

1/4

# PERSONAL COMPUTER WORLD

50p U.S.\$2.00

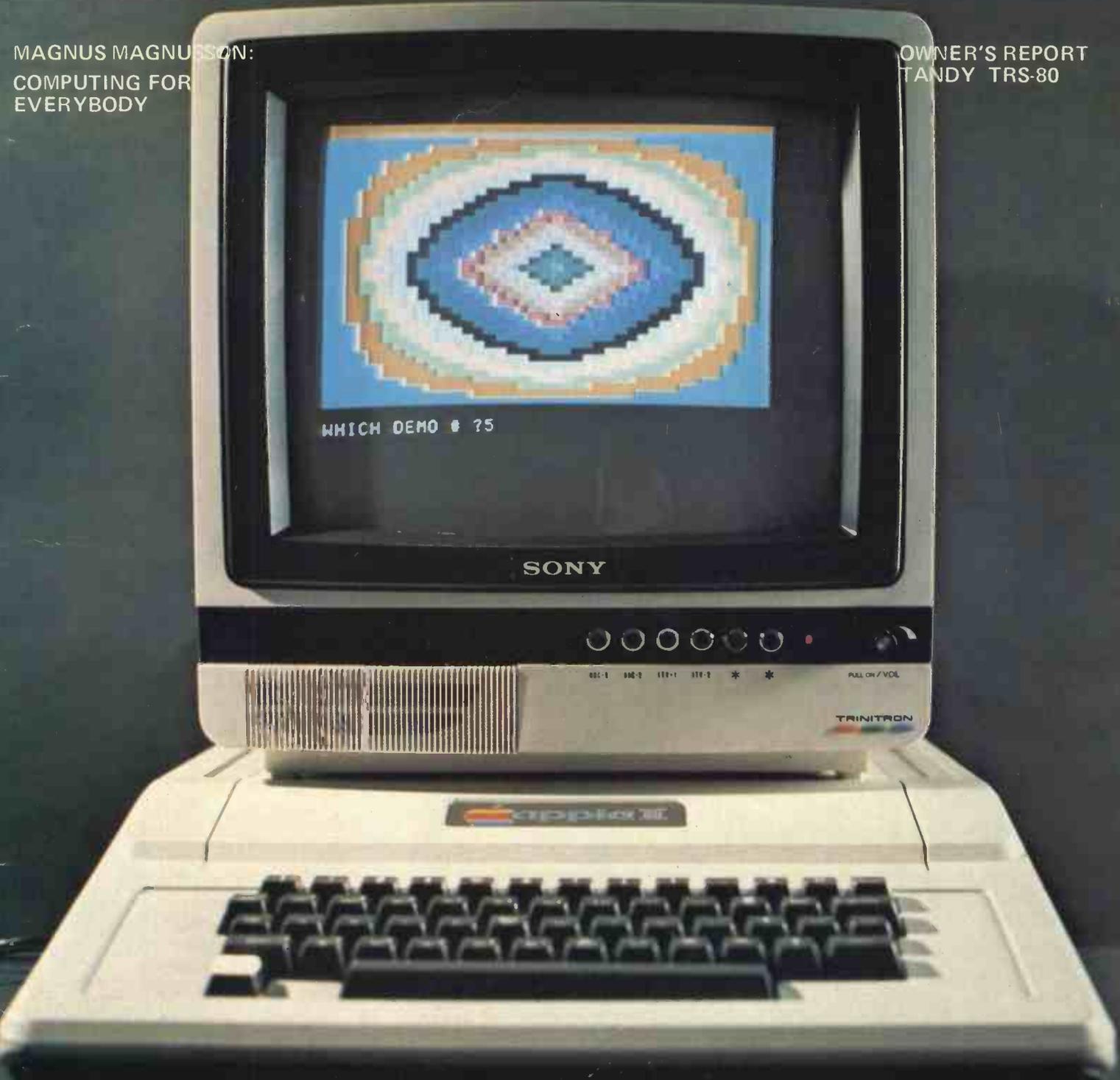
- IT'S THE THOUGHT THAT COUNTS
- A COMPUTER OR A KIT
- TAKE IT OR LEAVE IT
- HANGMAN
- TOO HIGH A PRICE

AUGUST 1978

Europe's first magazine for personal computers for home and business use

MAGNUS MAGNUSSON:  
COMPUTING FOR  
EVERYBODY

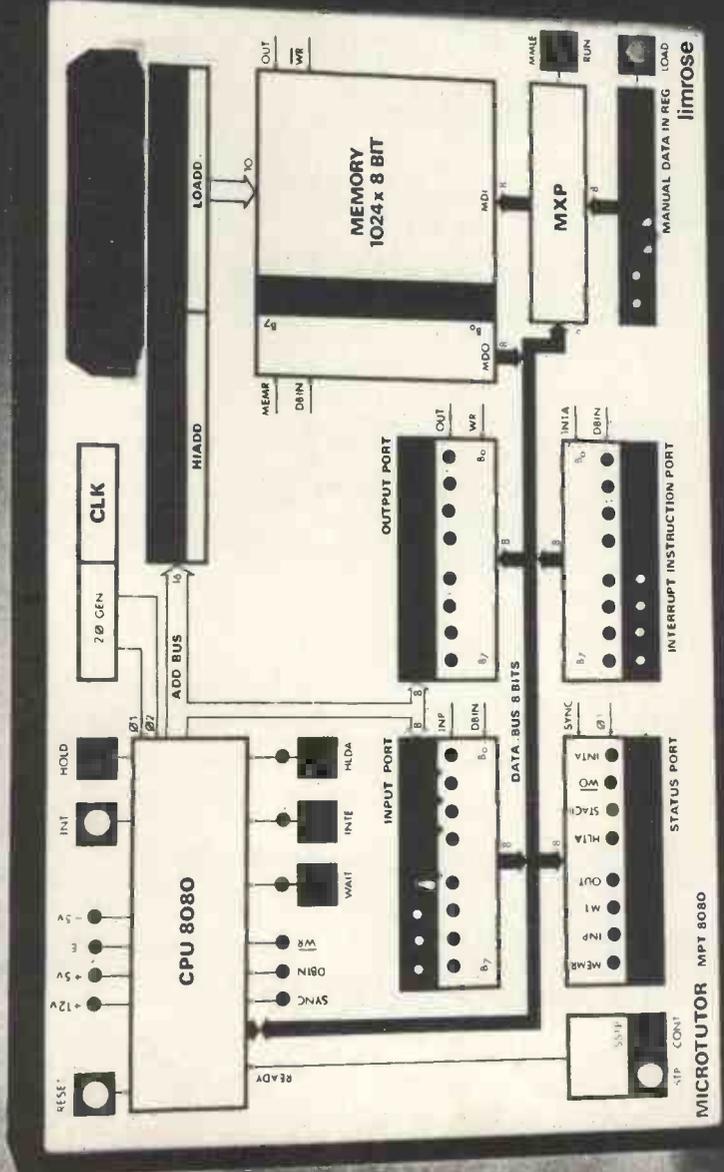
OWNER'S REPORT  
TANDY TRS-80



## COLOUR IS AN APPLE II

# THE LIMROSE MICROTUTOR 8080

New, Fast and low cost method for learning the "how" of microprocessors



Now... there is a new, fast, simple and inexpensive MICROTUTOR to give you truly "hands on" experience to master and apply microprocessors.

Limrose's MICROTUTOR MPT 8080 has been chosen by the British Post Office, and many other large companies in U.K. and overseas, to train their engineers in this new and exciting technology.

The MICROTUTOR MPT 8080 comes ready to use. Nothing else to buy or debug. It's front panel has all you need to understand how the industry-standard 8080 microprocessor works. And... you can expand it later to use as a 'Development System'.

It's inexpensive, it's versatile and comprehensive.

It has a full range of expansion modules, such as expansion motherboard, RAM, ROM, Teletype and VDU interfaces, Cassette-Interface, EPROM and PROM Programmers, and even an inexpensive EPROM erasure.

And, of course, the MICROTUTOR MPT 8080 is supplied with a comprehensive Instruction Book. By following the Instruction Book, most persons with limited technical knowledge can rapidly learn how microprocessors work.

AND... IT IS BRITISH.-MADE !!

- \* 8-bit Microcomputer with 1K RAM
- \* 8-bit Input Port
- \* 8-bit Output Port
- \* 8-bit Status Port
- \* Manual or Teletype Operation.
- \* Full vectored- Interrupts
- \* Buffered Data and Address Buses
- \* Advanced "Real-time" monitor
- \* Expandable with plug-in cards
- \* Comprehensive Instruction Book

**limrose electronics limited**  
 241-243 Manchester Road, Northwich, Ches., CW9 7NE. Tel. 0606 41696/7



# PERSONAL COMPUTER WORLD

Europe's first magazine for personal computers for home and business use

UK 50p  
US \$2.00

Vol 1, No 4  
August 1978

## CONTENTS

PUBLISHER'S LETTER.....	4	TAKE IT OR LEAVE IT <i>Roger Hargrave</i> The Game of NIM, with special regard to flow-charting.....	27
EDITORIAL PAGE.....	6	SHOWING CHARACTERS <i>Neil Harrison</i> A VDU driver for Z80 based micros.....	31
LETTERS.....	8	HANGMAN <i>Mike Dennis</i> Entertainment for the literate.....	35
TIDBITS Products...company news...services.....	10	GETTING IT TOGETHER <i>Mike Banahan</i> How to assemble assemblers.....	39
IT'S THE THOUGHT THAT COUNTS <i>W.V. Ringer</i> A fresh insight into the robot-man evolutionary process.....	14	PCW BOOK REVIEW.....	41
GETTING INTO NEIGHBOURHOOD CONSULTANCY <i>David Francis</i> Cows...Wellington boots...farming...business!.....	16	PCW OPEN PAGE <i>Mike Lord</i> The ACC...and other items.....	42
The R.O.M. FACTOR <i>J. Gregory</i> A tale of suspense that has a tragic end in the streets of Manchester.....	18	OWNER'S REPORT <i>A. J. Harding</i> A Dandy...the Tandy TRS-80.....	44
FROM IDEAS TO PROGRAMS Very simple...very basic...very useful.....	22	COLOUR IS AN APPLE II <i>John Coll &amp; Charles Sweeten</i> An Apple Newton could have done with.....	50
A COMPUTER OR A KIT? <i>Michael James</i> <i>Caveat</i> builder.....	23	TOO HIGH A PRICE <i>Dr. John Dawson</i> Information society or informer's society?.....	56
A MIGHTY MICROMITE <i>C. Corbett</i> The KIM, a best seller sure to go through many new editions.....	24	COMPUTING FOR EVERYBODY Edited version of a talk given at the opening of Computer Workshop, Manchester, by the Rector of Edinburgh University.....	58

*Editorial and Advertising Office:*  
62A Westbourne Grove, London W2  
Phone: 01-229 5599 (Publisher)  
01-727 8758 (Editorial and Advertising)

**Publisher:**  
A. Zgorelec  
**Editor:**  
Meyer N. Solomon  
**Policy Advisor:**  
Peter Crofton-Sleigh, FRAS

DOTPIKS © K. HAMME  
CARTOONS © GIGO & SAM

### CONTRIBUTORS:

We welcome interesting articles written simply and clearly. You need not be a specialist to write for us. MS should not be more than 3000 words long, lines double spaced, with wide margins. Line drawings and photographs wherever possible. Enclose a stamped self-addressed envelope if you would like your article returned.

Manufacturers, suppliers and dealers are welcome to contribute technical articles, and send product information, but we are pledged to an independent viewpoint and will publish evaluations and reasoned criticism or praise, space permitting. Naturally there will be right of reply. Views expressed in articles are not necessarily those of Personal Computer World.

We may make arrangements to offer our readers products at special prices, for a limited period, in line with the policy outlined above.

Published monthly by Intra Press, 62A Westbourne Grove, London W2. Phone: 01-229 5599. Contents fully protected by copyright. All rights reserved. Subscription rates: Britain £8 for 12 issues. Prices include postage. USA - \$10 for six issues, \$20 for 12 issues. Continent and elsewhere: £9.80 for twelve issues. Prices include postage. Printed by Carlisle Web Offset, 55 Conduit Street, Newtown Trading Estate, Carlisle CA2 7NR. Sole UK Distributors: Seymour Press Ltd., 334 Brixton Road, London S.W.9., England. Distribution to specialist shops by Intra Press.

Dear Reader,

We are constantly amazed at the variety of exotic places like the Solomon Islands, St. Lucia, Panama and New Guinea from which we get requests for subscriptions.

Businessmen and tourists staying at those London hotels (over thirty) at which PCW is available write to us when back home and take out subscriptions. Sometimes we get letters asking us about the availability of products and services, and naturally we help as much as possible.

A few days back we met an elderly computer fanatic from abroad. He wanted to know where he could get one of the new Personal Computers he's read about in PCW.

We gave him a few addresses, he thanked us, and mentioned that his problems weren't over yet, though he's thought of a solution.

Apparently, his country does not allow private individuals to import computers.

But, electronic components are a different matter. So, one computer freak has bought himself a machine, stripped it and disassembled it. Somewhere in a London hotel there is an empty computer case, and somewhere thousands of miles away there's a homebuilt cabinet housing a powerful little beast.

HEXADECIMAL KEYBOARDS	£17.50
Fully encoded TTL output with strobe. High quality keys on fibreglass board.	
Individual keys including blank	70p or £5.00 for 10.
2 digit hex display with latch	£9.00.
5V 2A PSU in case	£16.00.
Motorola D2 kit built and tested	£230.00.

Prices include P. & P. Write for details.

LINTRONICS, 313 Green Lane, Ilford, Essex.  
MAIL ORDER ONLY.

**The NORTH STAR HORIZON Computer**

HORIZON — a complete high performance micro-computer system with Integrated floppy disk memory.

HORIZON is attractive, professionally engineered ideal for business, educational and personal applications.

To begin programming in EXTENDED BASIC, merely add a CRT, teletype or other hard-copy terminal. HORIZON-1 includes a Z-80A processor, 16K RAM, minifloppy disk and 12-slot S-100 motherboard with serial terminal interface, all standard equipment.

and software, too!

HORIZON includes the North Star disk operating system and full EXTENDED BASIC from diskette ready at power-on.

This Basic, now in widespread use, has virtually everything desired in a Basic, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language call, and more.

OPTIONAL software (under CP/M) include — C-Basic Compiler/Interpreter Basic, Microsoft Disk Extended Basic, MAC Macro Assembler, Microsoft COBOL-80 and FORTRAN-80, and more.

**QUALITY AT THE RIGHT PRICE:**

HORIZON Z-80A processor board, RAM FPB, and micro disk system can be bought separately for either Z80 or 8080 S-100 bus systems.

Special introductory prices for PCW readers, as follows:

- HORIZON-1: £1275 assembled and tested.
- HORIZON-2: £1550 assembled and tested.
- 8K, 250nS Static RAM: £145 — 16K, 250nS Static RAM: £295 — FPB: £215 — Z80A Board: £185 — 2nd Serial I/O: £45 — Parallel I/O: £45.

All prices are for assembled and tested units, exclusive of VAT and carriage. Subject to change.

Full details: PCW, Interam Computer Systems Ltd., 59 Moreton Street, Victoria, London SW1V 2NY. Tel: 01-834 0261/2733

Businessman interested in Microcomputer business, hobby and games applications would like to contact interested persons to discuss feasibility of opening a shop in Central London on a partnership basis. Must be able to bring some capital to the enterprise as well as a good knowledge of Micro Hardware and Software. Enquiries will be treated with the strictest confidence. Please write to: Box 7, PCW, 62a Westbourne Grove, London W2.

# XITAN SYSTEMS

## A comart DISTRIBUTER

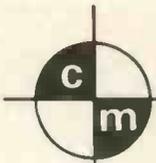
Z80 SYSTEMS  
OUR SPECIALITY

From £1539 ex vat.

CROMEMCO  
NORTH STAR  
LEAR SIEGLER  
INFO 2000  
DYNABYTE  
AXIOM EX800  
MODEL 43 TELETYPE



31 ELPHINSTONE ROAD, HIGHCLIFFE,  
DORSET, BH23 5LL.  
Phone (04252) 77126



# comart

## SPECIALISTS IN MICROCOMPUTERS

### PERFORMANCE WITH STYLE

In the home, office  
or laboratory everyone  
will admire your SOL!  
Yet it's powerful,  
easy to use, and  
very, very versatile.



#### The SOL Specification

Bus: S100  
CPU: 8080A  
Memory: 16K byte standard expandable  
to 64K  
Video: Standard. 16 line x 64 characters  
Keyboard: Standard. Upper/Lower case +  
numerics  
Serial Interface: Standard. RS232C or  
20mA  
Parallel Interface: Standard. 8 bits in and  
out  
PROM Monitor: Standard – SOLOS  
Cassette Interface: 120 char/sec CUTS  
or 30 char/sec Kansas City formats  
Power: 230v 50Hz 150w

The SOL Terminal Computer System  
may be used with cassette or diskette;  
on its own or linked to other  
computers.

SOL Computer Systems never grow  
old. Add new S100 modules to up-  
date and expand your computers'  
power when you need it.

Contact us direct or call your nearest Comart SOL Dealer:

NewBear Computing Store, Bone Lane, Newbury, Berks (0635 49223);  
The Byte Shop, 426/428 Cranbrook Road, Ilford, Essex (01-554 2177)  
Xitan Systems, 31 Elphinstone Road, Highcliffe, Dorset (04252 77126)

Comart Limited, P.O. Box 2, St. Neots, Cambs. PE19 4NY  
Tel: Huntingdon (0480) 215005

# Editorial Page

## Our greatest natural resource: people

Here we are in Europe, supposedly in the doldrums, and from all over the place we are receiving articles, ideas and suggestions which are the equal of anything the Americans have to offer. And that, in my opinion, is where our future lies: using components from elsewhere, designing our own.

## Friends

*John Anderson* of J.A. Computers is one of the breed of people who seem to abound in the microworld. He came down to demonstrate his "Merlin" and spent a couple of hours fiddling around with our computer, which is a bit of a recalcitrant beast. No sales pitch, no attempt to influence us: just plain unalloyed friendliness and a desire to help.

I could fill this page with the names of people who, though in business and with a shrewd (and necessary) eye for publicity, are approachable and helpful.

## Winners

Copies of the special edition of *Scientific American* are being sent to the following:

Philip Fox (Age 14)  
28 Western Avenue  
Riddlesden  
Keighley  
West Yorkshire BD20 5DJ

John Allen  
46 Needwood Road  
Elms Farm Estate  
Bedford

Daniel Wolpert (Age 14)  
32 Steeles Road  
London NW3 4RE

Phil Oldfield  
3 North Pole Road  
London W10

Rick Barnett  
131 Rushden Gardens  
Clayhall  
Ilford  
Essex IG5 1BN

Bertram A. Martin  
99 Northdown Road  
Solihull  
West Midlands

D. A. Hills  
19 St. Anthony's Drive  
Chelmsford  
Essex CM2 9EG

L. K. Darnton  
6/2 Harewood Drive  
Edinburgh EH16 4NQ

A. S. Bluck  
Unilever Research Ltd.  
Colworth House  
Sharnbrook  
Bedfordshire

G. D. Hayes  
42 North View Crescent  
Epsom Downs  
Surrey

Max Hadley  
64 Alma Road  
Southampton SO2 1BR

A. J. Borer  
Shorrock Developments Ltd.  
Shadsworth Road  
Blackburn  
Lancs BB1 2PR

co-authors:  
a) M. Groom  
20 Brockwell Gardens  
London SE24

b) J. McKinna  
127 Burbage Road  
London SE21

Rex W. Godby  
16 Williamson Road  
Bristol BS7 9BH

F. S. Dewhirst  
47 Grafton Road  
Keighley  
West Yorkshire BD21 1LJ

J. P. Fitzgerald  
34 Kenilworth Road  
Ealing  
London W5

Phil Cornes  
16 Somerford Avenue  
Crewe  
Cheshire CW2 8NE

S. J. Randall  
53 Moresby Avenue  
Surbiton  
Surrey KT5 9DT

Donald Milne  
58b Marquis Road  
Woodside  
Aberdeen

Steve Overy  
6 Stafford Street  
Barwell  
Leics

## New Consultants

*David Hebditch* is a data processing professional specialising in communication-based systems. Management Information Systems and Computer Security. He illustrates his own articles with cartoons and drawings. He will be a speaker at our exhibition.

