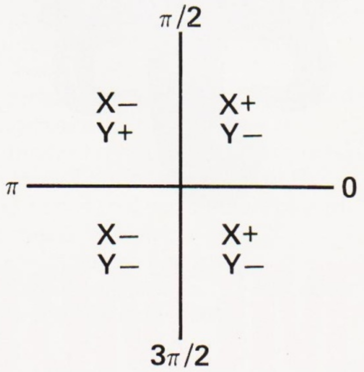


Figure 3. Signs of X and Y in ATN(X/Y)

Most program languages provide some trigonometric functions, though the repertoire varies a good deal from one implementation to another. Older languages tend to work only in radians, leaving the user to make any necessary conversions by use of an appropriate multiplying factor. More recently, provision has been made for both radians and degrees, and even the lesser-known grads, which are scaled at 100 to the circle. For simplicity, only radians will be considered here.

The normal approach to trigonometric calculations is based on the use of a power series. This is commonly evaluated by an iterative routine that can work with different sets of coefficients to calculate different functions. Written in BASIC, the routine might look like this:

```
100 T = 0
110 READ A
120 FOR B = 1 TO A
130 READ C
140 T = T + C
150 T = T * X
160 NEXT
```

READ A picks up the number of coefficients, while READ C picks up the individual coefficients. X is the independent variable, and the result appears in T when the loop drops out. If A = 3, the successive values of T are:

$$X*C3$$
$$X*(X*C3 + C2) = X*C2 + X^2*C3$$
$$X*(X*C2 + X^2*C3 + C1) = X*C1 + X^2*C2 + X^3*C3$$

A C0 constant can be added separately, if required.

In some systems subtraction and division are used instead of addition and multiplication, but the principle is the same.

For trigonometric functions, it is often convenient to make X the square of the independent variable. For example:

$$COS(X) = 1 - X^2/2! + X^4/4! - X^6/6! + \ldots$$

This series is derived on a strict mathematical basis, and would give perfect accuracy if an infinite number of terms were used, but for practical purposes we need to bend the rules a little. Suppose we want to limit the actual series to four terms, while approaching the accuracy obtainable with eight terms. We can proceed thus:

$$\begin{aligned} COS(X) = {} & 1 - X^2/2! + X^4/4! - X^6/6! \\ & + X^8/8! - X^{10}/10! + X^{12}/12! - X^{14}/14! \\ & = (1 + X^8/8!) - (X^2/2!)*(1 + X^8*2!/10!) \\ & + (X^4/4!)*(1 + X^8*4!/12!) \\ & - (X^6/6!)*(1 + X^8*6!/14!) \end{aligned}$$

If X is limited in range to $0 - \pi/2$, the terms containing X^8 will vary very little. The first term will vary from 1 to 1.000919, for example, and the variation of the others will be less. Taking the mean values of these terms, and modifying the coefficients accordingly:

1.0 becomes 1.0004595
0.5 becomes 0.500005
0.041666 becomes 0.041667
0.0013888 becomes 0.0013889

These relatively small changes may seem trivial, suggesting that the simple four-term series would suffice, but you can check that by using the short BASIC listing given above. The result of the changes does not give complete accuracy, but this could be improved by taking another four terms of the original series into account. The object here is to indicate how the serial coefficients can be modified to give better accuracy without having to use too many terms.

To deal with angles outside the 0 to $\pi/2$ range, it is necessary to use a reduction process. First, $2^* \pi$ is subtracted repeatedly until the result is less than $2^* \pi$. If the residual angle is greater than π, it is subtracted from $2^* \pi$. If that gives a value greater than $\pi/2$, the value is subtracted from π, and in this case the power series result is negated (See Fig 1).

You may like to try setting up a similar series for SIN(X), based on:

$$SIN(X) = X - X^3/3! + X^5/5! - X^7/7! + \ldots$$

Note that in this case the rules for angle reduction are slightly different, as indicated in Fig. 2. The sign of the power series result is reversed when the angle lies between π and $2^* \pi$.

For those who want to go hyperbolic, the series for SINH and COSH are as those for SIN and COS, but with the minus signs replaced by plus signs. Other series which may be of interest are;

$$e = 1 + 1 + 1/2 + 1/3 + 1/4 + \ldots$$
$$e^X = 1 + X + X^2/2 + X^3/3 + X^4/4 + \ldots$$
$$\log_e X = (X-1) - (X-1)^2/2 + (X-1)^3 - \ldots$$

The last of these is valid only for X = 0 to 2, and a reduction process is usually needed before applying it.

Other series expressions may be found in mathematical books.

ADAPTATIONS

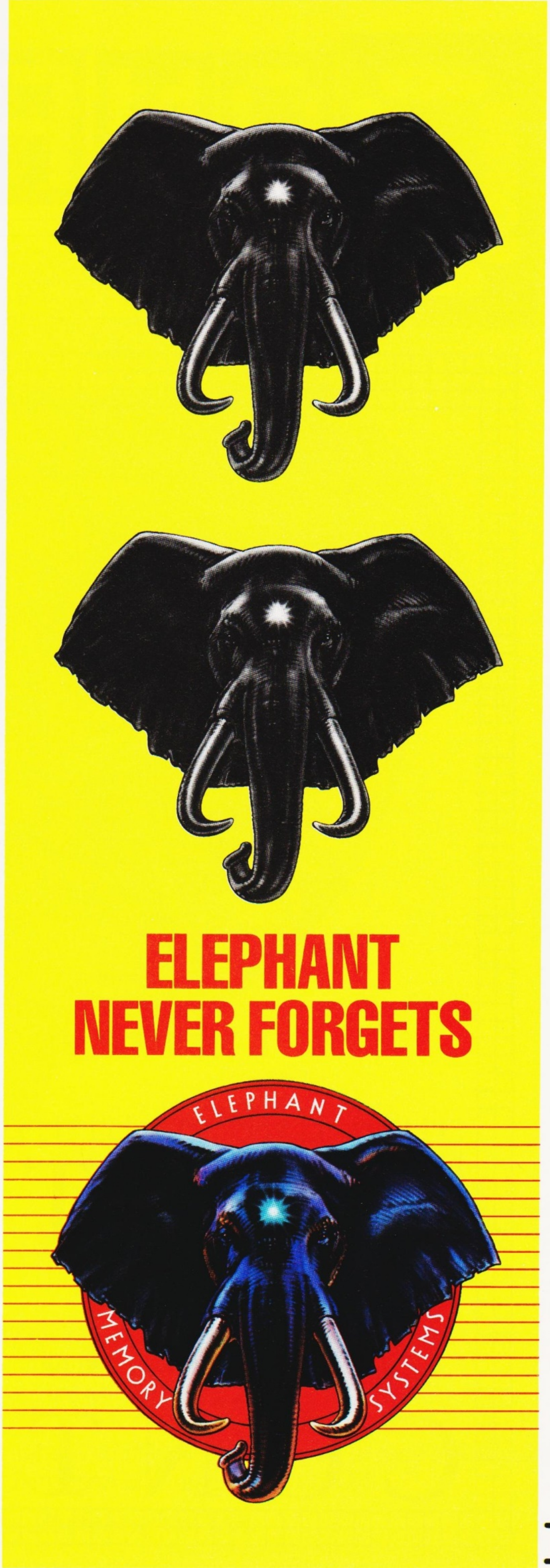
Some users of small computers have been known to express annoyance when they discover that the functions ASIN (Arcsin) and ACOS (Arccos) are not implemented on their machine, only ATN (Arctan) being provided. By doing so, they are rather giving themselves away, since it is not too difficult to work out that:

$$ASIN(X) = ATN(X/SQR(1-X^2))$$
$$ACOS(X) = (\pi/2) - ATN(X\ SQR(1-X^2))$$

However, ATN itself must be used with caution. The usual format for limited systems is ATN(Z), but Z is in fact the ratio of two values, X (opposite side) and Y (adjacent side). The ratio Z alone is not enough to define a unique value of ATN(Z). (See Fig. 3). Both X and Y, or Z and Y must be taken into account. Some implementations only give angles between $\pi/2$ and $-\pi/2$. (e.g. the AMSTRAD CPC464). In that case, π needs to be added if Y is negative, the most convenient form to achieve this being:

$$ANGLE = ATN(X/Y) - \pi *(Y<0)$$

This assumes that (Y<0) will return −1 if Y is negative, otherwise 0, which is true of many systems.



COMPUTING Today

In recognition of our new editor, and to celebrate ***Computing Today***'s seventh year on the newsstands, we are making some general improvements to the magazine. To begin with we have the new logo, which we hope will be to your liking (be careful not to mistake it for another magazine!). Also, the April issue will be substantially bigger than it has been for some while — and it will be better, with more features, ***two*** new series, and a host of other items that push ***Computing Today*** even further into the unexplored territories of the computer world:

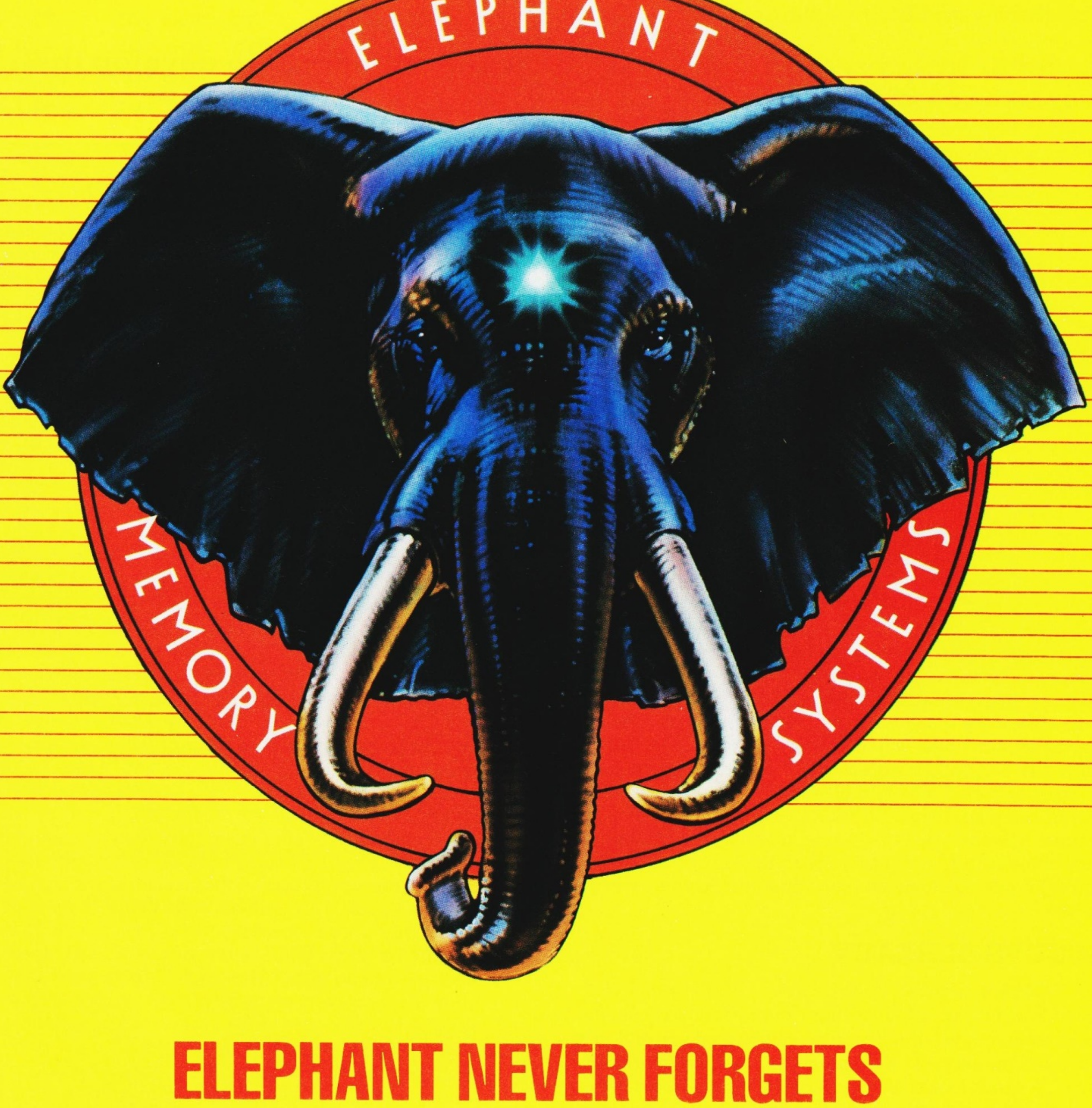
- The Lessons of History — a new series which chronicles the development of the computer, from the publication of Turing's monumental paper "On computable numbers with an application to the Entschiedungsproblem", and continues through until the present day.
- Introducing C — our second new series which examines the concepts behind, and use of this remarkable language.
- Bubble memories — despite astonishing figures for reliability and speed, the bubble memory remains ignored by the major manufacturers. We discuss the theory behind these devices, and consider the reasons for them being technologically 'left out in the cold'.
- Writing a spelling checker — our first FRAMEWORK feature designed to offer ideas and guidelines that allow **you** to perform the final coding of the program.