*Sheridan Williams* has already appeared as an author in *PCW*. He will be a speaker at our exhibition.

## Advertisers

It pays to advertise in *PCW*.

- ★ It's the magazine people want to keep.
- ★ A highly intelligent and articulate readership with purchasing power to match.
- ★ Unique in Europe, from the quality of its content to its widespread distribution: on sale to the public at all good newsagents.
- ★ On sale at specialist shops.
- ★ Available at over thirty prestigious hotels.
- ★ Estimated sales of No. 2 - 27,000
- ★ Over 3,000 subscriptions.

It pays to advertise in *PCW*.

Contact the Advertising Department at 62A Westbourne Grove, London W2.  
OR 'phone 01-727 8758.

## Subscriptions

Having difficulty in ensuring your copy of *PCW*?

You can make sure of getting your copy by taking out a subscription. Rates (for twelve issues):

UK - £8

USA - \$20

The Continent and elsewhere - £9.80

Subscriptions payable to Intra Press, 62A Westbourne Grove, London, W2.

*Back numbers:* Available to callers at the above address (newsagents, ground floor) or by post for 65p.

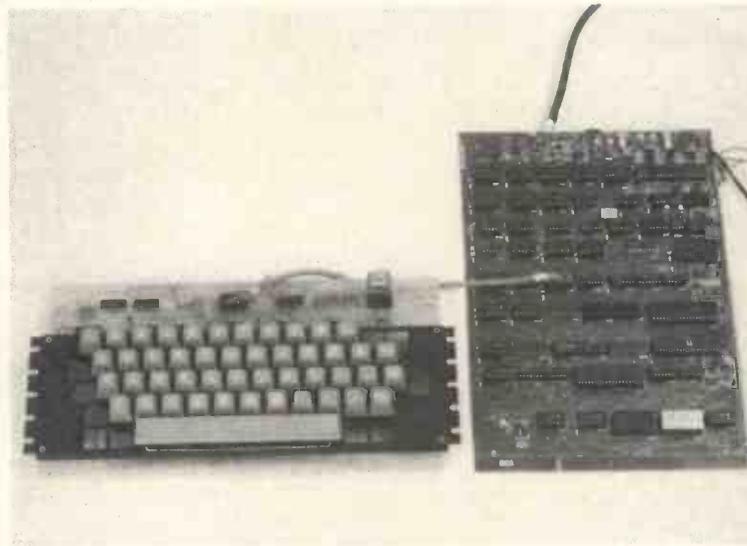
## SCHOOLSOFT

Exciting new programs on cassette for the Commodore Pet; -

1. Devil's vermin (£4.50) - hunt these elusive rodents.
2. Simuland (£4.50) - set yaw, pitch and throttle to land your plane in varying weather conditions.
3. Rockets (£4.00) - for two people. Avoid crashing and you win. Produces interesting patterns.
4. Minefield (£3.50)
5. Games collection - Hi-lo, beat the computer down from 27 to 1, noughts and crosses etc. (£4.00)

All these programs have been carefully tested on a wide range of people ranging from idiots to boffins. Everyone enjoyed them and we hope you will too.

Send your cheques, postal orders etc payable to 'Schoolsoft' c/o 67 Anderby Drive, Grimsby, S. Humbs., DN37 9ET.



## NASCOM 1 Z80 MICROCOMPUTER KIT

**Includes interface for:**  
 TV or Monitor — cassette,  
 dump — Teletype — spare,  
 PIO,  
 Expansion RAM-  
 BOARD up to 32K.

**Potential:**  
 High Level Language,  
 Mini Floppy Disk,  
 Excellent Z80  
 evaluation kit.  
 £197.50+VAT.

Write for details or send order to:  
 NASCO SALES LIMITED, Dept. PCW ,92 BROAD STREET,  
 CHESHAM, BUCKS. Tel: 02405 75151, Telex: 837571.

# Letters

## Benchmarks

Congratulations to Mr. Murrell (PCW Letters Vol 1 No. 2) for spotting the apparent inconsistency in benchmarks for BASIC running on a PDP 8/E (Vol 1 No. 1) I can assure him however that BM4 and BM5 times are neither misprints nor mistakes. I have tried both benchmarks again, deliberately loading the system more heavily for BM4 than for BM5. This I am sure is the reason for the discrepancy when the original tests were made. The new results inserted into the chart make interesting reading. BM4 took 41 seconds, sharing the machine with five Fortran compilations running on other terminals, while BM5, having the whole machine to itself, took 8.3 seconds.

Pete Harris,  
Warley College.

## A Dearth of Software

The recent developments in the microcomputer field have led to the availability of a large number of microsystems designed for personal use. Prices of such systems are well within the reach of the individual.

It seems, however, that hardware is readily available but that suitable software is not. All systems appear to have some form of monitor or operating system, but only those whose price is outside the range of the average user have provision of a high level language.

Prospective microcomputer purchasers whose previous experience has been on mainframe machines operating in high level language are therefore faced with either learning machine code (a grim prospect in some cases) or engaging in a search for a high level language system suitable for their application.

I would therefore suggest that an article giving details of currently available language systems, particularly of machines with which they will interface, would be of considerable value to mainframe-spoiled Extended Basic Freaks such as myself.

Charles F.L. Heritage  
Hurdle House, 1 Chestnut Lane,  
Amersham Bucks HP6 6EN

## The disabled and the computer.

Being extremely green at this computer game and having no hardware experience (come to think of it no experience whatsoever), I have only had my Apple about 3 weeks, but I have a glimpse of the tremendous possibilities now open to me.

Since breaking my neck whilst on National Service in Germany, I've been able to move my head, left, right and slightly backwards and forwards, and that's my lot.

If anyone tries lying on the floor flat on his back using the above movements and tries writing his name with a pen in his mouth (holding a pad above) they will quickly realise there's not a big variety of things someone in my position can do.

Now imagine the same person with the keyboard of a computer in front of him, you find the limits go up tenfold.

I know there are machines like "Possom" issued to certain cases, by the Government. It will connect the disabled to T.V., telephone, it will open doors and curtains, and work a typewriter, about a dozen functions and costs between £1,400 - £1,600, depending on the number of functions.

The processor I have is an "Apple II", 16K, and because I have to work with a stick in my mouth, the shift and control keys have had to be modified so that they work like a typewriter and use two movements to lock and release.

The Apple has opened up a new world I didn't know existed, it now makes jobs possible, the design of colour graphics, and, all the games (and don't knock the games, remember your capabilities while lying on the floor) and all this at a touch with a stick, and at costs comparable with "Possom".

All I have to do now is buckle down and do a lot of studying and practising.

I hope this gives some of your readers a glimpse of what a computer could do for the disabled and severely disabled. Also wonder if they have the same problem I have of doing a bit more, and a bit more, so find it's 3 a.m. or later and that you have to force yourself to turn the switch off.

Charles Smith  
22 West Court  
The Thistle Foundation  
Edinburgh EH16 4EB.

Dr. Johnson's article in your issue No. 2 is a nostalgic reminder of the great days of medical computing in the early sixties carried away on a euphoric trip. Every medical journal especially in the States carried articles such as this describing all the wonders to come.

Since those halcyon days reality has raised its ugly head and applied the kiss of death to these pipe dreams and schemes, and the great systems in Southern California, Massachusetts General Hospital, Kings College Hospital, Danderyd Stockholm and many others no longer appear on the front pages.

Of course, all these wonderful facilities should and could be available to us doctors. Unfortunately, Dr. Johnson gives the impression of having failed to analyse the reason why they have not and puts the blame on the usual scapegoat of medical inefficiency - the DHSS.

The reasons, well, an oversimplification would be to narrow it down to two main factors, the large main frames we had to deal with which under the blinkered eye of the DHSS were mainly concerned with non-medical management, payroll and statistics. The second was the doctors and their refusal to change so that the major systems crashed because the doctors refused to input clinical details. The excuse - it was too time consuming and too complicated and they could not input data that they considered was important. Too much detail was asked for by the computer specialists and in a busy general practice or medical unit this is understandable.

But what about the essential component of medical practice, the patient's record? In the early sixties an editorial in the American Journal of Electronics, I think it was, stated that they had no objection to every doctor writing an essay on each patient as long as they were prepared to throw it in the trash can and write down twelve keywords describing the patients' unstable system, for immediate use and future recall. Now this strategy formed the basis of our computer medical summary in the Community Health Information Project (CHIP) and of the DHSS major projects of the 60's, this record being exchanged as required between the G.P.s and hospital.

But with computers you only get out what you put in and the final problem is the input one; and what was to

be put in to this record which in a hospital folder may be two inches thick?

Dr. Johnson mentioned 25,000 G.P.s; how many of these are dedicated computer men? I would wager about 1%, they will be prepared like the Exeter group to use a keyboard to input and access the record and amend where required. The essence of the problem is that these 25,000 G.P.s, many of whom never record, will have to be won over with a very simple system.

Because of these problems, since 1970 Dr. Chris Evans and I have been experimenting with a simple keyword summary using computer interrogation of patients for their clinical history. This has been accepted by patients and doctors and today the patients attending the cardio-respiratory follow up and occupational disease clinics in the Department of Medicine at Charing Cross Hospital, are interrogated in this manner — a key word summary being printed out at the end of the interview, presented to the examining doctor and then goes in the record — computer storage of this later when the doctors have been won over to this technique.

This technique rather bypasses the doctor and since computer doctor symbiosis is the road to the future in medicine, we are now using a Computer Workshop MICKIE with voice input which will be used by the doctor during the whole consultation, the positive laboratory and other results will be blended with the record as the patient and doctor, as it were, pass along the diagnostic and care information pathway assisted at every step by the computer.

Let's have more articles on what has been done rather than on what might be done in the sweet bye and bye in medicine.

H.C. Price  
Consultant Physician  
Western Hospital  
Seagrave Road, London SW6 1RZ

#### Not so small but just as beautiful

I hope that the term "Personal Computer" does not mean that the physical size of the computer is restricted to micro computers. I know people who have "Personal Computers" that would occupy most of a garage for a car. These people own Elliott 803, 503, or 903 computers. There is an active user's group for these people that is well established, and has a wealth of experience going back to before 1965. There are a great many users (mostly in schools and colleges) still in existence. It is also possible to pick up these computers for next to nothing (certainly far less than £100). Perhaps the Elliott 803 or 903 user's group will respond to this letter and hopefully these wonderful computers will continue to find homes instead of going for scrap.

Sheridan Williams  
114, Beech Road,  
Saint Albans,  
Hertfordshire. AL3 5AU

Congratulations on the launching of PCW which looks to be just what's needed, I hope that when you become firmly established you will carry the odd article on hardware construction, so that we can find applications for our shiny new machines (those who have them!). One point I would like to make on the article "The Complete Kit Builder", and that is the fact that no mention has been made of that marvellous device — the "solder-sucker". Initially the cost may seem high (around £5) but now I wouldn't even loan it for fear of having to use solder braid etc. Using it is simple, hold the solder sucker very close to the joint and heat the joint with the soldering iron. As soon as the solder melts, press the plunger release, nine times out of ten every scrap of solder is removed, the tenth time another go will suffice. I have found this indispensable, a board full of TTL chips, all salvaged, even a 40 pin IC socket

removed without damage to board or socket. For any form of prototype work the saving in time and patience over using braid is well worthwhile.

Another suggestion for handling small CMOS chips — try using a large bulldog clip (with a weak spring) to short out all the pins, thus avoiding static problems.

Once again congratulations on a new and worthwhile journal.

D.J. Bunyan  
11, Sydney Avenue.  
SITTINGBOURNE  
Kent.

#### The EUROPA Bus

I read with interest the article on the future Euro-Bus, but would like to make some comments on the proposals, both from an amateur and professional view-point.

Using indirect connectors has both advantages and disadvantages, but appeals to the amateur more than the professional. However, if the option of 64 or 96 pin connectors is to be given, then surely the third row should be allowed as an expansion row for future processors (and even some existing ones).

The limitations imposed by Z80 design with 6800 as secondary appears to be short-sighted in this respect. The possibility of multi-processor systems should also be borne in mind.

For these reasons I would suggest the third row be allocated by experts in other types of processors if and when necessary, but agree that the pins so allocated be accepted (so far as is practicable) as a mandatory standard.

The option of using the second 64/96 way connector for bus signals (the 9900 CRU signal is given as example) is not a good idea, as the 9900 user may subvert one of the other lines rather than go to the expense of an extra connector. In addition, with the advent of smaller and better I.C.s it is quite easy to fit a large peripheral controller or many small interfaces on a single board. 64 pins is not many for peripherals of this nature, and it is easy to use 96 pins in a large system or, indeed, in a small multiplexer system. The amateur and cost conscious professional alike would rather use one full board than two or three half full ones which have restrictions placed on connectors by bus signals.

For these reasons I would suggest that the second connector be totally allocated for user purposes, and that the option of 64 or 96 pins should be given.

On the subject of signal types I would in general agree with the authors, except that with the advent of octal 74LS devices such as the 244, 245, 374 etc. it would be better to have positive true busses for the data and address lines. These devices are capable of driving large numbers of 74LS and NMOS devices without extra buffering, making for cheaper systems in the amateur and dedicated processor applications. The only alternative to this is the open collector bus as used in DEC machines. This is probably preferable, except that I.C. makers do not produce such devices in octal format, this doubling the quantity of interface devices and amount of wiring to them.

Although the authors allow for positive or negative convention busses in the article, this latitude would surely inhibit the appearance of standard modules on the market, as has happened in the case of S100 for example. Whilst some small dedicated systems will almost certainly deviate from any standard proposed, the official bus, to be of any real value, should be of fixed but expandable format.

To conclude, the Euro-Bus E78 would appear to be an answer to a much needed standard, but requires more thought and investigation before final adoption.

D.B. Stiles  
44 Kellaway Avenue,  
Bristol, BS6 7XU

# Tid Bits

## PRODUCTS . . . COMPANY NEWS . . .

### Millennium Appoint Microsystem Services

Microsystem Services have been appointed as an exclusive UK representatives of the American company Millennium. Millennium themselves have recently been acquired by American Microsystems Inc. but will continue to function as an independent company.

Millennium are five years old and, until now, have sold products on an OEM basis to such companies as Signetics, Intel and Tektronix. Their main claims to fame include the design of the first floppy disk development system and the first universal development system. Their latest product, the  $\mu$ SA microprocessor system analyser, is unique in that it combines in-circuit emulation and 'signature analysis' to provide a very powerful diagnostic tool.

Microsystem Services are already selling equipment to microcomputer users and microcomputer system developers and are confident that they have a ready market for the  $\mu$ SA analyser.

The main part of Microsystem Services business, to date, has been in marketing and providing service and support for the Data I/O range of PROM programmers. A few months ago they took on a range of 'ruggedized' NOVA-compatible minicomputers manufactured by the Rolm Corporation of America, in which great interest is being shown.

EH Research are also a recent acquisition and Microsystem Services are now offering a very advanced range of EH pulse analysers and generators. The final product offered by Microsystem Services is a temperature forcing rig from Thermonics, which enables ICs and small components to be tested over a very wide temperature range.

Microsystem Services are delighted with the agreement with Millennium, since it enables them to offer a unique and very powerful microcomputer test and development tool at a reasonable price, and they expect that the market for the  $\mu$ SA in the UK will be substantial.

Further information: Jim Knott

Microsystem Services  
Duke Street  
High Wycombe, Bucks  
Telephone: (0494) 41661

### New Low-Cost, Professional-Quality ASCII Keyboards

Electronic Brokers Ltd. is marketing a new ASCII keyboard designed especially to suit the needs of both personal and industrial users.

The model KB756 provides encoding for all 128 ASCII characters and control functions, imposing no limitations on software design or hardware capability.

Utilizing reliable gold-contact low-bounce keyswitches and low-power MOS encoder circuitry, the KB756 is aimed at bridging the gap between basic non-coded keyboards and expensive OEM models. Construction is of high-grade components throughout mounted on a rugged printed-circuit board.



The versatile interface allows user selection of parity, positive or negative logic data and strobe outputs, alpha lock operation and both D.C. level and pulse strobe signals. A latching shift lock key is included, and all outputs are TTL-DTL-MOS compatible.

The KB756 is economically priced at £60.00, and discounts are available for quantity.

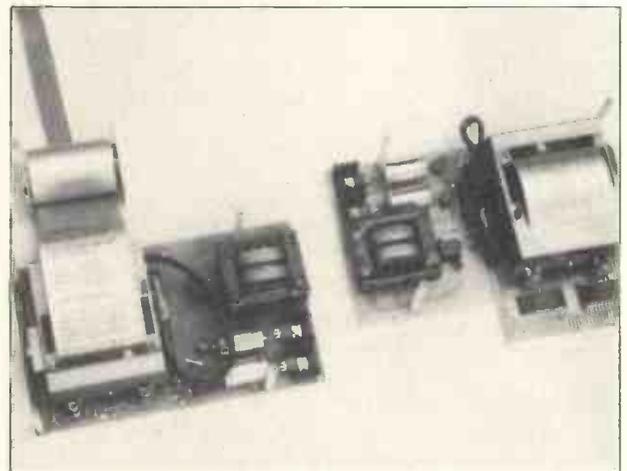
Electronic Brokers is also stocking a range of optional accessories including edge connector, mounting frame and enclosure cases.

For further information contact: Sally Carpenter  
Electronic Brokers Ltd.  
49/53 Pancras Road London NW1 2QB England  
Telephone: 01-837 7781

### New 411 Low Cost Panel — Mounting Digital Printer

Datac Limited announce a new low-cost panel-mounting printer unit.

Selling for under £200 one-off, the Datac 411 is an assembly designed to fit into the user's equipment, and can be fitted in either panel-mounting or desk top applications. It incorporates the 245 electrosensitive printer mechanism, and a single electronics printed circuit board which provides interfacing, control, character generation and drive circuitry as well as an integral mains power supply. Optionally a dc:dc converter plug-on module is available for +5V d.c. input applications.



Two standard input interfaces are available: 6 bit parallel (ASCII), character serial, which will print 20, 32 or even 40 characters per line; and 4 bit parallel (BCD) x 16 column parallel. The character serial interface is ideally suited to microprocessor-based systems and equipment where a data highway is used to transfer the information, while the BCD interface is easily connected to DVM's, counters, clocks, etc.

Contact: M J Robinson  
Datac Limited  
Tudor Road  
Altrincham, Cheshire WA14 5TN  
Tel: 061.941.2361/2

### Specialist Equipment for Post Office Viewdata or Press-Tel<sup>©</sup> Service

Jasmin Electronics Ltd., announce a bright new star in their range of digital systems.

The Viewdata Editing Terminal, or V.E.T. for short is a free standing microprocessor based, remote job preparation and information transfer system, designed initially for use by the information provider community.

Through the use of reprogrammable memory packages, and modular construction techniques, V.E.T. can be configured to perform a variety of tasks. In the case of the information providers, this allows the operator to proceed with the making up of any number of pages of viewdata or prestel<sup>©</sup> information, without need for direct communication with the main computer. During off peak periods, the already prepared data can be relayed into the view data system, without operator intervention. The V.E.T. also contains powerful operator prompt and editing firmware, with such features as page header identification formats, insert and delete blocks, lines and characters, transpose text, and open or close gaps in pages. The Jasmin programming design and implementation team are working closely with the Post Office research team to ensure that the editing functions in the terminals remains in line with the requirements of the information providers. A unique feature available for a small annual charge ensures that all V.E.T. user equipment will contain the

very latest operations firmware, thus ensuring that all providers are capable of making the most of viewdata.

The V.E.T. system can also be used in a variety of other roles, specialist visual display terminals in colour with graphics and, if necessary, data recording facilities, computer remote job entry terminals, specialist design centres, data loggers, and even hardcopy output terminals. The accompanying brochure gives a detailed description of the system and an easily assimilated price list with the various options costed separately.

Should any further information be required on the V.E.T. equipment, or other specialist equipment needed for Viewdata or Teletext applications, please contact:

**Sam Rhys-Williams, Marketing,**  
**Adrian Holmes, Technical Sales.**  
at: 65 Wharf Street South,  
Leicester LE1 2AD  
Tel: (0553) 58128/9

#### New Product

#### N-Channel 128 x 8 bit RAM I/O From Bywood – Type ISP-8A/650

Manufactured by National Semiconductors Ltd., the ISP-8A/650 RAM Input/Output chip is an LSI device which provides random memory and peripheral interfacing for microcomputer systems. The RAM portion contains 1024 bits of static RAM organised as 128 x 8. Since RAM is fully static, no refresh of clocks are required. Data out of the RAM is of the same polarity as data in and read out is nondestructive.

Available from Bywood Electronics the I/O portion consists of two peripheral parts of eight bits each. Any of the I/O pins in the two parts may be defined as an input or an output to provide maximum flexibility. Each part may be read from or written to in a parallel (8-bits byte) mode. In control-based applications, a single bit of I/O in either part may be set, cleared or read with a single micro-processor instruction. This facility improves efficiency and simplifies programming.

In addition to basic I/O, one of the parts, part A, may be programmed to operate in several types of strobed mode with handshake. Strobed mode together with optical interrupt operation permit both high speed parallel data transfers and interface to a wide variety of peripherals with no external logic.

The RAM I/O is an n-channel silicon gate device packaged in a 40-pin, DIL ceramic or plastic package. The ISP-8A/650 operates with a single 5-volt power supply, is fully TTL compatible, can be interfaced directly with the SC/MP-2 micro-processor and features independent operation of RAM and I/O.

Further information from:

**Mr. J.H. Miller-Kirkpatrick,**  
Managing Director,  
Bywood Electronics.

Telephone: Hemel Hempstead (0442) 62757

#### Commodore set up first Computer Dealership Network in the U.K.

In answer to the growing demand for Commodore's highly successful £695 PET personal computer, Commodore Systems Division has set up a carefully selected network of some 25 authorised PET dealers who will be providing regional sales and systems support throughout the U.K.

Kit Spencer, General Manager of Commodore's System Division, outlines some of the terms of the agreement:

"We have reached certain decisions about the type of dealers that we feel would be right to handle our PET computer and allied products which will shortly be available. We regard a Commodore Dealership as a long term agreement and therefore we must insist that our dealers meet the following criteria:

**Computer Knowledge and Understanding of BASIC are essential.** We make it a condition that the Commodore Computer Dealer must have at least one capable programmer on his staff. With regard to *Service and Systems Support*, we expect the dealer to provide this both during and after the 90 days warranty period and to hold sufficient spare parts and machines in stock to ensure this service.

"Our dealers must be capable of demonstrating, exhibiting and actively developing the market for a full range of Commodore Systems Products and be prepared to do volume business in excess of that so far believed possible in small computers. We make it a condition that our equipment is on permanent demonstration open to all".

"It is intended that each dealer appointed should be a self-contained unit able to offer a full sales, service and back-up on a



Outside Commodore's PET Personal Computer Centre Kit Spencer, (carrying PET) Derek Rowe (Applications) and a Pet User.

local basis to the potential user.

"We are pleased to announce our initial selection of Computer Dealers who will now market, sell and service our Computer Products under these conditions:

#### LONDON AND HOME COUNTIES:

- C.S.S. Systems Ltd (London)
- Sumlock Bondain Ltd (London)
- The Byte Shop (Ilford)

#### WEST COUNTRY:

- Computabits Ltd (Somerset)
- C.S.S. Systems (Bristol)

#### NORTH WEST:

- Cytek (UK) Ltd (Manchester)
- Rockcliff Brothers Ltd (Liverpool)
- D.A.M.S. Office Equipment Ltd (Liverpool)

#### MIDLANDS:

- Memec Systems Ltd (Thame)
- Taylor-Wilson Systems Ltd (Solihull)
- Betos Systems Ltd (Nottingham)

#### HUMBERSIDE:

- A.T.A. Management Consultants (South Humberside)
- Microprocessor Services Ltd (North Humberside)

#### WALES:

- G.R. Electronics Ltd (Newport)

#### CHANNEL ISLANDS:

- Commercial Business Systems Ltd (Guernsey)

#### YORKSHIRE:

- Holdene Ltd (Leeds)

#### NORTH EAST:

- Currie & Maughan Calculator Services (Northumberland)

#### SCOTLAND:

- Robox (Office Equipment) Ltd (Kirkintilloch)

#### NORTHERN IRELAND:

- Medical & Scientific Computer Services (Lisburn, Co. Antrim)

#### SOUTHERN IRELAND:

- Software Development Services (Dublin)

Kit Spencer's further comments: "We plan to extend this network over the next few months to provide a complete geographical coverage in the UK. Our recently opened Personal Computer Centre at 360, Euston Road, London, will continue to function as Commodore Systems Division's headquarters, accommodating demonstration, applications, sales and service centre".

**PETSOFT — The first independent distributor of software for the Commodore PET**

From early June PETSOFT will be offering a range of games, business programs and related software to users of the Commodore PET.

Priced from £2.50, programs will be available on PET compatible C-12 cassettes. A stamped and self-addressed envelope to PETSOFT at 318 Fulham Road, London SW10 9UG will bring a copy of their first catalogue by return.

Included are 'Alien Attack', a space game with moving graphics, and 'Dr Sinister's Personality Test' which assesses answers to 50 questions before printing out a personality profile, plus a range of mathspacks featuring matrix arithmetic etc. Also of interest to PET users will be the 'Line Renumber' program and an Assembler.

The next bi-monthly catalogue is expected to include a number of programs of American origin, various business programs, plus several sophisticated games such as the Concorde flight simulator currently under development.

PET users are invited to submit programs for inclusion in the library, for which royalties will be paid on all sales.

Initially available from PETSOFT direct, these programs will be offered by PET dealers later in the year.

Contact: William Hamilton & Allen,  
PETSOFT Division, 318 Fulham Road.,  
London, SW10 9UG. Tel: 01-352 1100

**CYTEK (UK) LIMITED**, the Manchester based systems and software house, has been appointed an authorized dealer for the Commodore "PET" Computer.

Over the next few months it will be introducing a range of standard packages to fulfil the needs of small businesses. Also consultancy services will be offered to enable users to realise the full potential of this versatile machine.

Demonstrations of the PET, its peripherals and programs are available without appointment at its Corn Exchange Showroom in Manchester.

Please note the new telephone number 061-832 7604.

**The NewBear Computing Store offers the following:**

**PETITEVID**

Visual Display Unit on a single tested card.

- 64 characters by 16 lines
- Asynchronous V24, RS232C port, Baud rates 110 to 1200
- Upper case only
- Full cursor control
- Composite video output

Requires:

1. Power supply +5v 600mA  
-12v 100mA
2. ASCII Keyboard TTL parallel and strobe
3. Video monitor or Domestiv T.V. and U.H.F. modulator

Cost: around £100.00

**MICROVID**

Complete video display for 8 bit Microcomputers.

- Uses DIRECT MEMORY ACCESS
- 9" inch tube with 64 characters by 16 lines
- Can be used on any 8 bit microcomputer that can be direct memory accessed
- Character set is upper case ASCII

Cost: approximately £180.00 as a manufactured unit.

Contact: Tim Moore at:

NewBear Computing Store  
Bone Lane Newbury Berkshire RG14 5SH  
Telephone: 0635 49223

**32K Roms Available from General Instrument Microelectronics**

General Instrument Microelectronics Ltd announce availability of two new 32K ROMS, the RO-3-9332B and RO-3-9332C. Featuring fully static operation, these two devices have 450 nanosecond and 350 nanosecond access times respectively, making them directly compatible with all popular micro-processor families.

Fully static operation allows users of 16K ROMS such as the Intel 2316E or GIM RO-3-9316B to directly replace two 16K ROMS with one 32K ROM. According to Peter Rush, GIM's Memory Products Marketing Manager, "Most other 32K ROMS are 'edge-triggered' or dynamic, making timing more difficult and often necessitating re-design".

The RO-3-9332 joins a range of G.I.M. N-Channel Static ROMS ranging from 2K bits to 32K bits. This is believed to be

one of the widest such family of mask programmed ROMS currently available. A 64K ROM will be released later this year.

GIM is the leading manufacturer of specialised ROMS for such functions as character generation or keyboard encoding. The company is fully aware that the major feature of ROM user requires is not simply device density or speed, but factory service — enabling the customer to go into full production with his ROMS as quickly as possible. GIM's capability to turn on production is claimed to be second to none, with prototypes in three to four weeks (factory to factory) and volume in six weeks.

"To give the user maximum flexibility, we have geared ourselves to be able to accept customers' ROM specifications in virtually every standard format. Further, our ROM prototypes are truly representative of the production volume ROMS that will be supplied. They are not ROMS of a different process as are promoted by some other vendors, e.g. one time programmable EPROMS. Therefore, GIM's ROM prototypes are usable for full system qualification."

The price for 1,000 pieces of the RO-3-9332B is £7.80/\$14.45 each. Prototype delivery from the factory is three to four weeks after pattern verification by the customer.

General Instrument Microelectronics is a leading supplier of MOS/LSI for the consumer and communications industry. Major plants are located in Scotland, U.S.A., Europe and the Far East.

Further Information: Peter Rush, G.I.M. Ltd.,  
Telephone: 01-439 1891.

**Modem Computer Services Move To The Strand**

Modem Computer Services Ltd., one of the UK's leading DP recruitment advertising, personnel selection and contract programming companies, have moved to larger offices at New South Wales House, Strand, London WC2.

This move is the first phase of an expansion programme which will include the appointment of additional staff to fill several key positions including those of Technical Director, Recruitment Services Manager and Senior Sales Executive.

Les King, Managing Director of Modem, commenting on the company's expansion plans said, "Modem is not just in the business of attracting applicants, although we do this particularly well. In addition to selecting the 'right man for the job', we see it as our duty to positively assist the client company in selling themselves to potential employees. Accordingly, our activities have broadened into the area of public relations and we have already been involved in the organisation of several exhibitions, press conferences and audio-visual presentations.

"Our increasing emphasis on PR will not, however, be to the detriment of our technical standards and we intend to set up a small team of DP professionals to undertake the specific task of candidate assessment and career counselling.

"We never have believed in the 'recruitment consultant' who does 'everything' rather superficially and consider it essential to separate the functions of marketing and selection in order to provide an efficient service.

"Many dozens of companies have found our methods of working far more effective than those of the typical 'paper shuffling' advertising or recruitment agency and the resulting expansion of our business, over the last year or so, has severely stretched our manpower resources.

"We therefore hope that the recruitment of additional staff will enable us to offer our services to a broader range of clients in addition to which we expect to become established as the UK's top DP recruitment company, within the next year".

The Company's full address is now:

Modem Computer Services  
New South Wales House,  
65/72 The Strand,  
London WC2.

Telephone: 01-839 3351

For further information, please contact:

Mr. Les King,  
Managing Director,  
Modem Computer Services Ltd.,  
Telephone: 01-839 3351

**COMPUTER PROGRAMS THAT WORK —**

**Fully tested Mathematics, Science and Games Programs in BASIC**

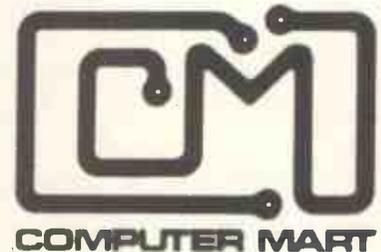
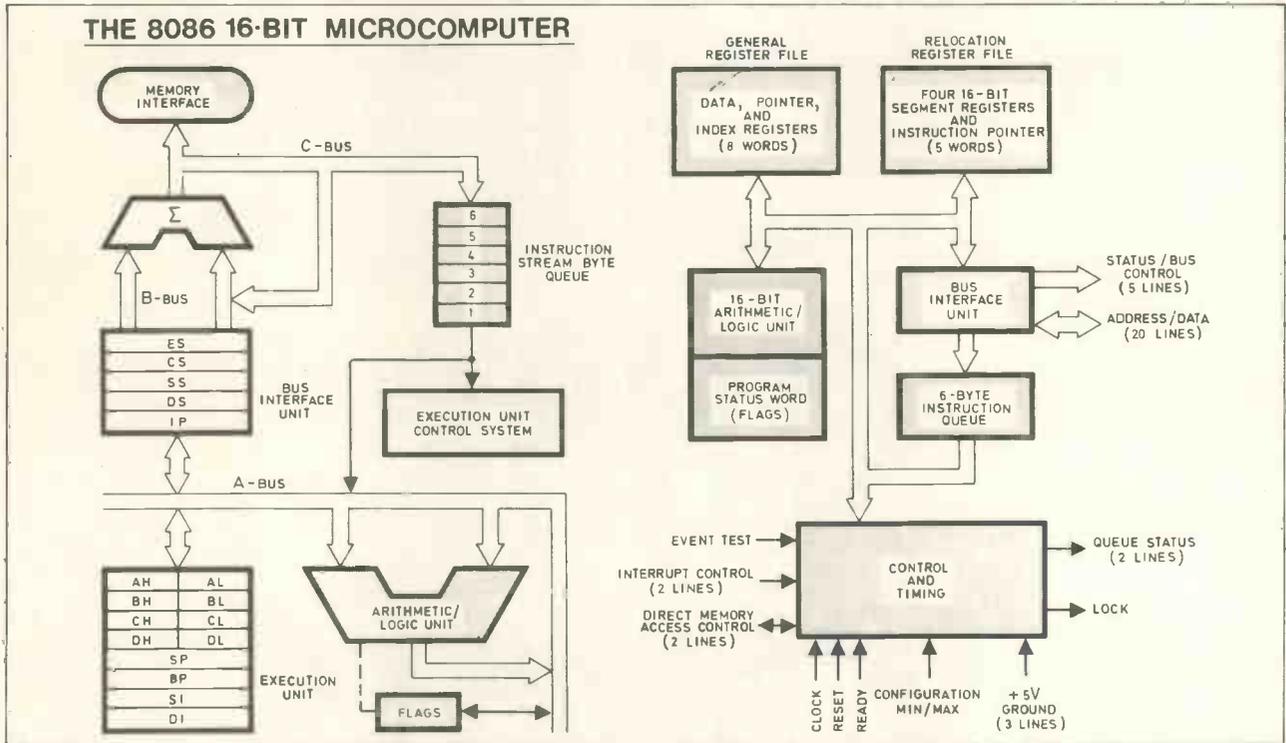
by Dr. J.D. Lee (University of Technology, Loughborough), Dr. G. Beech (The Polytechnic, Wolverhampton — Director of the Physical Sciences Program Exchange) and T.D. Lee (Loughborough).

This is the first all-British book of fully tested computer simulations and games. It is ideally suited to schools, colleges, personal users – in fact, anybody who needs a first class selection of relevant computer programs. The 24 programs described and listed cover biology, mathematics, chemistry and psychics with such topics as:

- Predator-prey ecology simulation
- Diagrams of atomic orbitals
- Satellite motion

Racing car simulation

The programs are so carefully designed that they are almost student-proof. Also, they are written in a standard BASIC dialect so that even users of small computers will be able to make immediate use of the programs. (Purchasers are able to order paper tape copies of the programs). Available from: **Sigma Technical Press, 23 Dippons Mill Close, Tettenhall, Wolverhampton, WV6 8HH.** Price per single copy £2.40



## WE ARE ONLY A SHORT TIME AWAY!

As the jet flies, we may be your closest computer store, and we carry all the major products sold in the United States. When you come to New York (as everyone does, sooner or later) stop in and see our display of computers, peripherals, parts, software and books. Take advantage of our friendly advice, service and low prices.

We sell and service the best products: PROCESSOR TECHNOLOGY, ALPHA MICRO SYSTEMS, SOUTHWEST TECHNICAL PRODUCTS, APPLE II, TDL, CROMENCO, S.D. COMPUTER PRODUCTS, NORTH STAR, REX, MICROMATION, KIM, RCA COSMAC VIP, HAZELTINE, and CENTRONICS.

*STAN VEIT, Storekeeper*

VISA MASTER CHARGE

**COMPUTER MART OF NEW YORK, INC.**

118 Madison Avenue • New York, New York 10016 • 212-686-7923

# IT'S THE THOUGHT THAT COUNTS

W.V. RINGER

In a short space of time we have seen the power of the computer progress exponentially, its availability become universal and its application proposed for a diverse range of problem areas with no boundaries specified. Now it is possible to foresee the day when the unity of science fiction predictions and the extrapolations of scientific will become a reality.

In the study of human psychology, where it has not been appropriate to study man directly, animal behaviour has been studied in order to deduce analogous patterns of behaviour for man. Sometimes only very basic analogies could be obtained in this way and there was always the problem of uncontrollable variables. We now have a remarkable tool, the computer, which is able to hold complex models of inner man and his external environment.

Not only is this tool becoming increasingly valuable for simulating human behaviour, but it is also producing, as a byproduct, a version of the human mind which will start to look more and more like a duplicate as the model is made more complex. The human mind is remarkable for its versatility but is not so good at computing large volumes of data rapidly. The computer is superb at computation but lacks versatility. The question arises: when, or if, the day comes when computers and programs are powerful and complex enough

to be versatile like man, will man be overtaken by the computer? For man to remain dominant, even when the computer has exceeded him in versatility and computational power, he needs a superior advantage over the computer.

The edge that man has is creativity, superior creativity. The computer can create too, but always within limits, the limits inherent in having to draw its life force or motivation from man.

The technology of robotics is like an infant struggling to make its first steps. It seems impossible to predict accurately what kind of adult this child will become in twenty years' time. In order to begin to understand how it might grow it would be useful to consider the composition of man and see to what extent parallels can be drawn with robot-man.

Man is a product of three factors: genetic inheritance, environmental influence and that inner individuality — the soul. It can be seen readily how to draw analogies between man and robot-man as regards genetic inheritance and environmental influence. Whether or not robot-man can achieve the creative spirit of man, is more obscure.

Genetic inheritance is determined by a 'blue-print' of molecular encoding in the DNA spiral. The DNA spiral is present in every living cell and is the unique result of merging two DNA spirals from the original parent cells.

This phenomenally complex double helix of molecules and atoms arranged in a very specific way is simply information condensed into physical form. The information contained in DNA specifies the shape and functions of the body through defining the types of cells and their spatial relationships. The DNA also contains knowledge which is 'implemented' in the body in a 'preprogrammed' or 'hardwired' fashion. We call this body-integrated knowledge, 'instinct'.

The parallels between man and robot-man are very close. It feels natural to explain human characteristics such as 'instinct' in computer terms such as 'pre-programmed', and it is natural to think of the DNA spiral as the equivalent of an engineering drawing and system specification.

A 'bionic' engineer, by using man as a model in the design of robot-man does, in effect, transfer a form of genetic 'substance' from himself to his inanimate offspring. In this way the creation of robot-man and his future generations can begin.

The effects of environmental influence on man can be likened to the results of a self-learning program. It is widely accepted that environmental deprivation has detrimental effects on the development of children. Conversely, a reasonably well structured and lively environment speeds the development of a child and allows it to approach the limits of the facilities with which it was endowed by genetic inheritance. A self-learning program such as a chess-playing program, also thrives on the lively and varied stimulus provided by an experienced opponent. The faster the pace and the more varied the moves, the faster the program will learn.

However, just as a child's running speed is limited by the configuration of its legs, pelvis, lungs and so on, so too the chess program is limited by its design. It can manipulate data concerning pieces and moves and appear to be very cunning, but it cannot operate outside the rules. It is not aware of anything outside the universe of its rules; it is limited by the definition of its creation. If we design into the chess program a *facility to change the rules randomly*, then all we have done is to create a program with more complex 'intelligence'. It can now 'invent' an infinite universe of rules and play games according to those rules and gain a wealth of experience concerning chess games and their variations, but it is totally blind to the universe of draughts . . .!

It seems the computer can never be creative enough to break out of the bounds of its defined universe. It can be superlative within it but never able to go outside it. The universe of the computer is a subset of the universe of man — it cannot be any other way because we are the creators.