All this and more in the next edition of Computing Today.

A NEW LOOK...A NEW DIRECTION

THE EVOLUTION CONTINUES..

REMEMBER



ELEPHANT NEVER FORGETS

Get the best from your computer with ELEPHANT disks. Certified 100% error-free and problem-free, and with quality maintained for at least 12 million passes, ELEPHANT disks are guaranteed to meet or exceed every industry standard and are compatible with virtually every computer on the market.

Look for the ELEPHANT sign at your local Dealers – or in case of difficulty, phone or write direct to Dennison Manufacturing Co. Ltd.

Dennison

Dennison Manufacturing Co. Ltd.
Colonial Way, Watford, Herts WD2 4JY, Tel: Watford (0923) 41244, Telex: 923321
France: Soroclass, 8, Rue Montgolfier - 93115, Rosny-Sous-Bois, Tel: 16 (1) 855-73-70
Germany: Marcom Computerzubehöer GmbH, Podbielskistr. 321, 3000 Hannover 1, Telex: 923818
Other Countries: Dennison International Company, 4006 Erkrath 1, Matthias-Claudius-Strasse 9, Telex: 858 6600

AMSTRAD LETTER WRITER

Phil Harvey and Richard Sargent

A clever little program for Amstrad owners who want to develop their own word-processing package.

If you want to use your computer to write the occasional letter to The Council complaining that your dustbins haven't been emptied, you hardly need a wordprocessor. Some typewriters cost less than most wordprocessors, be they on ROM, tape or disc — but you probably know that, or you wouldn't be bothering to read this article.

A BASIC WORDPRO

Here, we describe a small BASIC wordprocessor for the Amstrad — it does its job adequately (ie: letter-writing) and you can always change it around if it doesn't do all things you'd quite like it to, like checking your spellings and making the tea.

There are a number of points about the Amstrad CPC464 which make it possible to present such a small wordprocessing package. First of all, there is the CPC464 keyboard. It **is** a good keyboard, laid out to correct typists' standards, and since it is a pleasure to use, why not use it more often? Secondly, the Amstrad has the built-in tape recorder which is not only reliable, it would seem, but also has that essential item of equipment: the mechanical counter. Essential, that is, if you want to build up a library of documents on your C90 cassettes.

Finally, there is the 80-column screen, so you can see in advance **exactly** how your document will look, and the cheap Amstrad printer which will churn out your deathless prose.

If you're still not convinced of the wisdom of proceedeing, perhaps you ought to be told (if you don't already know) that the Amstrad has a keyboard BUFFER in software which means that no matter how fast you type, you won't be able to outstrip the ability of the BASIC program to keep up with you. Now that's a piece of knowledge to make a few Spectrum owners weep, at any rate.

HICCUPS ALREADY

Our CPC464 was linked up to an Epson RX80 printer and seemed to want to print all listings double-spaced — ie: the printer was receiving two line-feeds even though it was still in its factory-set condition of "NO automatic line-feed after carriage-return". A quick look at the Amstrad manual revealed the problem: all "unused" wires on the Amstrad printer-cable are grounded, but it happens that Epson use PIN 14 as an extra-linefeed control switch. Tying it to ground tells the Epson to add a LF to every CR it receives!

We got around the listing problem, as you can see. There is no need to cut the pin 14 wire on the printer cable because the print-routine of the wordprocessor only issues a single CR at the end of a printed line, so it needs the LF which the printer-cable adds. I daresay the listings will work properly too, on an Amstrad printer!

FEATURES

The idea behind "Letter Writer" is to keep everything as simple as possible so that the code can be examined, understood and modified. The entire text of the document typed is held in the array T$. It's a two-dimensional array, with the cells of the array representing the characters of the imaginary piece of paper you are typing on. Thus in line 140 the T$ array is dimensioned according to how many characters you would like down the paper (LINES). The screen (and thus the paper) is configured to the typist's standard of a 64 character line bordered by margins 8 characters wide.

The wordprocessor runs mainly in "overprint" mode. The cursor is a solid square and may be moved to any part of the document, or over the blank screen, by the four arrow keys. DELete iks a destructive backspace — that is to say it overwrites the text with spaces as it moves backwards through the document. Text may be entered anywhere except within the margins (which are indicated on the screen) and will overwrite whatever information is already there. The Amstrad repeating-key facility is retained. Overwriting, deleting and cursor movement are all as fast as they need to be, and there is no indication that BASIC routines are involved. Such is the speed of Locomotive BASIC.