The meaning of soul in the context of this discussion is a difficult one, and yet it is very important that we try to understand it. Those who argue that man does not have soul, ascribe all those difficult-to-define qualities to a different category, all of which could be included under 'intelligence'. There is also confusion and disagreement about the subject of computer intelligence. The definition of intelligence in any such discussion is crucial to the way in which the arguments need to be presented in order to gain a clearer understanding.

Computer intelligence can be presented as the ability of the computer, through the execution of programmed actions, to follow one sequence of actions or another depending on certain variable parameters recognised by it. Intelligence in this context is the apparent ability of the computer to make decisions. On close examination, however, it will be found that the element of choice in decision making is simply a pattern-matching exercise in which there can only be one outcome. It is like holding a jig-saw piece and testing it in various positions on the puzzle picture for a fit. In terms of the destiny of that piece in the universe of the puzzle, there is only one

solution, the right position. The word 'choice' conjures up an image of equivalent alternatives, any one of which can be chosen, either at random or depending on subjective feelings, according to correctness, experience, expediency and so on. A computer program is written for a purpose and although at a detailed level there may appear to be a real choice, nevertheless the style of the outcome of the program is predicted.

There is no choice as regards the purpose of a program, because the purpose of a program is also the definition of what the outcome will be. Computer-intelligence, therefore, is really man-intelligence translated into the design of a structured device. This device or tool can be thought of as a frozen distillation of human knowledge. Because the state of this knowledge is frozen in time it can never be more than what it is. It is inanimate and cannot grow, except like a crystal whose ultimate shape is determined inherently by the shapes of its atoms. A program can be written to include evolution, but this is a delusion, because it is another way of saying that the designer of the program visualises the ultimate possible boundaries to which the program will evolve; and, instead of designing the program to those limits immediately, he chooses to stunt the program at a primitive level to start with, allowing it to learn by experience and to modify itself giving the appearance of evolution.

Self-learning program evolution is an important aspect of robotics, it is far easier to 'grow' a highly complex program than to write it in its entirety. To deny the robot life-like 'properties' does not deny the usefulness of imitating life-like 'methods' for achieving the limits of technology.

The principle that applies to man as well as to robot-man is that the creation cannot exceed the level of being of the creator. Man's Creator endowed him with the motivating life-force he calls soul. Man has endowed his physical creation with the distillation of his intelligence in the form of programmed instructions embedded in the hardware and software. The whole is a hierarchical structure in which man is robot-man's god, and just as we cannot presume to know more than the limits of Creation enable us to know, so too robot-man cannot go beyond the creative limits of his creator. The gift to man is life; for robot-man it is the thought that counts.

#### Motorola 6800 single Board Computer Kits.

17 command 1K monitor Mikbug compatible, full 1K user RAM, VDU crystal controlled with own 1K RAM, memory decoder, on card voltage regulators, parallel ASCII input, LS buffered input and outputs, CUTS tape interface. £185.00  
 Mod. 2. Similar to above but with QWERTY keyboard £205.00  
 VDU kit: 1K memory, crystal controlled, parallel input, video output: 1024 characters in a 64 x 16 format. Should suit any processor. £60.00  
 After sales advice service available.  
 All prices exclude VAT and postage. Send SAE for leaflets. Mail order only.

HEWART ELECTRONICS, 95, BLAKELOW ROAD, MACCLESFIELD, CHESHIRE.

#### WANTED

Bright Programmers and Analysts for a small software house based in Central London. We specialise in mini and micro applications. Top Salaries + Profit Sharing! contact:—

Martin Collins, A.S.T. Ltd,  
 2 Duncan Terrace, London N.1.  
 or phone 0442-53549 (evenings)

# GETTING INTO NEIGHBOURHOOD CONSULTANCY

YOU HAVE A COMPUTER  
USE IT !

DAVID FRANCIS

There are two elements in getting enjoyment out your own personal computer. Firstly, there's the building of hardware and software, and, secondly, the practical application of these.

Too often enthusiasts spend time, money and considerable ingenuity constructing computers which never get used to the full and never give their owners as much enjoyment as they got building them and developing software for them.

This article hopes to introduce interesting (and, perhaps, profitable) ways of using your computer to your interest and the benefit of others. In essence, I am advocating that you and your micro get into some basic Operations Research and Systems Analysis. I hope that phrase does not frighten you; O.R. can be as easy or as complex as you wish, but behind it all is applied logic and common sense, and with the minimum of O.R. skills you will find you can be a valuable asset to your 'small businessman' neighbour or relative.

If you are a businessman yourself, then I hope you will apply some of these techniques to your own problems — but beware! — talk your problems over with someone else to give yourself a fresh perspective on them. Very often, problems that have been niggling you for ages turn out to be the symptoms of something else that may or may not be a problem in itself.

My father-in-law opened my eyes to the possibilities. He is a farmer — one of the annoying tribe that takes sheep out for walks along main roads on bank holidays.

## A Hidden Side to Farming

I had always looked on farmers as wellington-booted outdoor types, so could not understand why he should take a pocket-calculator and Philips 'Pocket-Memo' with him when he went to milk. Questioning revealed a hidden side to farming in which budgetting and planning up to two years ahead is the only safeguard the farmer has against finding himself with acres of ripened crops, but no cash to pay the wages for the harvest.

He's in dairying and I was soon swamped in the variables of milk production. Did you know that, for instance, a cow gives a fluctuating milk yield which depends on age, breed, calving season, and past yields?

Amongst other things!

When he wants to know what income he is going to get from milk sales each month over the next year, he's got quite a bit of number-crunching to do. Commercial computer packages are around (including one offered by the Milk Marketing Board) but they cannot match the accuracy of one's own predictions. But, on the other hand, time limited the computations that my father-in-law could do on his pocket calculator. The application of a computer not only saves him time and effort, but allows him far more chance to quantify any ideas he has.

In other words, he can 'model' — he can see what effect high or low farm prices will have on his income and cashflow. This is invaluable because he now knows what he stands to lose if he backs a strategy based on market prices that fail to materialise. If the loss might be crippling he can choose another strategy where the potential pay-off may not be as great, but the potential loss not as great either. Put another way, he can choose to maximise his potential profit, or minimise his potential loss, (or any course in between).

Before looking at this specific application, I will offer some advice on selling yourself, your skills, and your micro to a 'client'.

## Mind his business by minding your own

It is acceptable, in some ways, to go to your client with no technical skills at all and to learn them as you unravel his requirements. As long as you do not waste his time he will probably learn as much as you, and be grateful for the experience. On the other hand, if you go in with superb technical knowledge and start telling *him* how to run *his* business, your relationship will be short-lived.

The best option, however, must be to know your and your micro's capabilities and limitations on techniques and confine yourself to that area. In any case, keep it simple! It is so very tempting to try and make your program cater for every contingency and nuance of the problem, instead of concentrating on a speedy computer solution to the heart of it. I should know; the enthusiasm of creating the total solution is only matched by the feeling of utter helplessness and despair when you find your solution is so 'perfect' and inflexible that it is impracticable and not a solution at all. Your client, who spent time and effort in explaining what he needed and collecting data, will not thank you for presenting him with a jungle of unintelligible numbers.

You *must* avoid this last trap. Initially, you must produce easily understood solutions which can be checked. The public has had so many personal encounters with 'computer mistakes' that his initial reaction to your computer is almost certainly to check the accuracy of your output. So it's up to you to present him with output that is simple enough and laid out well enough to be checked. Which brings me to the second 'must'; check your own output *before* you hand it over. One careless error can destroy all your credibility. If you cannot check for 100% accuracy, at least check that your solutions are reasonable.

There is also a practical reason for keeping it simple: the more complex you make your program, the less flexible will be the solution. Don't be tempted to try and write a program that tells a manager what to do — there are always emotional factors in their decision-making which you cannot program for. My father-in-law, for instance, would not consider farming without dairy cattle — he's as hooked on them as you are on your micro! He would not be interested in knowing that mushroom farming is a far more profitable use of his stock buildings. Give your client a solution (or, better still, a set of alternative solutions) which represents the number-crunching element of the problem — let him modify and build on your solutions from there. Re-

member, help the manager to manage but don't try and make him redundant — he won't thank you for it.

If I sound as though I'm a bit neurotic on this subject, it's because of personal experience. I know that the approach is just as important as the knowledge of O.R. techniques. In this respect, one skill that you must develop is that of talking to your client in a way that encourages him to think about his job and its problems from new perspectives. At the same time, you must try to get to the heart of the matter.

#### Techniques

These skills can be learnt, and you will find an introductory chapter, at least, in most text-books on O.R., Systems Analysis or Work Study. *Your* problem will be to find techniques that do not destroy your natural style but blend into it. Personally, I am conscious of using only three techniques when interviewing:

1. *Redefine the Problem.* To clarify your own thoughts and to force your client to use a new perspective, repeat what he has said using your own words and from a new angle. This can be done simply by saying, 'So you're saying that . . .?'; or you can introduce practical detail by asking, 'What happens if . . .?'. In every case you are asking for confirmation and/or elaboration. Try to define the factors which cause or influence an event or problem. To this end I use the second technique.

2. *Extend the Problem Area.* Talk about what initiates the problem or leads up to it. Then talk about why the problem is a problem; in other words, what it in turn fouls up. If you are not talking about problems as such, but application areas such as budgetting, then talk about where all the information comes from and what all the output is eventually used for. Very often, you can find a short-cut between the two. Once you understand the situation the third technique comes into play.

3. *Bargaining.* No, not for your fee, but to agree on

what you are going to do for him. Most clients ask for the moon in some respects and, at the same time, load themselves and you with unnecessary menial tasks. It is at this point that you must know what your capabilities and limitations are, so that you do not commit yourself to the impossible. Bear in mind what the gist of the problem is, and offer constructive and practical alternatives. By talking through these you both add to your understanding of the problem. But, by and large, commit yourself to far less than you know you can do — firstly, because you will be over-optimistic to start with and, secondly, because the stimulation of talking over his problem often leads your client to redefine it, or even solve it over the following week. So keep in frequent contact with him and you won't find you've sweated blood on a non-existent problem.

The 'Expectation Gap' is the jargon for this last pitfall, and it aptly expresses how mutual agreements do tend to diverge as time passes. The movement is usually on both sides — so go back to your client frequently and just check that you have both still got the same objective. It is embarrassing to present a solution to a problem as you perceived it, when your client can't see any problem there to solve. Watch out for the 'Expectation Gap' — it attracts enthusiasts like flies!

#### Keeping a Confidence

Lastly, I must mention confidentiality. If you are tempted to show how successful you have been, by exposing a previous client's inefficiencies and problems to your new client, don't be surprised if he gets cold feet and keeps his problems to himself. Have respect for your clients and learn to 'forget' confidential figures.

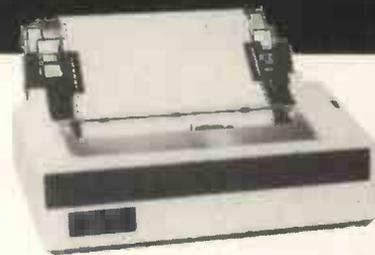
This article has introduced you to the 'personal' side of neighbourhood consultancy work. If you like the idea, have a try at it — you might end up another McKinsey & Co.

## COMPUTER PERIPHERAL EQUIPMENT FROM CROFTON



For full details send for our  
Data Sheet ref: MC1

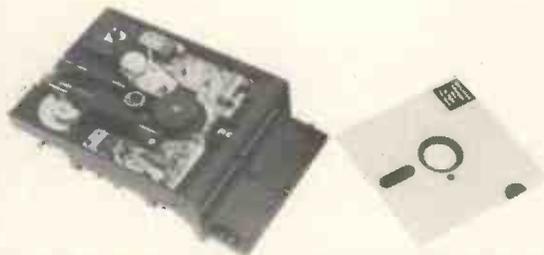
S.A.E. for return of post service



*TO BE RELEASED NEXT MONTH... OUR OWN IN HOUSE*

**DESIGNED 6800 Micro Processor**

See Back Page of P.C.W. September Issue for Information



**CROFTON ELECTRONICS LIMITED**  
35 Grosvenor Road, Twickenham, Middx. · 01-891 1923

# THE R.O.M.\* FACTOR

J. GREGORY

\*REQUIRES  
ONLY  
MONEY

Bernard had been driven by it for some time.

The first symptoms would be familiar to most people reading this short tale.

Had it started with a short article somewhere?

A glimpse on television?

A taste of the actual thing live at a demo?

Talking to freinds perhaps, or a dozen other more subtle and pervading ways?

Time now obscured the origin, but there was no

doubt that Bernard was addicted.

Soon he had been on the streets asking at the little 'shop' that had supplied him for years with his more common needs. He had even enquired at the new, anti-septic, bubble-packed pushers with American connections.

He just had to get it! Somewhere! Somehow! Soon!

He had paid high prices for information hidden inside expensive covers, passed on by people who had said they wanted to help. Users had said that they knew where he could lay his hands on some really good stuff. He knew he could but at a price . . . a price he could not afford.

His room was littered with the soft stuff, he had exhausted every ounce he could from it. He had even consulted Dr. Dobbs' in a futile attempt to import something he could afford, something that could just start that mind-expanding experience that Bernard knew would bring him deep satisfaction.

All avenues seemed closed.

The internal torment he had known for months was becoming more evident to his friends and colleagues. He had joked with them, admitted in what he thought was a casual way, his desire to get hold of some of the hard core stuff, Just to see what it was like". Some had been sympathetic, but he recognised the responses - They thought he was strange.

Now Bernard sat . . .

Bernard sat thinking . . .

Bernard could have been anywhere . . .

Bernard's mind was where Bernard was. That wonderful internal universe, where all things are possible, any numbers of desires attainable, but sadly intangible, untouchable.

It was no use just thinking! Action was required . . . but what action?

He sat, eyes closed - mind open. The sounds of Mike Oldfield acting as an external stimulus to his tangental thoughts. Occasionally his unstructured mental playtime would be interrupted by that internal Devil's advocate that presents us all with 'dummy arguments' to act upon.

Haven't you anything more worthwhile to do? It demanded.

Would you really use it?

Is it really what you want?

Is it something else that you are searching for?

Is this an end in itself or a means to an end? It continued.

How will it affect your family, your friends, the dog, the goldfish? Oh! The money! . . . The time! . . . The effects! . . . The side effects! . . . Would it be the start of a long road without end? Would he always need more and never quite get it together?

The questions were nebulous, the answers infinite.

Bernard half resolved to discuss it with someone else - later, perhaps tomorrow . . .

A change in the music rhythm prompted a new train of thought. It was time to consider a definite way of getting what he wanted . . .

He had to have a plan.

One word kept returning to loom large. One word that is often little more than just a word, and in Bernard's case was definitely only a word . . . Money.

He put aside the recurring proposition that, if he had the money, there would be many other claims on it. Desire is an obscuring drive and Bernard's desire was about to oust that magic mysterious force for balance we know as 'common sense'. His attempts to justify what he proposed, just did not stand up. He knew that this was not going to improve his sexual prowess. He knew that this would not really enable him to find new ways of running his life, his home, and his bank manager with greater efficiency.

It was best to face up to it, accept that he had a burning desire and that it had to be fulfilled . . . Quickly. 'Common Sense' had been silenced.

So, there remained, as in the beginning, one obstacle, that final hurdle that separates Madmen from Eccentrics . . . Money.

His associative reasoning was already pouring out everything and anything pigeon-holed under 'Money' in the hope that by the wonders of thought a key could be found to the problem.

- WORK — already doing all possible . . . pass on.
- BORROW — already doing all possible . . . pass on.
- BEG — already doing all possible . . . pass on.
- STEAL — ILLEGAL — stop.

Had it really come to this?

In the words of the song (also poured from the pigeon hole), Bernard was poor, but he was honest.

He paused.

'The only difference between men and boys is the cost of their toys' arrived in his thoughts for consideration.

Is it just a game? He asked himself, as his now open eyes scanned the assorted furnishings, the still active Hi-Fi system, and the children's abandoned toys.

His mental advocate reappeared, for a moment. "Look what happened to the 10 option colour T.V. game, the one you built, played for a while, and which now gathers dust!". The stylus was reproducing only the steady heart-like beat of the centre groove, as Bernard stood to turn the record over. The soothing introduction to the second side accompanied his desire's reply to temporary sanity.

"The T.V. Game had been completely different — not at all the same — just a poor form of escapism."

Bernard returned to his chair and again closed his eyes. What he wanted now would really be the best of all the interests, hobbies and pastimes that he had flirted with, pursued and performed. The music tempo increased, the beat hardened. Guttural sounds and controlled cries of anguish and ecstasy filled the room, matching the build up of Bernard's mental resolve.

This would be the climax to all his activities.

BUT . . . the cash, the where-with-all, the means to this end. Where was it to come from?

Again his mind entered the loop that contained the known sources of financial gain.

- EXTRA JOB — already doing all possible . . . pass on.
- WIFE TO WORK — already doing all possible . . . pass on.

SELL SOMETHING — what value a 10 option T.V. game? . . . pass on.

INHERITANCE — no one to pass on . . . pass on. Bernard's eyes opened wide . . . . .

In that flash, that inexplicable human brain blowing experience that sends a tingle of something to every part of the system. That wonderful marvellous gift to all mankind . . .

That great internal feeling of 'perhaps' that rushes through a rough internal test of validity . . . emerges as a 'could be' . . . then races to the final probability checks . . . passes with a clearance of 'feasible' and finally explodes from the vocal chords as a clear "I've got it."

Or, if one is given to plagiarism: "EUREKA".

Bernard had a plan . . . . .

Never mind by what divine inspiration, (and at this stage Bernard was not ruling that out), the solution had come to him, Bernard had a plan . . . simple . . .

. . . He would win what he wanted!

The reader may have realised that the subject of Bernards fancism is not the sort of item offered by the average competition. No ordinary source was going to offer what Bernard wanted, as bait for the old. "Use your skill and judgement to arrange The Grenadier Guards in order of value to a Salford Housewife."

If Bernard was to win his release to a heaven on earth, he would have to turn to that twilight world of committed underground revolutionaries. That little known band of individuals, whose aim in life is to entice unsuspecting students, graduates and deviates from normality, into a life of half human, half machine existence.

He had one of their insidious publications that he had acquired 'under the counter' on special order.

He recalled that it offered a package of the goods he wanted, in return for what seemed like his soul.

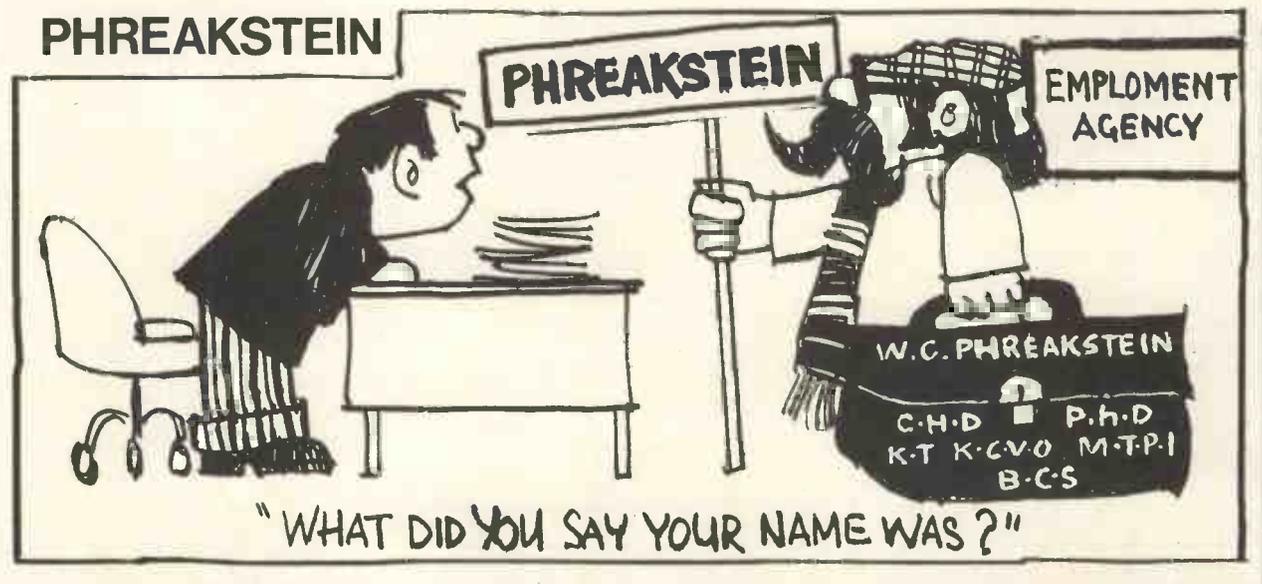
He took a breath, a long deep breath, there could be no other way now.

He looked at his watch, it was late, but not too late. Yes . . . he would do it now.

The speakers erupted into a triumphant sailor's hornpipe that reflected Bernards enthusiasm for the resolution he had made.

His plan would not fail. How could it . . . .

(PCW Poor Bernard. He didn't win the competition and can now be seen on the streets of Manchester with a begging bowl into which people casually throw defunct integrated circuits).



# L.P. ENTERPRISES

Room 0.21  
313 Kingston Road, Ilford,  
Essex, IG1 1PJ, England  
01-553 1001

From the representatives in Europe . . . for America's leading Micro-computer magazines and books, for the hobbyist, educationist and professional alike, we bring you a little light browsing!  
Reading maketh a full man . . . Francis Bacon (1561-1626)

Tick or indicate quantity ordered

<b>From Adam Osborne Associates</b>	
<b>INTRODUCTION TO MICROCOMPUTERS</b>	
Volume 0: The Beginners Book	£5.95
Volume 1: Basic Concepts	£5.95
Volume 2: Some Real Products (Revised Late 1977)	£11.95
6800 Programming for Logic Design	£5.95
8080 Programming for Logic Design	£5.95
Z80 Programming for Logic Design	£5.95
8080 A/8085 Assembly Language Programming	£6.95
6800 Assembler Language Programming	£6.95
Some Common BASIC Programs	£5.95
<b>BUSINESS PROGRAMS IN BASIC</b>	
Payroll with Costing Accounting	£9.95
Accounts Payable & Accounts Receivable (Available from Midsummer 78)	£9.95
General Ledger (Available from late summer 1978)	£9.95
<b>From Scelbi Computer Consulting Inc.</b>	
6800 Software Gourmet Guide & Cookbook	£7.95
8080 Software Gourmet Guide & Cookbook	£7.95
8080 Programmers Pocket Guide	£2.25
8080 Hex Code Card	£2.25
8080 Octal Code Card	£2.25
8080 Guide and One 8080 Code Card	£4.20
8080 Guide and Both Code Cards	£6.00
SCELBAL High Level Language for '8008/8080' Systems	£39.25
SCELBAL String Handling Supplement	£8.00
SCELBAL Extended Maths Supplement	£4.00
Understanding Microcomputers & Small Computer Systems	£7.95
SCELBI 'BYTE' Primer	£9.95
8080 Standard Assembler (In Book Format)	£15.95
8080 Standard Editor (In Book Format)	£9.95
<b>From Peoples Computer Company</b>	
Reference Books of Personal & Home Computing	£4.95
What to Do After You Hit Return	£7.00
Or. Dobbs Journal Volume 1	£10.00
<b>From Kilobaud/73 Magazine Inc.</b>	
Hobby Computers Are Here	£3.95
New Hobby Computers	£3.95
<b>From Oymax Inc.</b>	
Instant BASIC by Jerald R. Brown	£4.95
Your Home Computer by James White	£4.95
My Computer Like Me . . . When I Speak	
BASIC by Bob Albrecht	£1.65
Games with a Pocket Calculator by Thiagarajan & Stilovitch	£1.75
Games, Tricks and Puzzles for a Hand Calculator by W.Judd	£2.49
<b>From BYTE Publications Inc.</b>	
Paperbytes:	
Tiny Assembler for 6800 Systems	£5.75
Bar Code Loader for 6800, 8080, Z80 & 6502 Micros	£1.75
Best of BYTE Volume 1	£8.95

Tick or indicate quantity ordered.

	Price UK	Price Overseas If Different
<b>From Creative Computing Press</b>		
Best of Creative Computing Volume 1	£6.95	
Best of Creative Computing Volume 2	£6.95	
101 BASIC Games (Revised & Reprinted Feb.78)	£5.50	
The Colossal Computer Cartoon Book	£3.95	
Computer-Rage (A new Board Game)	£6.95	
Artist and Computer	£3.95	
<b>From Everyone Else</b>		
Magazine Storage boxes (holds 12 minimum)	£1.75	
Sybx: Microprocessors from Chips to Systems by R.Zacs	£7.95	
Sybx: Microprocessors Interfacing Techniques by R.Zacs	£7.95	
Dilithium: Home Computers Volume 1: Hardware	£6.50	
Dilithium: Home Computers Volume 2: Software	£5.95	
Getting involved with your Own Computer	£4.75	
TV Typewriter Cookbook by Don Lancaster	£7.50	
TTL Cookbook	£7.50	
CMOS Cookbook	£7.95	
IC Timer Cookbook	£7.50	
IC OP-AMP Cookbook	£9.50	
RTL Cookbook	£4.25	
Computer Programs that work (In BASIC)	£2.55	
<b>From Basic Software Library (from Scientific Research Instruments)</b>		
Vol 1: Business and Personal Booking Programs	£17.50	
Vol 2: Maths and Engineering Programs	£17.50	
Vol 3: Advanced Business Programs	£26.95	
Vol 4: General Purpose Programs	£7.95	
Vol 5: Experimenters Programs (General Purpose)	£7.95	
Vol 6: General Ledger Program	£32.50	
Vol 7: Professional Programs	£26.95	
<b>Magazines: Back Issues</b>		
Personal Computing	£1.75	
Interface Age	£2.25	
Dr. Dobbs Journal	£1.75	
Computer Music Journal	£2.50	
Peoples Computers	£1.75	
*BYTE	£2.25	
Creative Computing	£1.75	
Calculators & Computers	£1.75	
ROM	£1.75	
Kilobaud	£2.25	
73	£2.25	
<b>MAGAZINES: Subscriptions</b>		
Personal Computing (Twelve Issues Yearly)	£16.00	£17.00
Interface Age (Twelve Issues Yearly)	£20.00	£20.50
Dr. Dobbs Journal (Ten Issues Yearly)	£13.00	£13.50
Computer Music Journal (Four Issues Yearly)	£8.50	£9.00
Peoples Computers (Six Issues Yearly)	£8.00	£8.50
Kilobaud (Twelve Issues Yearly)	£20.00	£21.00
*BYTE (Twelve Issues Yearly) via USA	£15.00	
BYTE (12 issues yearly) via U.K.	£21.00	
Creative Computing (Six Issues Yearly)	£8.50	£9.00
Calculators & Computers (Seven Issues Yearly)	£10.00	£10.50
73 (Twelve Issues Yearly)	£20.00	£21.00

THIS LIST CANCELS ALL PREVIOUS PRICE LISTS: EFFECTIVE MAY 1978

578/VC

**HOW TO ORDER**

Please note our prices include postage and packing, but not insurance, if wanted add 12p for every £10. of books ordered. Make cheques, PO's etc. payable to:-

L.P. Enterprises.  
CREDIT CARDS accepted  
BARCLAYCARD VISA / ACCESS  
DINERS CLUB / AMERICAN EXPRESS

Phone: 01-553 1001 for Credit Card orders (24-hr service)

Send to address above	All Orders must be Prepaid
Indicate Payment Method:	Total Enclosed £ . . . . .
..... My cheque, P.O., I.M.O. is enclosed in Sterling on U.K. Bank	
..... Charge to Barclaycard/Visa/Access/Diners/American Express	
Credit Card No . . . . .	Expiry Date . . . . .
Name . . . . .	
Address . . . . .	
	POSTCODE . . . . .
Signature . . . . .	

All publications are published in U.S.A. and shipped air-freight by L.P. Enterprises. In unusual cases, processing may exceed 30 days.  
\*BYTE subscriptions are processed in U.S.A. and are air-freighted & posted from Amsterdam and will take 3 months to start.  
U.K. Subscriptions start within 3 weeks.

TRADE ENQUIRIES WELCOME

# SHOW NEWS

PERSONAL COMPUTER WORLD SHOW  
21-23 September 1978

Show News . . . Show News . . . Show News . . . Show

WILL MARTIN,  
Exhibition Manager  
Phone: 01-486 1951

## Applications for Space Pour in

Within a week of publishing the exhibition brochure more than a quarter of the available space in the Personal Computer World Show has been booked. It is likely that at the time of reading more than half of the available stands will have been taken up.

The list of exhibitors at the time of going to press includes the following names:

Comart	Technalogsics
Pelco	Computer Workshop SWT
Micronics	Sirton Products
Crofton Electronics	Cytek
Nascom Microcomputers	Tandy Corporation
Datac	Comelec Electronics
Personal Computers	Newbear Computing
Bywood Electronics	

There are at least twenty more companies who have already expressed a keen interest in the Show and who will soon be joining the list.

## New Products

The Personal Computer World Show will be the best place to unveil new products in 1978 and we already have news of three companies who plan to do just that — visitors to the Show should make sure not to miss the stands of Tandy, Nascom Microcomputers and Bywood Electronics . . . who will use the *PCW* Show as the occasion for launching exciting new products.

## Applications for Space

Firms who have somehow missed out on a copy of the exhibition brochure should ring me, Will Martin, on 01-486 1951 right away.

## Conference

Although the programme cannot be announced in full at the time of going to press we can tell you that we are all set for an exciting conference and a smash-hit exhibition.

As we pointed out in our advertisement in the July issue of *PCW* we have divided the conference into three seminars — one on each day of the Show covering the following subjects (please note that we have reversed the order of the first two days from the original announcement — sorry, our mistake!)

Thursday 21 September	— Small Computers and Small Businesses
Friday 22 September	— Small Computers in Schools and Universities.
Saturday 23 September	— Computing for the Beginner and Hobbyist

If you've been to conferences before you'll recall those minor irritations which have added up to discomfort. Well, so far as we are able, we'll see you come first. Just a couple of examples: There'll be people on hand to guide you and inform you; there'll be a moving microphone in the conference hall so that your questions are heard clearly by everybody.

Speakers to date include:

Business computers — John Burnett (Computer Work-

shop), Laurence Payne (Comutech), Richard Waller (CAP Microsoft), David Hebditch (*PCW*).

Education — Professor G.D. Dawson (University College, London), Julia Howlett (National Physical Laboratories), Colin Wells (Downs School, Kent)

Beginners and Hobbyists — Mike Dennis (*PCW*), John Coll (*PCW*), David Goadby, John Miller-Kirkpatrick (Bywood Electronics), Sheridan Williams

## Special Features

As mentioned in our advertisement in July there will be particular attractions for the hobbyist and home computer enthusiast in the Show's feature area. *Our Innovators' Corner will give the homebrew fans the chance of showing, free of charge, the systems they have been developing* in back rooms and garages. If you would like to display what you have achieved please send us a short description of your system giving what you think are its outstanding features, possible commercial usefulness and if possible a photograph.

The feature area will include video games, The *PCW* Microchess Championship, famous personalities etc.

## Prizes

We hope to persuade exhibiting companies to donate prizes for the best homebrew systems and the ball has already been set rolling by Newbear Computing Store who have kindly donated a Petitevid terminal as a prize. In addition there will be cash prizes of £200 each for:

1. Best Software
2. Best Homebrew System
3. Best School Application
4. Best Home Application

In addition, we shall be presenting the *PCW* Trophy in each category. And, don't forget: **The Microchess Championship!** Enter now by writing to *PCW* for details.

## Visiting the Show

A special information leaflet for visitors to the Show giving details of exhibits, features, conference seminars, transport, accommodation etc., will be ready soon. This will be mailed to all of you who wrote to us for visitor and conference information and also widely to business and educational interests.

## Tickets

Admission to the exhibition will be £1.00 including VAT. Conference delegates will get automatic admission to the exhibition on all three days. Those of you who have visited exhibitions before will know at once that *PCW* is really keeping prices low to the public. It's what we believe in.

If you are planning a visit to the Show and want to save time at the entrance and be sure of your ticket please complete and return the form with your cheque/P.O. to **Interbuild Exhibitions Ltd, 11 Manchester Square, London W1M 5AB**. Special rates apply for bookings of 20 or more tickets.

Personal Computer World Show 21-23 September 1978  
Please send me . . . Admission tickets at 80p each including VAT.

I enclose my cheque/P.O. for £ . . . .

Name . . . . .

Address . . . . .

. . . . .  
. . . . .  
. . . . .  
. . . . .

Please make cheques payable to:  
Interbuild Exhibitions Ltd.

VAT no. 232 7647 60

# From Ideas to Programmes

## Ready to go

You've got your home computer system together and your head is fairly buzzing with ideas but, oh dear, how to translate them into programs?

Probably your computer hardware and ideas are somewhat individualistic and canned programs make interesting reading but what you really want to do is make up your own.

If you read plenty of programs written by other people you will gradually assimilate the many hidden assumptions, stratagems, tricks, short-cuts etc. which are in most complex, efficient programs. This, however, can be a tedious process, especially when you feel you will learn faster and more fully by writing your own.

What you need is information on some of the techniques used in constructing good programs.

## Don't run before you can walk

So that you can become familiar with the instruction set of your own particular microprocessor, start by writing around a dozen "mickey mouse" programs.

Possible candidates are:—

- (1) Flashing LEDs on and off at a rate determined by switches. (Read the switches into a register and count down or up till a carry occurs, for timing purposes. Think up ways you could extend the count to increase the flash on and off times).
- (2) Have the LEDs counting up in binary fashion at a given rate.
- (3) Add binary numbers from the switches and display the total on the LEDs.
- (4) Fill up all RAM with numbers (try the same number, sequentially increasing or decreasing numbers and so on) and then sequentially fetch and display them.
- (5) Shift the contents of a block of locations in RAM to other locations. etc.

You should begin very simply; programs containing perhaps half a dozen instructions and gradually building up to more complicated arrangements as you gain confidence. All this is extremely valuable and will enable you to grasp some of the possibilities and limitations involved in programming with your microprocessor (and/or the high level language you may be using). In particular, program looping, addressing techniques, branching instructions and data manipulation methods can be tried out until their use is understood.

Don't chuck these programs, when you've finished with them, in the bucket. Variations of them can be useful later on for simple diagnostic checking of correct hardware functioning.

## Firm up your ideas

Are there any common threads in the applications you intend for your system? This is a very important question and deserves a lot of thought. If there are, you can save yourself time and effort.

Let's say your interest is video games. If every time you have a new idea for a game you write the entire program from scratch, the effort will gradually remove the pleasure, especially as your games become more sophisticated. Most of your game programs will need to construct obstacles, boundaries etc. on the screen. So write a subroutine which does this from a given list of variables (position co-ordinates, shapes or whatever). You need then only change your variables for each different game for use by this subroutine; the remainder of each new program can then be tackled separately. This procedure of writing specialised subroutines instead

of specialised programs saves wasteful duplication and tends to tidy up your methods of creating programs.

The benefits of flowcharts in encouraging clear thinking during the preparation of these subroutines and programmes cannot be stressed too much.

## Who needs a monitor?

You've written your program and it doesn't work. A few minutes careful thought at this stage can prevent fruitless searching in the wrong place for the bug.

Firstly, is the problem in the hardware, the software or has the program been entered wrongly? It is all too easy to blame the hardware, so, if it has had a history of reasonable reliability, only start checking wires, voltages etc. if all else fails. Unless the program is very short, sequential checking of all the instructions in the program memory is definitely not recommended either. So, if there are no clear indications otherwise, the best thing to do is to consider the problem to be in the software.

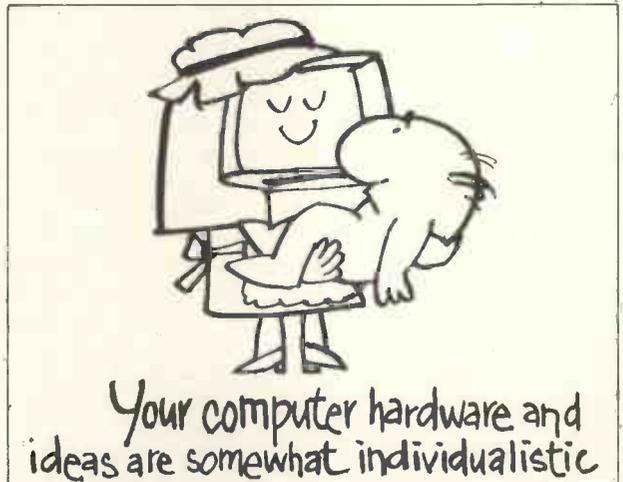
Now think through your program, with its faulty actions when run as clues, to see if there are any obvious bloomers.

No joy? If you have a monitor facility, now is the time to step through your program to determine what is happening.

If your system is very simple and lacks a monitor or even a single step facility, don't despair! It is perfectly possible to debug your program, with a little ingenuity. In fact, even if your system is well endowed the following points are worth noting.

- (a) Liberally sprinkle your programs with no-operation instructions (except in timing loops or time critical routines). This can make it easier to insert the odd extra instruction as required. In addition you can replace a no-op which is in the faulty area of the program by, eg, an instruction to display some register on LEDs and thus see what's going on there.
- (b) Liberally sprinkle your programs with comments so that in six months' time (or even the next day) you'll still be able to understand how it's meant to work.
- (c) Don't be afraid to completely rewrite the program if, for the life of you, you can't figure what's going wrong. You'll probably discover a better method of achieving the same result and the faulty reasoning in the original will undoubtedly dawn on you one morning, weeks hence, when you're shaving!

Remember, you're supposed to be smarter than your microprocessor so you should win in the end.



# A COMPUTER OR A KIT ?

## THE INCOMPLETE KIT BUILDER

MICHAEL JAMES

There's nothing new about electronic construction kits, they've been around for some time. As a way of learning about electronics they're excellent but what about as a way of owning a computer? The difference between the two is: the interest of the builder. A growing number of people are looking toward cheap computer kits as simply a way of owning their home computer. The interest lies in the end product — the computer and its programs — rather than the electronics involved in building it. This means that a group of people who would never even consider building a £5 transistor radio are now thinking of paying at least £150 for one of the most sophisticated pieces of electronic equipment available — in the form of a kit!

For the benefit of the computer (and some electronics) hobbyists let us look at the pros and cons of kit building, at some of the problems, and the few pleasures. Taken at face value, the task of building a kit is nothing more than following the instructions. If you do everything exactly as specified nothing can go wrong . . . or can it? No matter how well thought out the instructions may be, they still have to be interpreted. We have to assume that they were written by an electronics expert for the uninitiated, and what is common knowledge for an electronics expert may be obscure to the layman. I never fail to marvel at how, after being told the difference between two sorts of plugs (fairly standard and obvious), I am left to sort out the difference between mylar and polystyrene capacitors.

This brings us to our first major problem — identifying the components. You must not only be able to identify the more obvious resistors, capacitors, transistors, etc., but their more exotic sub-species, e.g. wire-wound resistors, tantalum capacitors, etc. The solution to this problem is to obtain a component identification sheet (sometimes, but very rarely, given away with the kit). This is a good idea even for the experienced constructor, for even though I can tell a tantalum capacitor from the rest, identifying which lead is positive is still sometimes a problem. Where to get a component identification chart is something of a problem in itself. They're sometimes given away with electronics magazines and turn up in data books. (A few catalogues help as well).

There is another problem about identifying components — even if you can tell a resistor from a capacitor you still need to know its value. For resistors this is relatively easy and means knowing the colour coding. Again a data book helps. But for capacitors this is a matter of inspired guess work! Even after seeing quite a lot of capacitors there is still a residue that I'm never 100% sure about. Some have the value written clearly, some use a colour code, and the rest need personal knowledge.

For example, one make of capacitors marked 10n means .010nF because you need to know that there is an implied decimal point and zero to the left. And if you don't know what the n stands for there's even more trouble in store for you. Most parts lists give capacitors in  $\mu\text{F}$  (micro Farads) or pF (pico Farads), ( $1\mu\text{F} = 1,000,000\text{pF}$ ), but a lot of capacitors are marked in nF (nano Farads), ( $\mu\text{F} = 1,000\text{nF}$ ), and if you don't get your conversion right then you're in trouble. It is true that the identification problem is eased a little by knowing how many of each part you're supposed to have. I say *supposed* because parts can be missing and you can get into an awful mess because of an early mistake.