However, these functions alone would not make for efficient letter-writing. An INSERT mode is needed, and so too is a backspace which closes up the text after the offending characters have been removed. CTRL X moves all text between the cursor and the end of the line to the right, thus creating space for new text. But please note: if the line was a full one, the shift-right process causes characters to leave the end of the line and then, regrettably, they are lost. CTRL Z removes the character beneath the cursor and then moves the text occupying the rest of the line one position to the left. Again, the consequences of this action do not extend beyond the line in question, so what we have in Letter-Writer is a program that can format a line, but not a whole document, and it is therefore more like a "correcting-typewriter" than a true word processor.

CTRL X and CTRL Z don't work with much speed, since they work by shifting data in the COLUMN dimension of the T$ array. They are, however, perfectly alright for corrections to the odd word or two. Because of the keyboard buffer their action is somewhat fascinating to watch. Tapping CTRL X rapidly stores a lot of "shift right" instructions in the Amstrad's keyboard memory and for the next 10 seconds or so you can watch a section of your sentence chug eastwards as the space you want is opened up in your line.

CTRL V will centre a line of text, and is useful for headings, especially letter headings.

The other commands are CTRL S for "save document", CTRL L for "load document", CTRL P for "print document" and CTRL Q for "quit letter-writer". The save and load work upon the entire T$ array and can take some time if you have dimensioned a large array and are using the slow speed 1000 baud transfer. Printing, on the other hand, can be controlled in a simpole manner by the user. All characters will be sent to the printer, from the start of the document — T$(1 ,1) — up to the reverse-slash. The reverse-slash may be inserted anywhere in the text, but if it is absent, all T$ will be printed.

Figure 1.

```
CHR$(12)   ^L Load
CHR$(16)   ^P Print
CHR$(17)   ^Q Quit
CHR$(19)   ^S Save
CHR$(22)   ^V Centre
CHR$(24)   ^X Move text right
CHR$(26)   ^Z Move text left
CHR$(127)  delete key
CHR$(143)  cursor symbol
CHR$(240)  cursor up
CHR$(241)  cursor dowm
CHR$(242)  cursor left
CHR$(243)  cursor right
```

Figure 2.

```
CD0BBC  CALL SCR GET LOCATION
015000  LD BC,80
09      ADD HL,BC
C305BC  JP SCR SET OFFSET
-------------------------------
C305BC  CALL SCR GET LOCATION
01B0FF  LD BC,-80
09      ADD HL,BC
C305BC  JP SCR SET OFFSET
-------------------------------
```

THE LISTING

In the listing "$" has printed as "£", but otherwise everything is as it should be. The program is made up of simple modules which manipulate T$ in the appropriate way. The screen is constantly updated by the short routine starting at line 3800, which works by repeatedly calling the routine at 3600 w3hich updates one single line. You will observe that, for reasons of eccentricity, the text in T$ is passed to a one-character-size WINDOW on the screen. This seems to work just as fast as the more traditional PRINT@ or LOCATE x,y so no complaints, please! The translation of the CHR$() codes are shown in figure 1 and the source listing of the short machine code section is shown in figure 2. Machine code? We hear you ask. . . It's there to help the Letter-Writer scroll its screen if you are working on documents more than 25 lines in length. All the machine does is call a routine in ROM which causes a hardware scroll of the Amstrad screen. However, when using machine code within the Amstrad, beware of the cassette buffer! This is a delightful little beast which meddles with the high-memory pointer when you aren't looking. It is not documented in the manual that accompanies the computer. At switch on PRINT HIMEM returns 43903. After a cassette read or write PRINT HIMEM returns 39807 — a large cassette buffer has been formed by lowering HIMEM. If HIMEM is set at the beginning of a program, as would be the case with other computers, then the first time use of the cassette system will grab 4K of memory below HIMEM, and reset HIMEM to a new, lower, level. This can play havoc with your vision of the memory map and may well scramble machine code that you've placed in high-memory. The method around this problem, suggested in an appendix in Amsoft's £20 BASIC SPECIFICATIONS book is shown in line 9700. A dummy cassette operation sets HIMEM once and for all and the buffer so created will be the one used by subsequent cassette operations until a system RESET wipes it away. Therefore, after RESET, always start the program with RUN 9700. After a BREAK, the program may be started with an ordinary RUN.

Letter-Writer is written so that it can be modified. Try, for example, setting right margin and left margin (RM and LM) to zero. You then have the full 80 column screen at your disposal, and if you keep your lines down to 64 characters in length (average), you will have 16 spaces at the end of each line to cater for possible inserts. The cursor symbol can be altered, as can the codes for the various letter writing functions. There is also scope for greater versatility in the print-document routine, and scope for faster cassette routines. So, re-code if you wish, but most of all, happy letter-writing!