Related to the problem of identifying the parts is the orientation problem. Some components have to be soldered in the right way round — IC's, diodes, transistors, electrolytic capacitors, etc. Now, although the instruction writer usually thinks of explaining all this, two things can go wrong: 1) the drawing of the component or, more often, its own markings might be unclear, and 2) a completely different component might be supplied in a later issue of the kit. The latter happened to me just recently when the picture of the diode I was trying to install showed a black band at one end — alas, the one I had been given had no band, just one end smaller than the other and no clue as to which way round it should go.

We have come this far without mentioning the most important problem encountered by a first-time kit builder — soldering. It's true to say that this, in one way or another, is the biggest cause of malfunctioning kits. Even if you think you know how to solder, a computer kit will be the most difficult and expensive thing you have tackled. If you don't know how to solder, then my advice is *do not attempt a computer kit until you do*. A good way to learn how to solder is to buy a cheap electronics kit and try to make it work. Good soldering is not difficult but it does need patience and practice. Bad soldering techniques can damage components through overheating, but even if this is avoided we still have another enemy — static.

Unlike most other electronic kits a computer will contain a lot of MOS chips. These are sensitive to static electricity. The first time I saw someone working on MOS in the days when it was a new invention, I thought it was an unsubtle attempt at suicide. Sitting at an aluminium-covered workbench with metal wrist bands connected to earth (hopefully), and no shoes and socks, is not my idea of fun! Personally I think this is taking things a bit too far. A simple piece of cooking foil and a well earthed insulated soldering iron are usually enough — unless you have a nylon carpet or are particularly prone to attacks of static!

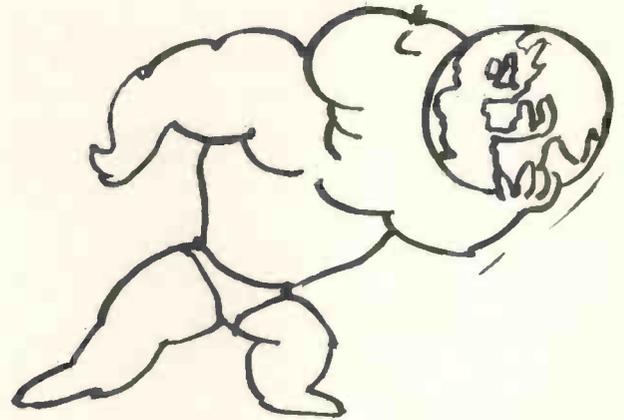
Supposing that we've got over all the problems — identified all the components correctly, soldered them into place with due care for heat and static — what are the odds that the complete unit will work first time? Nearly everyone expects their handiwork to function first time, but I reckon a 50% success rate is more realistic! How long to put it all right? Well, that depends on several factors, mainly how much help you can get, but one month is not unusual.

If you are interested in electronics or how a computer works then kit building is for you. After building and making your computer work, you will *really* be a computer expert. If however you are interested in computers at the software level, then unless you are prepared to do a lot of learning about electronics techniques, suffer many disappointments when it doesn't work, and are prepared to wait while missing or replacement parts are obtained, then a computer kit is not for you. It simply isn't worth the money you save in terms of the time it takes. But if you do decide to build your own . . . Ah the satisfaction when it does work!

# A MIGHTY MICROMITE

## THE KIM-1 EVALUATION UNIT

C. CORBETT



### The Board

The KIM-1 is a ready built microcomputer board which hails from the same stable as the PET home computer. In terms of facilities available to the user, KIM must be placed at a level lower than PET — and can be compared with similar products including the Motorola D2 (which uses an M6800 CPU) and NASCOM-1 (which uses a Z-80). Like these other evaluation kits, KIM is programmed by machine language, thus requiring a *detailed* knowledge of the microprocessor instruction set and characteristics. To relay some of this information, this article presents the major features of KIM that are important from the hobbyist's point of view. Incidentally — the microprocessor description will be relevant to PET users too, since the MCS6502 is the micro used in both.

KIM is a complete microcomputer circuit on a single circuit board requiring no extra soldering. Accompanying the board is a comprehensive set of manuals<sup>1 2 3</sup> a system connection chart, and a programming reference card. The User Manual describes the computer and user interaction via an on-board monitor. The hardware Manual describes the characteristics of the LSI chips and the peripheral interface circuits used by KIM. Finally, the Software Manual presents the full instruction set and contains an introduction to programming techniques in general.

### Extras

Vital extras required before KIM can be used are a 5 volt 1.2 amp power supply for normal operation, an extra 12 volt supply for the magnetic tape interface, and a cassette recorder. Although the 12 volt supply is not used by the microprocessor, the ability to store and retrieve programs from, say, a cassette recorder, is vital with this type of unit and so should be used. A circuit for a suitable supply is given in the User Manual, which can be bought separately.

### Keyboard

To interact with KIM once power is applied, use is made of a keyboard and six-digit hexadecimal display on the

circuit board. A 2K-byte monitor program in ROM allows the examination and alteration of store locations (with the displays showing address and contents information), the controlled execution of user programs and the storage and retrieval of programs via magnetic or paper tape. If the KIM owner happens to own a serial-line device (teletype or VDU), it can be used as an alternative to the keyboard and display for program entry and correction.

The keyboard arrangement is illustrated in Figure 1 below.

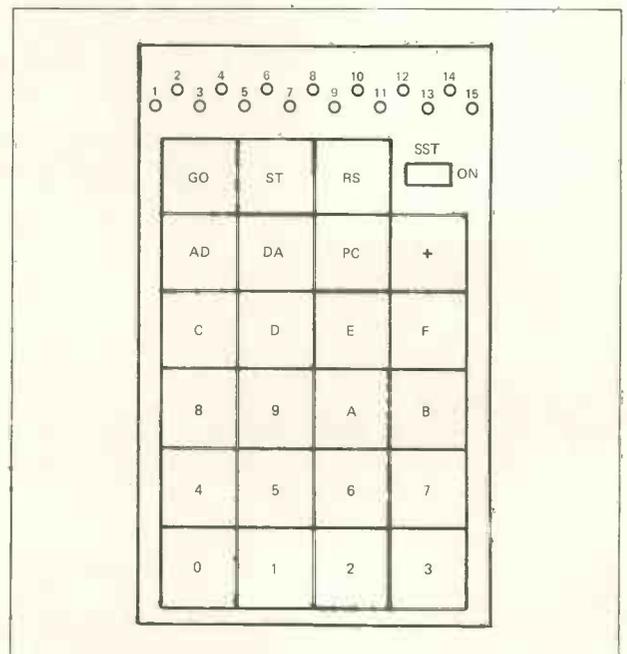
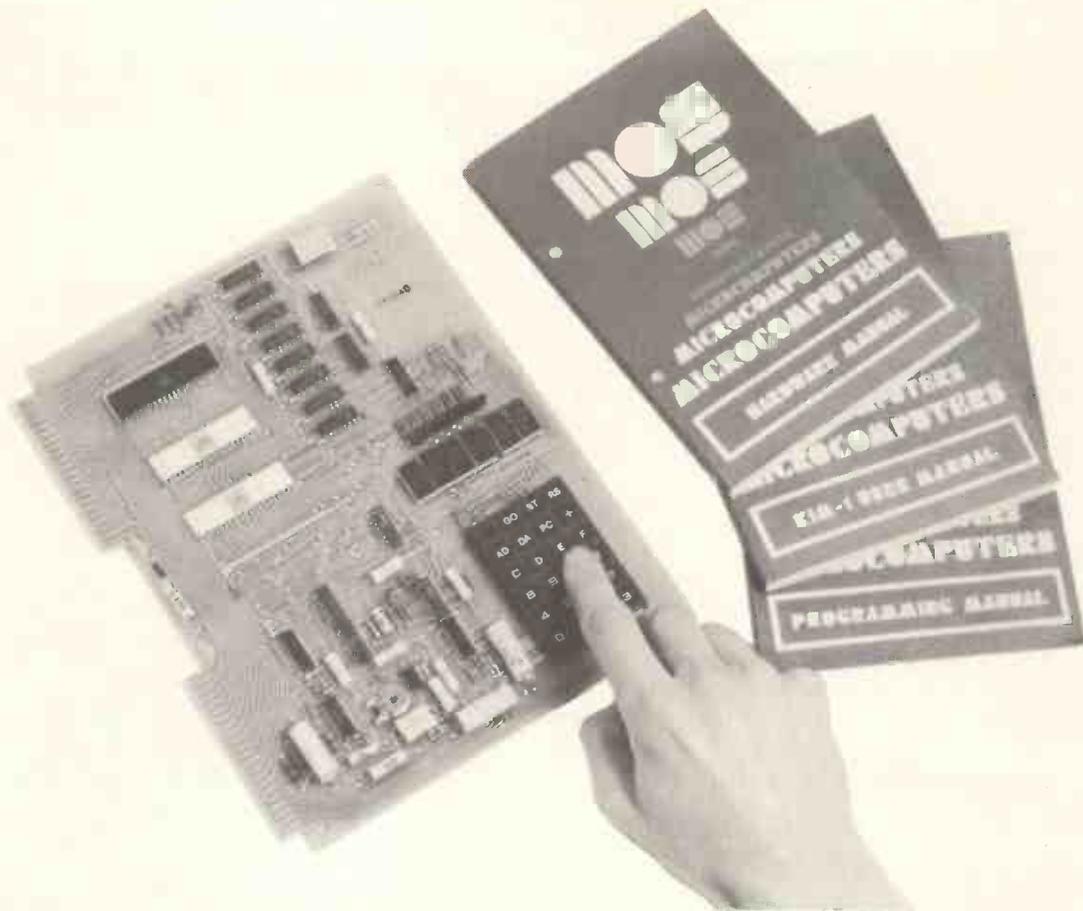


Figure 1 KIM keyboard layout

A block of sixteen keys labelled 0 to F allow entry of hexadecimal information which is interpreted according to the function selected (address or data input mode).



A brief summary of the other keys and their functions follows:

AD: set address input mode.  
 DA: set data input mode.  
 + : increment address (go to next location).  
 PC: recall last value of user's program counter.  
 RS: reset and return to monitor.  
 GO: start program at displayed address.  
 ST: terminate program execution.

The small slide switch on the keyboard sets KIM into single-instruction execution mode, allowing the user to step through his program for debugging purposes.

Unlike other evaluation kit monitors, KIM's version is written in a reasonably structured way allowing user programs access to a *useful set of subroutines*. Below are some examples:—

NAME	ADDRESS (HEX)	OPERATION
GETCH	1E5A	Reads a character from serial device (VDU or teletype) and returns the character in the accumulator.
OUTCH	1EA0	Prints the character whose code is in the accumulator on entry.
GETKEY	1F6A	Returns in the accumulator the key pressed on the on-board keyboard.
AK	1EFE	Checks status of the keyboard — returns with a non-zero accumulator if a key is depressed.
SCAND	1F1F	Refreshes the LED displays. The data actually displayed is from three RAM locations which can be set up by the user program.

These routines are all particularly useful for games programs etc. for entering and displaying numbers. Many games programs using these routines, and other useful software, are given in the "First Book of KIM", a U.S. publication which should shortly be available from distributors here.

### The Monitor

The monitor provides for the control of two external devices — a teletype/VDU and a cassette/reel-to-reel tape recorder. Programs are stored on tape with I.D. numbers to simplify tape search for a particular program (this can still take a long time, depending on how much tape has to be searched!). Serial line devices are interfaced via an on-board 20mA current loop circuit — the baud rate is software — adjusted by the monitor which times the width of the start bit for a particular character.

### Hardware

In terms of hardware, KIM provides the following major facilities:

(a) An MCS6502 microprocessor chip. The microprocessor contains several registers: *one accumulator* (for all arithmetic operations); a stack pointer (the stack is an area of store used to save subroutine and interrupt return addresses); two *index registers* (these allow efficient access of tables of data through a variety of addressing modes); a *program counter* (to access the instructions in store); and a *condition code register* (used to indicate microprocessor status after arithmetic operations, to set decimal or binary arithmetic modes, and to control interrupts).

The microprocessor operates on eight bit data words and all registers bar the program counter are 8-bits wide. The sixteen-bit program counter allows the access of instructions from anywhere in the 64K address space. Data is addressed via thirteen different addressing modes, including immediate, direct, absolute (allowing a full 16-bit address to be specified), indirect and various types of indexing utilising the index registers. Four types of interrupt are handled — reset, non-maskable, maskable and software (BHK instruction). The program counter and condition code register are automatically placed on the stack after interrupt — if a service routine is to use the remaining registers, they must be explicitly saved by the interrupt service program. All interrupts except reset can be vectored (i.e. "directed") to user

programs by changing the values of certain RAM locations.

The instruction set is shown in Figure 2 (mnemonics only). Arithmetic, looping, testing, logical and house-keeping types can be identified.

ADC	Add memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift left one bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result not Zero
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrease Memory by One
DEX	Decrease Index X by One
DEY	Decrease Index Y by One
EOR	"Exclusive Or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index X by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift Right One Bit (Memory or Accumulator)
NOB	No Operation
ORA	"Or" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	REturn from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store Index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Pointer
TYA	Transfer Index Y to Accumulator

Figure 2.

The microprocessor does have some restrictions when compared with others, for example the 6800. The eight-bit stack pointer and index registers limit the amount of store that can be addressed by them to 256 sequential locations. This restriction is often offset by the availability of a large variety of addressing modes.

(b) Two MCS6530 Peripheral Interface/ROM/Timer chips. These two complex LSI chips contain the monitor program in 2K x 8-bits of ROM (1K in each). Each chip also contains 64 bytes of RAM, most of which is used for scratch storage by the monitor. Apart from store, each MCS6530 contains 15 programmable input/output lines and an interval timer. One set of I/O lines from one chip are used to control the keyboard, display and cassette interface circuitry. The other set of 15 lines are available for the user to interface other external devices to KIM. Two data direction registers allow any I/O line to be programmed as an input or output.

(c) 1K x 8-bits of store (RAM) available for user written

programs. This is provided by eight 1K by 1-bit static RAM chips which respond to addresses 0000 to 03FF hex. The first 256 bytes (page 0) are directly addressable through 2-byte instructions, while the second 256 locations (page 1) is the processor stack area ("hard-wired" in the microprocessor).

(d) A tape recorder interface allowing the recording and retrieval of programs. Data is recorded at a rate of about 30 bytes per second but does *not* use the Kansas City Standard.

(e) Expansion and application connectors. One connector offers the unused set of 15 I/O lines for connection to user devices (lights, switches, audio amplifiers, central heating controls, etc.) The main expansion connector carries the 16 address and 8 data lines of the microprocessor, plus the necessary control and timing signals to allow more RAM, ROM etc. to be interfaced. Enabling signals for 4K of extra store are already decoded and brought out to the application connector. KIM does not have on-board buffering (nor any further space on the board) and so external hardware must have appropriate tri-state address and data buffers.



KIM 1 plays a game of chess.

### Summary

Through these hardware and software facilities the KIM microcomputer module offers, comparatively, a flexible and well-structured system for the hobbyist. The 1K of RAM and relatively large 2K monitor are certainly unusual for a unit of this type — the hobbyist will find the store sufficient to support a wide variety of games, of which a considerable number are now available through the large USA user community.

KIM can be obtained from the distributors listed at the end of this article — cost is about 149 pounds with a 90-day warranty period. Memory expansion boards and a system motherboard can also be obtained.

Finally, it should be stressed that the KIM microcomputer board offers facilities suitable for many applications, apart from home computing. The powerful real-time processing capabilities relative to its small size make it an attractive component for use in process-control, instrumentation and intelligent controller applications.

### REFERENCES

- (1) "The KIM-1 User Manual", MOS Technology Inc., Darmstadt, Germany. (Available from distributors).
- (2) "MCS6500 Microcomputer Family Hardware Manual", ditto above.
- (3) "MCS6500 Microcomputer Family Programming Manual", ditto above.
- (4) "The First Book of KIM, Eds. J. Butterfield, S. Ockers and E. Rehnke, (ORB, P.O. Box 311, Argonne, Illinois, 60439, USA). Price \$9.

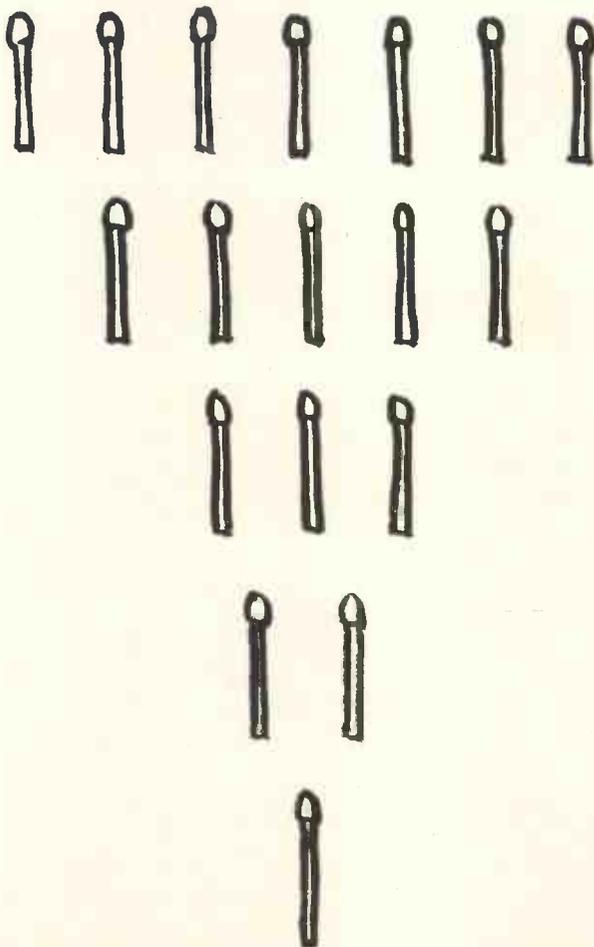
### DISTRIBUTORS

G.R. Electronics Ltd.  
80 Church road,  
Newport, Gwent  
NPT 7EH. Tel: 67426  
A. Marshall (London) Ltd.  
40-42 Cricklewood Broadway  
London, NW2 3ET Tel: 01-452 0161

# TAKE IT OR LEAVE IT

ROGER HARGRAVE

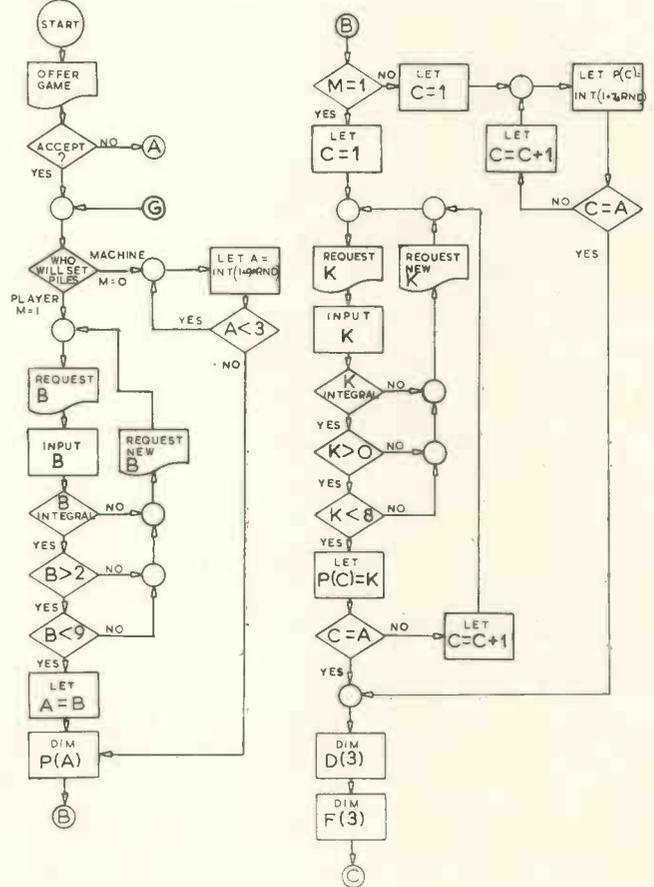
## NIM—An Exercise in Flowcharting



The game of Nim is so called from an obsolete word meaning to "take". It is played by removing objects from groups of similar objects — traditionally piles of matches are used, but in theory any objects will do, even Jumbo Jets if you can hijack enough of them! Actually there are two versions of the game — in one, which you might call positive Nim, the objective is to force your opponent to pick up the last match from the last pile, while in the "negative" version you endeavour to pick it up yourself. In both versions the players take alternate turns and may only touch one pile at each turn. At least one match must be removed each time, and if desired the whole pile may be taken. This article is an attempt to produce a flowchart for a computer based version of "positive" Nim. Notation from BASIC is used in many of the boxes as a form of shorthand, but should not be taken too literally; for example in the use of the INPUT statement when setting up the piles.

### NIM 1

#### PRELIMINARIES



#### Preliminaries

The first step must be to set some limit on the size of the game. In principle, both the number of piles and the numbers of matches in the piles (which may of course differ from pile to pile) can be as large as you care to make them. In practice, neither the programme nor the memory allocated to storing the data relating to the piles can be of unlimited size, and we must also consider the possibility of displaying the position. For these reasons, nine was chosen as the upper limit of the number of piles, and since at least three piles are needed to give a reasonable game this is set as the lower limit. As will become apparent later, the programme becomes considerably more lengthy as the maximum size of each pile increases by a factor of two, and therefore seven, being one short of  $2^3$ , is taken as the highest number allowable. This is not unrealistic when compared with the sort

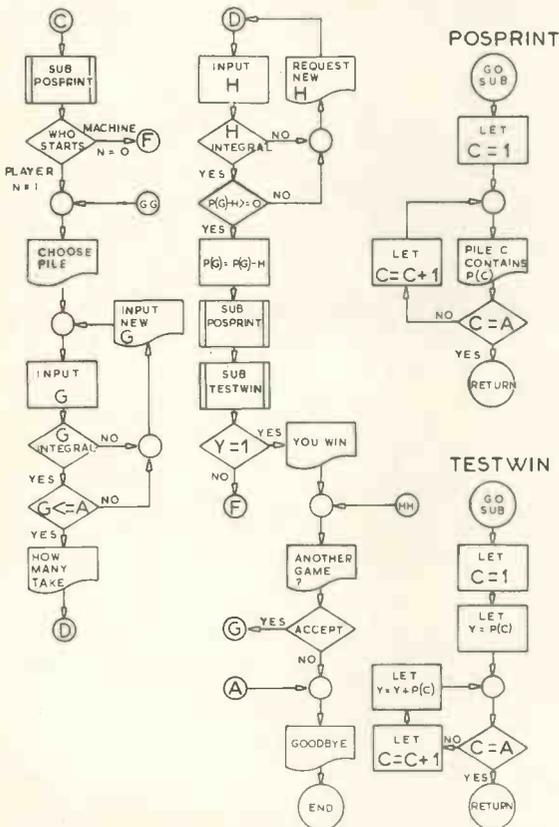
of position which would probably be set up from a small box of matches ("average contents 47" or whatever).

Starting the game is fairly straightforward. First an introduction is printed, and the machine offers to play. If the offer is accepted, the human (hereinafter referred to as the player, as the legal profession would say) is given the chance to set up the piles, to eliminate the suspicion that the machine has "fixed" the game from the outset. If this is taken up, the machine checks that the player's inputs in response to its prompts are integral and within the set limits; it rejects them and repeats its original prompts if they are unsuitable, but otherwise sets up array P and goes through subroutine POSPRINT to give a display of the position. Should the player prefer, there is an alternative routine for the machine to provide the starting position. It *doesn't* fix this to its own advantage, though it could easily be made to do so.

Memory space is allocated to two small arrays called D and F which will be used during the machine's turn, and the game can start.

## NIM 2

### PLAYER'S TURN - END-SUBROUTINES

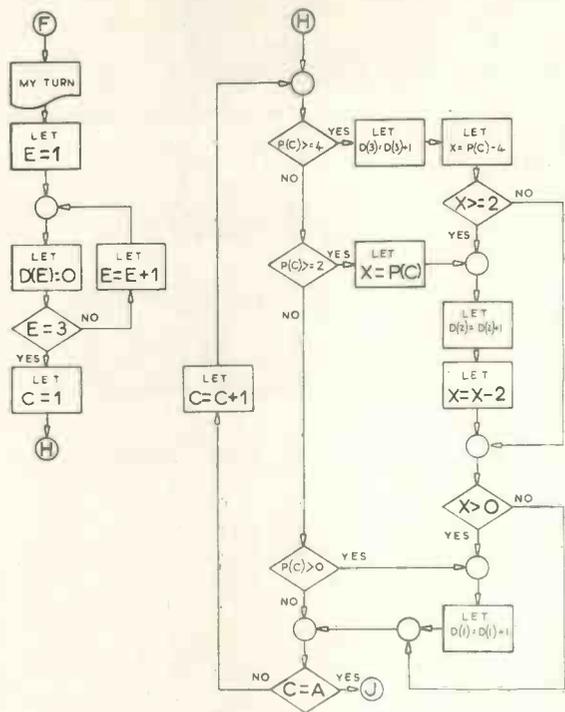


### Player's turn

The player is now offered the chance of first turn, and if this is accepted, all the machine has to do is to make the necessary prompts and check that the player's move is possible and within the rules of the game. It then goes to POSPRINT to display the new position, followed as a matter of course by TESTWIN which is a subroutine to check if the game has been won by the move which has just been made. Since there were at least three piles at the start, the player cannot in fact have won yet, so the machine announces that it is now its turn. Naturally, if the player declines the first move, then the programme branches at once to the machine's move.

## NIM 3

### MACHINE'S TURN - COUNT COEFFICIENTS



### Machine's turn

We have now reached the most interesting stage, where we must try to devise a winning strategy for the machine to apply. Many books on mathematical games have a section on Nim, and the interested reader can discover from one such that there are two kinds of position which can confront a player when he begins his turn. One, the "losing" position is such that whatever move he makes, he can only win if his opponent later makes a mistake; the other, "winning", position can be converted by the move about to be made into a losing position for the opponent's next turn.

The method of determining whether a position is losing or winning can be difficult to perform in one's head when large numbers are involved, but it is in theory fairly simple. First the number of matches in each pile is expressed as the sum of a series of powers of two, just as when converting from decimal to binary notation. For example,  $6 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$  or  $2 = 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$ , and so on. Then for each power of two in turn, all the coefficients from all the piles are totalled together. Figure 1 gives an example of how this is done: -

Figure 1

Pile No.	Number of matches in pile			
	Decimal	Binary		
		$2^2$	$2^1$	$2^0$
1	7	1	1	1
2	3	0	1	1
3	4	1	0	0
4	2	0	1	0
5	2	0	1	0
6	1	0	0	1
Totals of coefficients		2	4	3

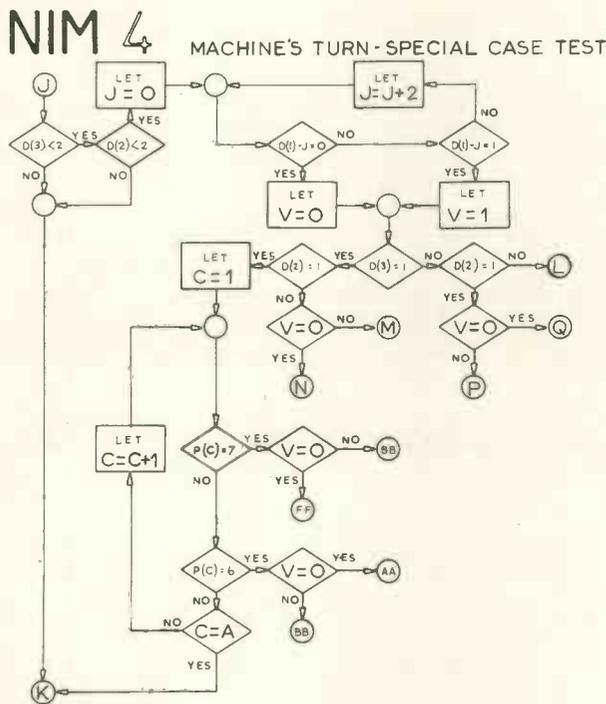
Having found the totals of the coefficients, it is a general rule that if all the totals are even, i.e. exactly divisible by two, then the position is a losing one. The example in Figure 1 is, however, a winning one, and can be converted into a losing one for the opponent by

reducing the  $2^0$  column by one. It makes no difference how this is done — it is equally effective whether one removes the necessary single match from pile 1, 2 or 6.

This rule works absolutely for negative Nim, but in positive Nim there is a "special case" position, for in this version if none of the piles contains more than one match, or to put it another way, if the totals of the coefficients for powers of two greater than 2 are all zero, then *in that case only* it is a losing position if the total of the coefficients of  $2^0$  is odd. This includes the ultimate situation where there is a single pile of one match, and the game is won.

**Initial check**

The first thing to do is therefore to obtain the totals of the coefficients from the figures contained in array P. Array D is cleared, and then by means of a network testing for reducing powers of two, D(3) is loaded with the total of the coefficients of  $2^2$ , and D(2) and D(1) with those for  $2^1$  and  $2^0$  respectively.



**Special case check**

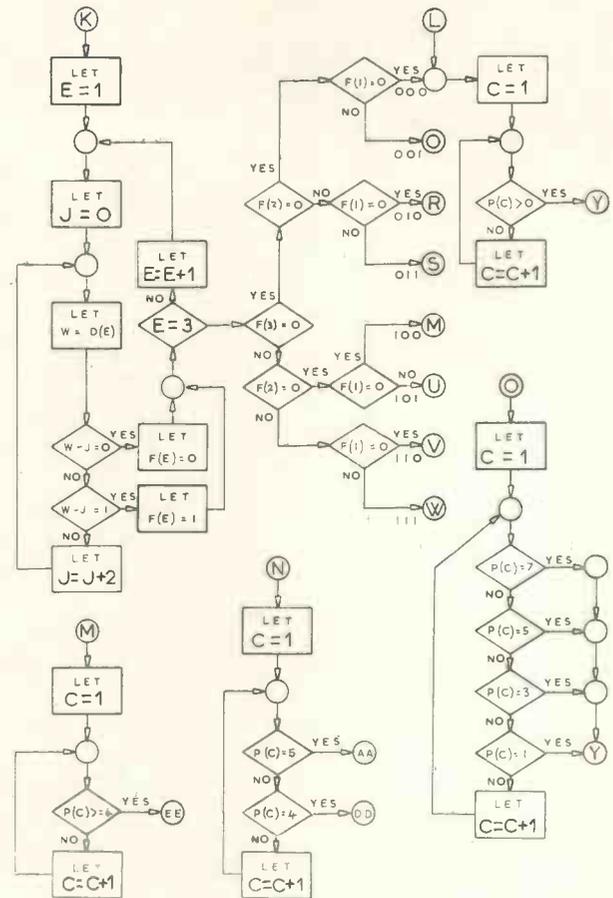
Next, a check is made to see if the position is such that a special case losing position can be set up. This always happens towards the end of a game, unless the machine is losing.

Since only one pile may be touched at a time, if either D(2) or D(3) equals two or more, a special case position is not attainable, so the programme goes on at once to the general case section. If one *only* of D(2) and D(3) is equal to one, then a losing position can be achieved. If both are equal to zero, then the game is already won or lost beyond recall, and all the machine can do is to remove the lone match in one of the piles. The most complicated situation occurs if D(2) and D(3) both equal one, since this can arise in two ways. Either there is one pile containing 6 or 7 matches, in which case a losing position can again be achieved, or else there is one pile with 4 or 5 and another with 2 or 3 — in this situation the programme goes on to the general case test.

In those cases where action can be taken at this stage, account must also be taken whether the existing total of the coefficients of 2 is odd or even. Variable V is set as an indicator of this, 0 meaning even and 1, odd. Figure 2 is a truth table showing the action to be taken in each case. X as usual indicates a "don't care" situation.

**NIM 5**

**MACHINE'S TURN - GENERAL TEST**



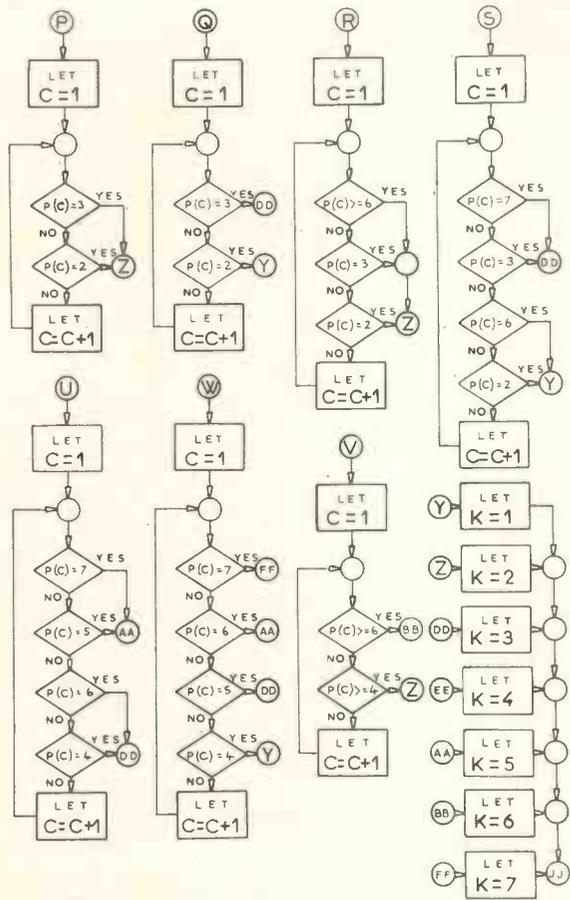
It should by now be clear why the programme would be more complex if the number of matches in each pile were allowed to exceed 7, since a D(4) would then be needed, and Figure 2 would be much more complicated to accommodate this. A similar increase in complexity, would also be apparent in the general case section.

**Figure 2**

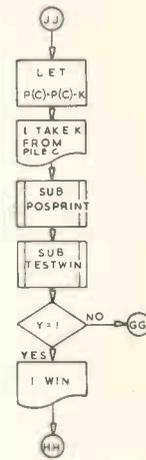
D(3)	D(2)	V	Action
$\geq 2$	X	X	Go to general case
X	$\geq 2$	X	Go to general case
1	0	0	Find pile with 5 and deduct 5; or pile with 4 and deduct 3 (to change V)
1	0	1	Find pile with 5 and deduct 4; or pile with 4 and deduct 4 (to preserve V)
0	1	0	Find pile with 3 and deduct 3; or pile with 2 and deduct 1 (to change V)
0	1	1	Find pile with 3 and deduct 2; or pile with 2 and deduct 2 (to preserve V)
1	1	0	Find pile with 7 and deduct 7; or pile with 6 and deduct 5 or if no pile has 6 or 7, go to general case
1	1	1	Find pile with 7 and deduct 6; or pile with 6 and deduct 6; or if no pile has 6 or 7, go to general case
0	0	X	Remove one from any pile

# NIM 6

MACHINE'S TURN - ACTION ON TESTS



# NIM 7



Note that once a game has been started, it must be played to the bitter end, in order to exit after TESTWIN, though with a little experience the result can be seen quite clearly in the game. To be honest, Nim is not that much fun to play once you know the secret, and is therefore more suitable as a demonstration piece than as a serious contest of wits. It is possible to beat the programme, either by taking first move in a winning position or by making the machine take first move in a losing position. The player can also cheat, but I leave the reader to work out how!

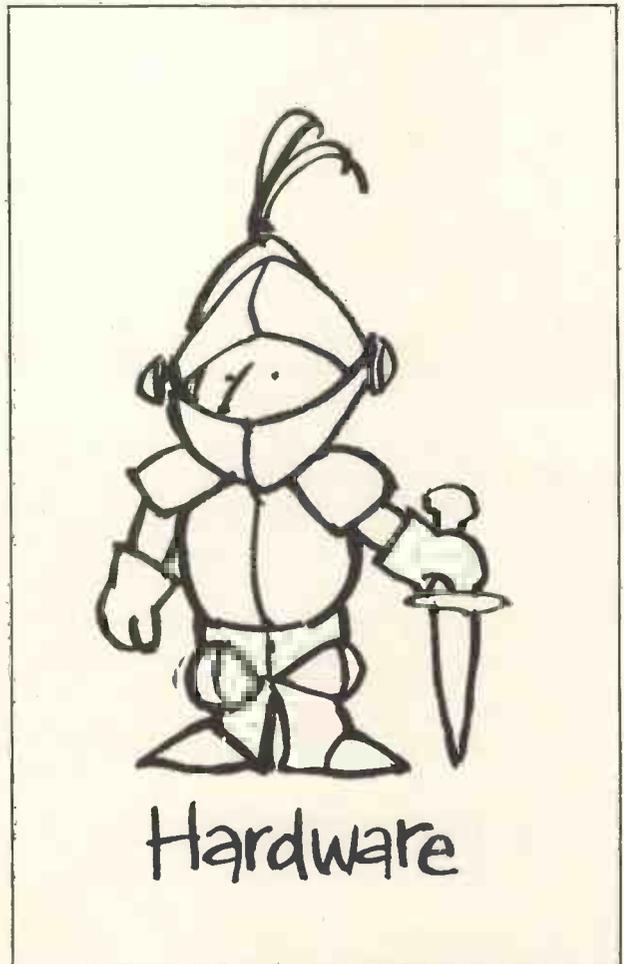
### General case check

If the situation calls for a general case check, then array F is loaded with 1 or 0 according to whether the contents of the corresponding cell in array D are odd or even. For example, if D(3) is odd then F(3) is 1, if D(2) is even then F(2) is 0, and so on. Action is then taken according to Figure 3. Note that in case 000, the machine is faced with a losing position! In this situation it does not really matter what move it makes, but the advantage probably lies in keeping the position as complex as possible for the player, and therefore a minimal move is specified in this case.

Figure 3

F(3)	F(2)	F(1)	Action
0	0	0	Deduct one from any pile
0	0	1	Find an odd pile (7, 5, 3 or 1) and deduct 1
0	1	0	Find a pile with 7, 6, 3 or 2 and deduct 2
0	1	1	Find a pile with 7 or 3 and deduct 3; or with 6 or 2 and deduct 1
1	0	0	Find a pile with 4 or more and deduct 4
1	0	1	Find a pile with 6 or 4 and deduct 3 or with 7 or 5 and deduct 5
1	1	0	Find a pile with 7 or 6 and deduct 6; or with 5 or 4 and deduct 2
1	1	1	Find a pile with 7 and deduct 7; or with 6 and deduct 5; or with 5 and deduct 3; or with 4 and deduct 1

Having made its move the machine reports it to the player, displays the new position by means of POS-PRINT, and performs TESTWIN. If it has won it announces the fact and offers a fresh game. If not, it hands the move back to the player.



NEIL HARRISON

# SHOWING CHARACTERS

## Software for a Memory Mapped VDU

Like many other computer hobbyists I chose a 'memory mapped' TV display for alphanumeric output on my home computer. This technique is very flexible and cost effective for the amateur since the job of positioning the characters on the screen and generally making it behave like a commercial VDU is done by the microprocessor rather than a pile of dedicated logic. Naturally, to do this the microprocessor must have a program which performs the various VDU functions. This article describes the TV driver program developed for my own Z80 based system but the techniques used are quite simple so that modification to another Z80 system or even a different microprocessor should not be difficult.

## The Memory Mapped Display

Figure 1 shows the essential features of a memory mapped TV display. The read/write memory (RAM) is used to store in ASCII code the characters to be displayed. The Video display circuitry constantly reads the data from this memory and converts the ASCII code to displayed characters on the screen. The feature which makes the display 'memory mapped' is that the character-store memory is also interfaced to the microprocessor busses. This means that the microprocessor can read from or write to any character position on the screen just like any other RAM memory byte. As a result, putting characters on the screen is simply a matter of moving data from one memory address to another; a simple enough task for any microprocessor. In general the programmer can ignore the complications of the display circuitry and simply relate a character position on the screen to a memory address. Figure 2 shows the display format and memory addresses for my system. The character store memory is from E000 to E3FF, these addresses corresponding to the top left and bottom right of the display respectively.