The listing for Amstrad Letter-Writer.

```
10 REM **********************************
20 REM *                                *
30 REM * WORDPROCESSOR - (C) P.Harvey *
40 REM *                                *
50 REM **********************************
60 REM V2
70 REM  RM = Right margin
80 REM  LM =  Left margin
90 REM
100 REM +- Initialisation and set up -+
120 WIDE=80:ROWS=25:CLS:LM=8
121 RM=8:LINES=25
122 MODE 2:COLUMNS=WIDE-RM:IF LINES<25 THEN LINES=25
140 DIM T$(COLUMNS,LINES):CX=LM+1:CY=1:BL=0:BELL=6
150 GOSUB 3900:GOSUB 3800
160 ON BREAK GOSUB 9800
180 GOSUB 4300
2999 REM +- Text entry routine -+
3000 REM
3020 IF T$(CX,CY+BL)<" " THEN FOR X=LM+1 TO COLUMNS:T$(X,CY+BL)=" ":
     WINDOW X,X,CY,CY:PRINT " ";:NEXT X
3040 WINDOW CX,CX,CY,CY:PRINT CHR$(143);:I$=INKEY$:IF I$="" THEN 3040
3060 PRINT T$(CX,CY+BL);
3080 IF I$<" " OR I$>CHR$(126) THEN 3200
3100 T$(CX,CY+BL)=I$:WINDOW CX,CX,CY,CY:PRINT I$;
3120 CX=CX+1:IF CX>COLUMNS THEN CX=LM+1:GOSUB 3400:IF FG=0 THEN CX=COLUMNS
3130 IF CX=COLUMNS-BELL THEN SOUND 1,40
```

```
3382 IF I$=CHR$(17) THEN 9800
3383 IF I$=CHR$(24) THEN GOSUB 4500:GOTO 3000
3384 IF I$=CHR$(26) THEN GOSUB 4600:GOTO 3000
3390 GOTO 3000
3399 REM +- Move down one text line -+
3400 IF CY<ROWS THEN CY=CY+1:FG=1:RETURN
3420 IF CY+BL=LINES THEN FG=0:RETURN
3440 BL=BL+1:CALL ADDR:GOSUB 3600
3490 FG=1:RETURN
3499 REM +- Move up one text line -+
3500 IF CY>1 THEN CY=CY-1:FG=1:RETURN
3520 IF BL=0 THEN FG=0:RETURN
3540 BL=BL-1:CALL ADDR+10:GOSUB 3600
3590 FG=1:RETURN
3599 REM +- Update current text line -+
3600 IF LM>0 THEN FOR X=1 TO LM:WINDOW X,X,CY,CY:PRINT " ";:NEXT X:PRINT "¦";
3610 FOR X=LM+1 TO COLUMNS:WINDOW X,X,CY,CY:PRINT T$(X,CY+BL);:NEXT X
3620 IF RM>0 THEN FOR X=COLUMNS+1 TO COLUMNS+RM:WINDOW X,X,CY,CY:PRINT " ";:
     NEXT X:WINDOW COLUMNS+1,COLUMNS+1,CY,CY:PRINT "¦";
3630 RETURN
3799 REM +- Update entire screen -+
3800 GOSUB 4300:Y1=CY:WINDOW 1,COLUMNS+RM,1,25:CLS:CY=1
3810 GOSUB 3600:CY=CY+1:IF CY>25 THEN CY=Y1:RETURN
3820 GOTO 3810
3899 REM +- Clear entire text file -+
3900 FOR Y=1 TO LINES:T$(0,Y)=CHR$(174)
3910 FOR X=LM+1 TO COLUMNS:T$(X,Y)=" ":NEXT X,Y
3920 WINDOW 1,WIDE,1,25:CLS:RETURN
3999 REM +- Centre text on line -+
4000 FG=0:FOR X=COLUMNS TO LM+1 STEP -1:IF FG=0 AND T$(X,CY+BL)=" " THEN 4045
4020 IF FG=0 THEN X1=COLUMNS:FG=1
4040 T$(X1,CY+BL)=T$(X,CY+BL):X1=X1-1
4045 NEXT X:IF FG=0 THEN 4090:FOR X=X1 TO LM+1 STEP -1:T$(X,CY+BL)=" ":NEXT X:
     IF FG=0 THEN 4090
4049 FOR X=X1 TO LM+1 STEP -1:T$(X,CY+BL)=" ":NEXT X
4050 FG=0:FOR X=LM+1 TO COLUMNS:IF T$(X,CY+BL)=" " THEN 4080
4060 IF FG=0 THEN SP=(INT ((X-(LM+1))/2))+LM:X1=X:FG=1
4080 NEXT X:IF FG=0 THEN 4090
4086 FOR X=LM+1 TO COLUMNS:IF SP>=X THEN T$(X,CY+BL)=" ":GOTO 4089
4087 T$(X,CY+BL)=T$(X1,CY+BL):X1=X1+1:IF X1>COLUMNS THEN SP=COLUMNS+1
4089 NEXT X:GOSUB 3600
4090 GOSUB 3400:CX=LM+1:RETURN
4099 REM *- Save text file to tape -*
4100 WINDOW 1,WIDE,1,25:CLS:INPUT "Enter file name:";N$
4110 OPENOUT N$
4120 FOR Y=1 TO LINES:FOR X=LM+1 TO COLUMNS
4122 PRINT £9,T$(X,Y):NEXT X
4130 NEXT Y:CLOSEOUT:GOSUB 3800:RETURN
4199 REM +- Load text file from tape -*
4200 WINDOW 1,WIDE,1,25:CLS:INPUT "Enter file name:";N$
4210 OPENIN N$
4220 FOR Y=1 TO LINES:FOR X=LM+1 TO COLUMNS:INPUT £9,D$
4223 IF D$<" " THEN D$=" "
4225 T$(X,Y)=D$:NEXT X
4230 NEXT Y:CLOSEIN:GOSUB 3800:RETURN
4299 REM +- Set up machine code routines -+
4300 RESTORE 4310:FOR M=39783 TO 39783+19:READ D:POKE M,D:NEXT M:RETURN
4310 DATA &CD,&0B,&BC,1,&50,0,9,&C3,5,&BC
4312 DATA &CD,&0B,&BC,1,&B0,&FF,9,&C3,5,&BC
4399 REM +- Send file to printer -+
4400 FLAG=0:FOR Y=1 TO LINES:FOR X=1 TO COLUMNS:IF X<=LM THEN PRINT £8," ";:
     GOTO 4415
4410 E$=T$(X,Y):IF E$="\" THEN X=COLUMNS:FLAG=1:GOTO 4415
4412 PRINT £8,E$;
4415 NEXT X
4416 IF FLAG=1 THEN Y=LINES:GOTO 4422
4420 PRINT £8,CHR$(13);
4422 NEXT Y:FLAG=0:PRINT £8,CHR$(13);:RETURN
4499 REM +- Move text right of cursor RIGHT -+
4500 IF CX=COLUMNS THEN 4520
4510 FOR X=COLUMNS TO CX STEP -1:T$(X,CY)=T$(X-1,CY):NEXT X
4520 T$(CX,CY)=" ":GOSUB 3600:RETURN
4599 REM +- Move text left of cursor LEFT -+
4600 IF CX=COLUMNS THEN 4620
4610 FOR X=CX TO COLUMNS-1:T$(X,CY)=T$(X+1,CY):NEXT X
4620 T$(COLUMNS,CY)=" ":GOSUB 3600:RETURN
9699 REM +-Create space for m/c routine -+
9700 OPENOUT "DUMMY":ADDR=HIMEM-25:CLOSEOUT
9708 MEMORY ADDR:ADDR=ADDR+1:GOTO 120
9799 REM +- 'ON BREAK' routine -+
9800 WINDOW 1,WIDE,1,25:STOP
9900 WIDTH 80:PRINT £8,CHR$(&1B);"A";CHR$(6);CHR$(&1B);"E";CHR$(&1B);"G":LIST £8
```

DATA SECURITY

Bill Horne

Computer data security is a tricky problem that emerges with saddening regularity. But is there such a beast as Absolute Security?

When the media are short of fresh news, they can always renew the subject of computer data security. It is an emotive subject, because it is made to appear as a threat to the cloak of secrecy which many of us like to visualise as concealing our personal affairs. As databanks multiply, more and more facts about us are stored away, and even people with completely clear consciences may feel uneasy at the thought of those facts being available for covert examination.

Yet many facts were filed away for reference in card indexes and the like before the computer came into existence. What has changed is that computers have allowed the files to grow in size, and have allowed the files to be examined remotely, without an actual visit to the place where the files are stored. The rate of access has also been speeded up, and massive printouts can be generated with little human assistance.

To consider the significance of these changes, we need to take as a base the situation which exists where computers are not used; the old-fashioned filing cabinet system ruling the roost, as it still does in many instances.

OLD STYLE

Properly organised, a single four-drawer filing cabinet can hold a very large amount of information, but sifting through that information can be a slow process. If the contents of the cabinet are considered worth protecting, it will normally be locked outside working hours, and perhaps opened only when access to the information is required. In any case, there will usually be someone around to notice whether any stranger is looking at the contents of the cabinet.

The ability to open the cabinet is usually shared between a limited number of people, those who hold a key or know the necessary combination. But keys can be lost or copied, and combinations can be discovered if someone is careless. With the cabinet open, the modern copier allows quite large amounts of data to be extracted and taken away without the loss being noticeable. Gone is the traditional image of a spy holding a miniature camera above a document in the light of a desk lamp. Undercover work has moved with the times.

There is usually a house rule banning the copying of classified documentation, but enforcement is difficult, especially as copiers are often banished to obscure corners because the activity around them is deemed to be disturbing to those at nearby desks. However, a spy would be wise to limit his copying sessions, since excessive use of the machine might arouse interest, particularly among those waiting their turn. This limits the amount of material which can be handled in this way.

The main safeguard against such copying is the integrity and sense of responsibility of those who are permitted access to the information. If they are careless with their keys, or write the combination on the wall because they are afraid of forgetting it, unauthorised access, either during the working day or at night, becomes reasonably simple.

Where highly classified documents are concerned, a meticulous checking system is used to make sure that nothing is missing, but that cannot detect whether unauthorised copies have been made.

Such a filing system is therefore inherently vulnerable. A certain amount of information is needed by anyone seeking covert access. He must know where to find what he wants, he must be able to gain access to the filing cabinets and he must know how to open the cabinets. To obtain and use this information, he must involve himself in some degree of illegal action, the most serious crime being to be found out.

MASS DATA STORAGE

Databanks vary enormously in size, from lists which occupy part of a floppy disc to tabulations filling banks of replaceable hard discs. With the larger sizes, only part of the complete files may be available at a given time. Television licence records are only of immediate interest during the month in which they fall due, unless a tabulation of addresses in a given area is required. It would be possible to separate the main file into months, and collate the area list by mounting the sub-files in turn. Bank account data, on the other hand, is in constant use, and must be continuously accessible.

This distinction is important, because it adds to the amount of information needed to access given files. A would-be intruder needs quite a lot of information, in any case, unless he is a regular user of the data bank.

The regular user has the advantage that he knows the procedures, he knows when particular information will be available, and his presence at a workstation will arouse no comment. Working alone, however, he is at a disadvantage. If he walks out with a bulging briefcase, he may attract unwelcome interest. A floppy disc could be concealed in the folds of a newspaper, but compatible equipment is needed to reveal its contents.

The 'inside man' is better placed if he has an 'outside' confederate to whom he can pass information by landline. That is probably the preferred arrangement. It solves the problem of acquiring the necessary know-how, and if used discreetly is almost undetectable.

So the main safeguard once again depends on the integrity of those who have regular access. Once a 'mole' is established, the doors to data theft are wide open, providing that access to the outside world is available, as it usually is with large databanks.

CLOSED SYSTEMS

Leaving the question of outside access alone for the moment, let us look at closed systems, which have no external data links. These may be quite small micro-based arrangements, with all their data contained in a few floppies. The main danger here is a casual attitude on the part of the users. Cases are known where quite important data in floppy disc form is left lying around unprotected. Because it is not directly readable, it is not considered to be a security risk.

An amusing example of this attitude dates back to the days of paper tape. A tidy-blitz was in progress, in the course of which it was decreed that all program listings not in current use should be removed to a high-security safe. The punched tapes from which the listings were produced, however, were kept in a nest of small drawers which were wide open for anyone to investigate. When the security people realised this, there were several heart-attacks. It emerged when an important listing was created from one of the tapes, and the original listing was known to be locked away.

That incident could well have been the starting point for a subsequent about-turn in security practices where program data was concerned. It reached a point where even a 'noddy' special test program had to be given a classification and either locked away or destroyed after use. That slowed test work down considerably . . .

SECURE ROOMS

With any closed system, it is possible to put the hardware and files in a secure area, to which access is limited, and this is very wise if the system handles sensitive material of any kind. However, it is a useless precaution if it is not implemented properly. Cases are known where some terminals are placed outside the secure area, on the grounds that they are not to be used for work on classified data. Even if measures are taken to ensure that this limitation is effective, there is a risk that some clever blighter will find his way round them.

An extension to the secure room concept was a requirement that certain installations should be enclosed in a Faraday Screen, so that no electromagnetic radiation could escape to be detected and recorded outside. This involved enormous expense, especially as adequate air conditioning was essential, and other complex systems were required. The requirement may have originated from an incident in Moscow, where it was found that a hole had been bored into a duct carrying teleprinter cables. A probe was seen projecting through the hole, and it was caught in a pair of mole grips (how appropriate), sawn off, and examined. It was an electro-magnetic sensor, and the guess was that it had been used in an attempt to intercept teleprinter traffic, though there must have been easier ways to accomplish that.

The possibility of detecting anything sensible in the radiations of a computer must be minute, and after a while the requirement was eased considerably, the move being interpreted as the first stage in a tactical retreat to common sense.

This sort of precaution seems to stem from security paranoia, a disease which can strike in all sorts of unexpected places. Where secret information is concerned, some precautions are necessary, otherwise secrecy has no meaning, but a point can be reached where the measures taken are too extreme to be useful.

SOFTWARE SECURITY

An example arises with the security of program material, whether it relates to military matters or games. It is clearly desirable to hide the content of some programs from prying eyes, but it can be contended that absolute security is impossible to achieve. Indeed, some of the less effective methods have proved to provide no more than an interesting challenge to those who like solving puzzles.

However it is stored and loaded, a program has a specific 'stored image'. With some machines, the operating system may be used to ensure that only keyboard actions relevant to the program have any effect, but operating systems can be changed. It is not an easy process, but problem-solvers don't like things to be too easy. With 'soft' systems the difficulties are must less significant.

Once the blocks to dumping the stored image have been overcome, interpretation of the program and the removal of any protection features requires no more than time and patience. With specialised systems there may be more difficulty in discovering the machine code, but that does not arise with any standard microprocessor.

So it can be said that breaking into a protected program can be made more difficult, but it is extremely doubtful whether it can be made totally impossible. Like some security precautions in other fields, it can discourage rather than provide an absolute bar. It may equally well add to the ranks of those who see all kinds of security measures as mountains to be climbed.

Perhaps the most absurd attitude towards program security arises when a manufacturer objects to the revelation of any part of his product's operating system. There are probably hundreds of people whose first reaction to a new system is to study the firmware, which they can do quite simply. If there are nasties concealed in the coding, that will soon become common knowledge, on the grape-vine if not more publicly through the specialist media.

But we stray a little from the main theme. Let us return to the subject of data theft.

REMOTE TERMINALS

It was in the mid nineteen-sixties that the concept of remote terminals really got under way. Originally using standard teleprinters, they advanced to video terminals in comparatively recent times, but the underlying concept throughout has involved the use of telephone lines which are part of the national network. If you know the right telephone number, you can get in touch with most computers that use outside lines. You will then need to know something about the operating procedures, and probably a 'password' which will identify you as an authorised user.

All this information is difficult to obtain, but it can be done. There was an odd incident some years back in which a remote terminal user called up his usual computer, and was told that he was already logged on. Since the work involved was classified, a rare old panic ensued. His password and identity code were hastily changed, but the real worry was that there seemed to be no way of identifying the intruder or discovering anything about him. The authorised user, by the way, was an abstruse mathematician who had no interest whatever in politics or international intrigue, and who was meticulous where security was concerned, so the fault was unlikely to lie with him.

Fortunately, there is a procedure which can minimise this sort of intrusion. It has been known for many years, but does not appear to be widely used. The routine involved is that the user calls up the computer to request service, and then breaks the connection. The computer calls him back, so an attempt to intrude can only succeed if the proper remote terminal position is used. Like many effective concepts, the idea is simple. The only snag is the possibility of a wrong number. There is a delightful story about a remote auxiliary power station which was designed to ring up headquarters automatically if anything went wrong. Unfortunately, it became connected to an answering machine, and they went on talking to each other for hours . . .

CONCLUSION

From the points which have been discussed, it must be clear that data security, whether in filing cabinets or on discs, depends in the long run on the integrity and sense of responsibility of the people who use the equipment. Ultimately, they are the weak link. There may be others who can contribute. Those who install, commission and test a system must know a good deal about it, and those who maintain it have continued access which they might misuse. All have a duty to avoid spreading key information too widely.

In theory, 'vetting' procedures ensure that only those who are trustworthy are allowed to have access to the key information, but recent trials have suggested that vetting is not entirely effective. Obvious undesirables may be weeded out, but telepathy would be needed to do a completely thorough job of screening. Those who engage in undercover work are unlikely to make their sympathies obvious in advance. Moreover, there is the 'social conscience' syndrome, which impels inherently trustworthy people to 'leak' secrets because they believe it is in the best interests of the nation as a whole.

All that, however, is no more applicable to computing than to any other form of data storage. The difference in the computer case lies in the speed with which information can be stolen, and the sheer amount of information which can be involved.

The incident of the remote terminal service found to be already in use led to security work being transferred to close systems, but many data banks cannot afford to do that, because their main use is external access. While that is obtained directly through the national telephone network there must be some risk of leakage, unless the 'ring back' system is adopted — and even that may not be completely fireproof.

It must be concluded that there is, in reality, no such thing as absolute secrecy. Perhaps if we accepted that we would save a lot of pointless expenditure of energy, trying to conceal the truth about ourselves and what we are doing.

That, however, is most unlikely to happen.

ACT MICRODEALER

xi APRICOT

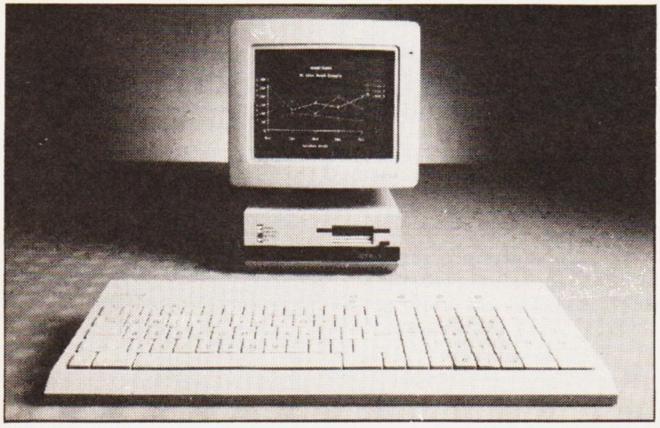
CPU	8086
MEMORY	256K RAM
LANGUAGES	Microsoft BASIC, Personal BASIC
MASS STORAGE	No cassette drive Integral Sony 3½" 315K microfloppy disk drive Integral 5 or 10 Mb hard disk
OS	MS-DOS 2.11 with GSX bundled CP/M-86 (not yet available) Concurrent CP/M-86 (not yet available)
KEYBOARD	QWERTY, cursor, numeric 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.

APRICOT F1

CPU	8086
MEMORY	256K RAM
LANGUAGES	MS-DOS, Concurrent DOS (Optional)
MASS STORAGE	No cassette drive One integral 3½" 720K Sony microfloppy disk drive
OS	MS-DOS 2.11, Concurrent DOS (optional)
KEYBOARD	QWERTY, cursor, numeric pad
INTERFACES	Infra-red link for keyboard or mouse, expansion slot, RS-232C, Centronics
DISPLAY	TV or optional monitor
GRAPHICS	80 by 24 text 640 by 256 four-colour, 320 by 256 16-colour maximum high resolution
SOUND	No

Notes. The Apricot F1 is designed as a low-cost entry-level machine for small businesses (a cheaper cut-down version, the F1 e, is for schools and colleges). It includes several bundled applications including SuperCalc, SuperWriter and SuperPlanner. An optional five-slot expansion bus may be added: also a 10 Mb Winchester unit. There is an optional infra-red mouse/trackball. RAM is expandable to 768K.



HAMPSHIRE

KENT

LONDON

MANCHESTER

MIDDLESEX

SCOTLAND

SURREY

SUSSEX

WALES

WEST MIDLANDS

CBM MICRODEALER

COMMODORE 64 COMMODORE 715B

Notes: The Commodore 64 is a popular micro with a great deal of games software available. There is also some business software, such as spreadsheets and word processors, available, but this suffers from the lack of an 80-column screen. Graphics and sound have extensive capabilities, for example eight multi-colour sprites and three channels of sound covering nine octaves each.

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

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

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.

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:
JASON INSKIP
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.

AMSTRAD extension cable extends video and power head giving comfortable viewing distance. Only £9.95 (including p&p). Amware, 43 Milford Court, Gale Moor Avenue, Alverstoke, Hants PO12 2TN.

TO FILL THIS SPACE, PHONE JASON ON 01 - 437 0699

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.

BOOKS

HALF PRICE Computer Books. Hardware, Software. Special Offer. SAE P. T. Kidd, No. 4 Pumptree Mews, Macclesfield SK11 8UP.

FLOPPY DISKS

AAA

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

DON'T LEAVE IT TO CHANCE! GIVE YOUR BUSINESS A BOOST BY ADVERTISING IN

COMPUTING TODAY

PHONE 01-437 0699 FOR DETAILS

COMPONENTS

68000 SYSTEM KITS

A range of high quality, double sided, 8 by 8 PCB's which form the core of a 68000 system.

CPU contains 68000 L8 micro with a mHz crystal clock, dynamic refresh sequencer, bus buffering etc. PCB £80. Component set £110.

B10 provides 8/16 K Eprom, Ram, 20 I/O lines, RS232 interface, counter-timer, interrupt handling etc. PCB £80. Component set £98.

RAM 256K dynamic. Page/address selectable. PCB £85. Component set £245.

A boot loader/monitor is available in EPROMS at £30.

Add VAT to above prices. Delivery 14 to 28 days.

P. L. Watson, 101 Village Road, Bromham, Bedford MK43 8HU.
Tel: 02302 2867

LIBRARIES

LOOK! Any program for any micro at 20% discount. Also the only tape exchange club specialising in today's Top 20 (Spectrum, Commodore). Membership free. Write stating micro, to Ham Software Library (PCT), Ham Lane, Elstead, Surrey GU8 6HQ. We've got the lot.

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 . . .

BERKSHIRE

Micro General **PRINTER SPECIALISTS**

Call for advice on printer selection.
Epson & Microline stockists. Interfaces & cables available for all popular micros.
6 THE BIRCHWOODS, TILEHURST, READING.
TEL: 0734 25226

CHESHIRE

Computer Junk Shop

We Buy, Sell, Break Computers & Peripherals.
10 Waterloo Rd, Widnes, Halton. Tel: 051 420 4590.

CORNWALL/DEVON

A. B. & C. COMPUTERS (CT)
Duchy House, 6 Lower Aylmer Sq., St. Austell.
Tel: 0726 64463/67337

Wide range of popular Micros, Printers, books and accessories. We stock all U need — try us first for service and competitive prices

HERTFORDSHIRE

NEWBRAIN & SANYO
HARDWARE & SOFTWARE
Printers, Epson, KDC, Juki etc. Monitors, Tape Recorders, Books, Expansions, CP/M. **Sanyo 550/555 Computers.** Access/Ma·l Order. Ask for details.
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

LONDON

A. J. Duchesne
(Computer Consultants) Limited
Specialists in Small Business Computing
10-12 Creechurch Lane, London EC3A 5AY
Telephone: 01-621 0433

SOUTH LONDON

CROYDON COMPUTER CENTRE
CCC
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.

LOOKING FOR MICROCOMPUTER HARDWARE OR SOFTWARE, THEN LOOK NO FURTHER THAN COMPUTAMART

CLASSIFIED ORDER FORM
COMPUTAMART

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 £17.50!

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

TELEX 86736 SOTEX G

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

Send 24p A4 sized SAE for full list.

TRADE ENQUIRIES: ANIROG SOFTWARE LTD. 29 WEST HILL DARTFORD KENT (0322) 92513/8
MAIL ORDER: 8 HIGH STREET HORLEY SURREY 24 HOUR CREDIT CARD SALES HORLEY (02934) 6083
PAYMENT BY CHEQUE P O ACCESS/VISA 50p POSTAGE & PACKAGING