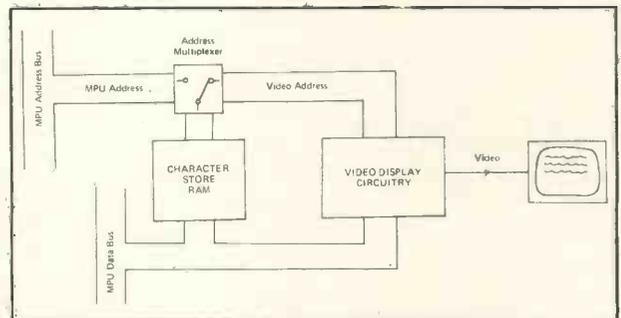


Fig. 1 The memory mapped display

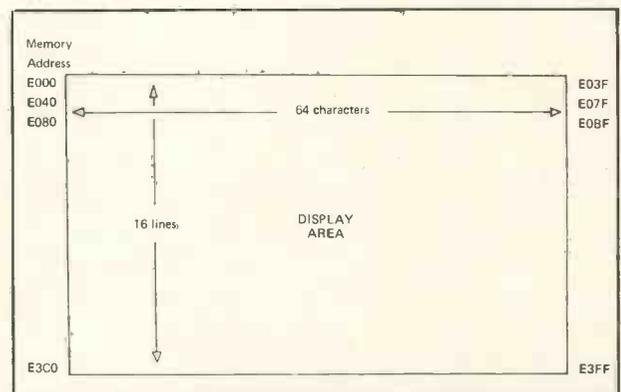


Fig. 2 Character display format

## The VDU Routines

The listing shows the VDU driver which consists of two sections each called as a subroutine by the user program. The shorter section is an initialisation routine, INIT, called via location E803. This clears the display of random letters and numbers caused by the character

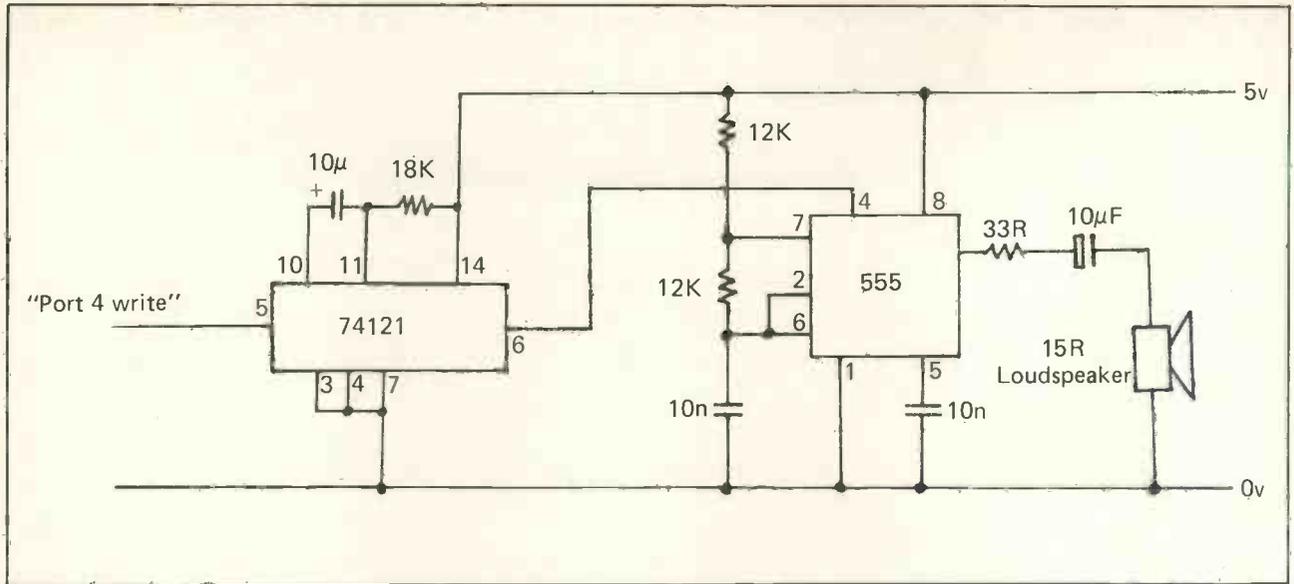


Fig. 3 'Bell' generator

store RAM having undefined data when the system is powered up. This is simply done by writing an ASCII 'space' code (20 Hex) into every location of the display RAM. In addition the contents of four bytes of scratchpad RAM at E400-E403 used by the main character output routine are set by copying values from a table at E8D7. The initialisation routine need only be called once when the system is powered up.

The main character output routine, TYPE, called via E800, is responsible for placing the ASCII character to be 'printed' on the screen in the correct position. TYPE is called as a subroutine with the ASCII code for the character to be 'printed' in the C register. All registers are preserved except the Accumulator which returns with the code of the character displayed. The current printing position, i.e. where the next character will be put, is shown by the cursor, an 'underline' ( \_ ) which is moved as successive characters are displayed. After initialisation characters are displayed starting on the top line and working down the screen until the bottom line is reached. As further new lines are 'printed' the displayed text 'scrolls' upwards, so that the most recent sixteen lines of output are shown. The following six control codes are recognised, all others are ignored:

CARRIAGE RETURN	( $\phi$ D Hex)	places the cursor at the beginning of the current line.
LINE FEED	( $\phi$ A Hex)	advances the cursor by one line, scrolling the display if necessary.
FORM FEED	( $\phi$ C Hex)	clears the display and places the cursor in the top left hand corner of the display.
BACKSPACE	( $\phi$ S Hex)	moves the cursor back to the previous character position.
TABULATE	( $\phi$ 9 Hex)	moves the cursor to the next character position which is a multiple of eight.
BELL	( $\phi$ 7 Hex)	executes an OUT instruction to Z80 port 4 which rings a 'bell' on my system.

The OUT instruction to port 4 associated with the 'Bell' code is used to trigger off a tone from the circuit shown in Figure 3. The monostable ICI has an output pulse width of approximately half a second which enables the oscillator, IC2, producing a short tone from the loudspeaker. This useful feature was relatively easy for me to implement as the decoding circuitry for port 4 already existed to provide parallel input from my ASCII keyboard. The OUT instruction at E866 and the NOP which follows can be replaced by a subroutine call

so that a 'Bell' routine to suit your system can be added fairly easily.

Lower case character codes are detected and converted to upper case by the instructions at E830-E835. If your display has lower case characters these bytes should be replaced by NOPs.

In addition to the character store memory the routines need four bytes of RAM to store temporary variables. I used some dedicated memory at E400-E403; though of course, any convenient place in memory can be used if the routines are modified accordingly. The data stored in DLYCNT at E400 is used to control the output speed of the display;  $\phi\phi$  gives the fastest, FF Hex the slowest. CURPOS (E401) and LINENO (E402) hold the current position of the cursor. CURPOS holds a number in the range 0 to 63 decimal, corresponding to the position (left to right) of the cursor on a line. LINENO holds the number of the line that the cursor is on, 0 to 15 decimal, top to bottom of the screen. These two numbers are used by the subroutine CURADR (E8A7) to calculate the true memory address of the cursor. The cursor is displayed by simply writing the code for 'underline' to the correct screen position. CHRSAV (E403) is used to save the character that was previously stored there so that it may be replaced if necessary when the cursor position is changed.

**Modifications**

The following table shows the parts of the VDU driver which must be changed for a different display format or character memory address. The correct values for your system can be calculated from the formulae shown.

Address	Data	Comments
E83C	D	Number of characters per line
E847	L-1	Number corresponding to bottom line
E84B & C	V+B	Start address of next to top line
E84E & F	V	Start address of top line
E851 & 2	(L-1) x B	Number of bytes to be moved in a scroll
E856 & 7	V+(L-1)xB	Start address of bottom line
E8A8 & 9	V	Start of character RAM
E8AB & C	B	Number of bytes per line
E8CA & B	V	Address of first character, top line
E8CD & E	V+1	Address of next character, top line
E8D0 & 1	(BxL)-1	Byte count for CLEAR

Where the letters V, B, D and L stand for the following:

- V = start address of character store RAM
- B = number of bytes per line
- D = number of displayed characters per line
- L = number of lines

In the program listing the following values apply: V = E000, B = 40, D = 40 and L = 1Q (all hexadecimal).





# HANGMAN

MIKE DENNIS

## Introduction

This is a version of the game that you probably played as a child to while away those long winter evenings before your father bought you a home-computer! It was played in the 'old days' between two people. Your opponent chose a word and told you how many letters it had. Your aim was to try and work out what that word was by guessing letters. If you had a good guess, your opponent would fill that letter in its' appropriate place in the hidden word. If the letter occurred more than once then more spaces would be filled in. If you made a wrong guess, however, your opponent would draw in one more line to make up a graphic picture of a hangman — hence the name of the game. It was fairly standard to let you have seven bad tries before you were deemed to have lost and were told the word.

In this version, the computer is your opponent and chooses a word at random (well, pseudo-random) from a supplied list of words in memory — the library. It calculates the number of letters in the word, tells you how many and then asks you for your guesses. The sample run should make the rest of it fairly obvious.

### Sample runs

```
I am thinking of a word that has
5 letters.
Your guess? A ----- H
Your guess? E ---EE--
Your guess? I ---EE-- HA
Your guess? R ---EE-- HAN
Your guess? S ---EE-- HANG
Your guess? L ---EEL
Your guess? W W--EEL
Your guess? H WHEEL
Well done!
Again?
N
*
```

Some points to note.

- 1/ We haven't got graphics but it should be straightforward to incorporate them.
- 2/ Good tries inhibit the printing of the Hangman string.
- 3/ When you lose, the computer tells you what the word was.
- 4/ At the end you have a choice of playing again or not. N will return you to your system monitor. Any other key-stroke will play again.
- 5/ The strings must be entered exactly as they are written. CRLF means Carriage Return, Line Feed. If your monitor automatically line feeds after CR then modify all 'LFs' to 20 (ASCII space).

## More general program points

Originally written using the Z-80 instruction set, I have hand-assembled the program for 8080 users. I apologise for the occasional quaint mnemonic. As a result of the revision, there are no relative jumps etc in the listing and the program is fairly long — about 500 bytes excluding your list of words. I have, however, tried to keep any count loops in the B register to facilitate any re-conversion back into Z-80 code.

To interface the program to your own system requires the following alterations to the listing:—

- 1/ Insert the address of your keyboard input routine at 0103. The program expects to find an ASCII character in register A.
- 2/ Insert the address of your character output routine at 0109. The program outputs in the C register. If you decide to alter the program to output in register A then watch out for the instruction at 01C5 for it must be changed to 3E.
- 3/ Insert the address of your own character string routine at 010C. This routine must:—
  - a/ Increment the HL registers
  - b/ Output the data pointed at by HL as an ASCII character
  - c/ Decrement B
  - d/ Test for zero and loop back to (a) if non-zero.

I know that there are better ways to output string but this program was written using TDL's Zapple monitor. As it stands, the program will run without any modification in a system using this monitor.

- 4/ Insert the address of your monitor re-entry point at 0193.

Any word of nine characters or less can be used. The words are written into the library in ASCII. You MUST enter 00 after every word. On my keyboard, this is Shift-Control P. The maximum number of words allowed is 254 and you must enter the number of words *plus* one into address 0291. For example, if you have twenty words then the data for 0291 must be 15 (decimal 21). In practice, the pseudo-random number generator is not necessary as you soon forget what the words are. However, I have written it as a sub-routine for use in other programs. Another possibly useful sub-routine is that at address 010F which clears a screen of 40 x 24 characters.

Hope you have as much fun playing it as I did in writing it. Your children certainly will !

HANGMAN by Mike Dennis Copyright 1978

```

0100 C3 010F' Jmp ;past vectors
0103 KI C3 F003' Keybd.in routine ;
0106 00 0000 ;
0109 C0 C3 F009' Chr.out routine ;
010C S0 C3 F440' Stg.out routine ;

010F KLR 21 03C1 LDHL,03C1 Page count+1 ; for 960 characters
0112 OE 1E LDC,1E ; the cursor
0114 KLR CD 0109' Call.....CO ;
0117 OE 20 LDC,20 Space ;
0119 2B DEC HL ;
011A 7D LDA,L ;
011B B4 ORA,H ; Cleared screen yet?
011C C2 0114' JNZ.....KLR ;
; Real programme starts here
; Sets up buffers
011F Init. 21 02A0' LDHL,Data+1 ; We're going to
0122 11 02B0' LDDE,Buffer +1 ; initialise the sixteen
0125 06 10 LDB,10 ; data buffers.
0127 Do it 2B DEC HL ;
0128 7E LDA(HL) ;
0129 1B DEC DE ;
012A 12 LD(DE),A ;
012B 05 DEC B ; Finished it?
012C C2 0127' JNZ.....Do it ; No, so carry on.

; Outputs sign-on message
; Gets mystery word from list
012F Start 06 22 LDB,22 Msg1 ; Set up for Message 1.
0131 21 022C' LDHL,I am..... ; Output it. Finishes with HL
0134 CD 010C' CALL.....S0 ; registers pointing at where
0137 CD 01A0' CALL.....Wrd cnt ; wrd cnt will go.
013A 06 0A LDB,0A ;
013C CD 010C' CALL.....S0 ; Finish message 1 " x letters."

; Asks for your guess
; Goes and looks for a match
013F Your 06 0D LDB,0d Msg2 ; Set up for message 2.
0141 Guess 21 0258' LDHL, Your guess? ;
0144 CD 010C' CALL.....S0 ;
0147 CD 0103' CALL.....Space ; Output it
014A E6 7F AND A,7F ; Output a space for tidy display.
014D 4F LDC,A ; Get your guess.
0150 CD 0109' CALL.....CO ; Remove parity bit.
0153 57 LDD,A ; Obey rule for Zapple.
0154 CD 0103' CALL.....Space ; Echo guess.
0157 LDA,D ; Store try somewhere safe!
0158 CD 01E6' CALL.....Match ; Output a space
; Restores try
; Go see if good guess.

; Outputs mystery word so far
; Outputs HANGMAN to show how
; many guesses you've got wrong.
; Checks to see if word guessed
; correctly or if you lost.

```

```

; Store start address of word.
; Go look for end of this word.
; Found it when A=00. E then has
; number of letters in word.
; Decrement random number.
; If B=0, then we've got the word.
; Put wrd ct. in A.
; ASCII bias added.
; Restore HL.
; Save wrd.ct. in middle msg.1.

22 02A5' LD 02A5,(HL)
23 INC HL
IC INC E
BE CPA,(HL)
01B6 JNZ.....No
01B9 DEC B Found word?
01BA C2 01AD' JNZ.....Count
01BD LDA,E
01BE ADD,50
01C0 POP HL
01C1 LD(HL),A
01C2 RET

01C3 Space 06 02 LDB,02 ; Set up for two spaces.
01C5 OE 20 LDC " " ;
01C7 CD 0109' CALL.....CO ;
01CA 05 DEC B Finished? ;
01CB C2 01C7' JNZ.....No ;
01CE C9 RET ;
01CF 00 ;

;RANDOM. This is a pseudo-random number
;generator. It will generate a string of
;256 different bytes but then repeat.

01D0 Random 5A 022B' LDA, last no. ;
01D3 B7 ORA zero? ;
01D4 C2 01D8' JNZ.....Skip 1 ;
01D7 3C INC A ;
01D8 Skip 1 47 LDB,A Save byte ;
01D9 E6 8E ANDA,8E Mask ;
01DB 78 LDA,B Restore it ;
01DC EA 01E0' JFPE.... Skip 2 ;
01DF 37 SCF ;
01E0 Skip2 17 RLA ;
01E1 32 022B LD 022B,A save no. ;
01E4 47 LDB,A ;
01E5 C9 RET ;

;MATCH. Tests your try with the letters
;in the hidden word. Stores your good tries
;at 02A0' and also in buffer. Puts try in
;C if a good one.1

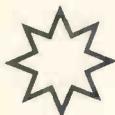
01E6 Match 48 LDC,B clear c ; Cheap zero in C.
01E7 43 LDB,E ; Get wrd ct. in B.
01E8 D5 PUSH DE ; Save wrd ct.
01E9 21 02A0' LDHL,02A0 ; Point to "no.of good tries".
01EC E5 PUSH HL ; Save it.
01ED 2A 02A5' LDHL,02A5 ; Get start address of hidden word.
01F0 11 02A7' LDDE,02A7 ; Get start address of buffer store.
01F3 cont. BE CPA(HL) match? ; Does your try match first letter in word?
01F4 C2 01FC' JNZ.....bad try ; No....bad try.
01F7 12 LD(DE),A ; Yes so save letter in buffer in right
01F8 4F LDC,A ; place and also in C as 'good try flag'.

```

```

015B OK? 50 LDD,B
015C A1 ANDA,C
015D C2 0164 JNZ,.....Yes
015E 2E A3 , LDA,A3
0160 34 INC(HL)
0161 56 LDD(HL)
0162 43 LDB,E
0163 2E A7' LDH,A7 Point to buffer
0164 CD 010C' CALL,.....SO
0165 42 LDB,D
0166 INC B
0167 21 0298' LDHL, Hangman
0168 CD 010C' CALL,.....SO
0169 LDA,D
0170 FE 07 CPA,07
0171 JZ,.....You lost
0172 You win 7B LDA,E
0173 2E A0' LDI, A0
0174 BE CPA,(HL) No?
0175 C2 013F' JNZ,.....Your guess
0176 06 14 LDB,14
0177 21 0265' LDHL, Well done
0178 CD 010C' CALL,.....SO
0179 CD 0103' CALL,.....KI
0180 AND A,7F
0181 E6 7F LDC,A
0182 CD 0109' CALL,.....CO
0183 FE 4E CPA,4E "N"?
0184
0185 ;At this point you have finished a
0186 ;game. The computer has asked whether
0187 ;you want another go, accepted your
0188 ;answer and tested for N(for no, silly!)
0189
0192 CC FOLE' CALL,.....monitor
0193 C3 010F' CALL,.....KLR
0194 00 0000
0195 00 0000
0196 00 0000
0197 00 0000
0198 00 0000
0199 00 0000
0200
0201 ;Subroutines from here.
0202
0203 ;Wrd ct. Gets random number, tests for in
0204 ;range. Uses number to find word in list.
0205 ;Works out number of letters in word.
0206 ;Stores start address of word at 02A5'
0207
0208 Wrd ct E5 PUSH HL
0209 LDHL, No. words
0210 CD 01D0' CALL,.....Random
0211 96 SUB,(HL)
0212 D2 01A4' JNC,.....Go
0213 2E AF' LDH,AF Wrd list-1
0214 97 SUB A
0215 LDE,A
0216 INC HL
0217
0218 ; Cheap zero in D.
0219 ; If good try, then C is not zero.
0220 ; Yes so don't inc. hang count.
0221 ; Set up address to inc. hang ct.
0222
0223 ; Inc. hang ct.
0224 ; Get new count in D.
0225 ; Put wrd ct. in B.
0226 ; Output word guessed so far.
0227 ; Set up for hangman
0228 ; For if B was 00, B would decr.
0229 ; 255 times and this routine
0230 ; would output rubbish.
0231 ; Have you lost?
0232
0233 ; Yes, you did
0234 ; Have you won?
0235 ; Compare wrd ct. with no. of
0236 ; good tries. If equal, you win.
0237 ; Try again.
0238 ; Set up for message 3.
0239
0240 ; Get character.
0241 ; Kill parity bit.
0242 ; Obey rule for Zapple
0243 ; Echó it.
0244 ; Was it 'n'? If so then.....
0245
0246 ; Yes, you did
0247 ; Have you won?
0248 ; Compare wrd ct. with no. of
0249 ; good tries. If equal, you win.
0250 ; Try again.
0251 ; Set up for message 3.
0252
0253 ; Get character.
0254 ; Kill parity bit.
0255 ; Obey rule for Zapple
0256 ; Echó it.
0257 ; Was it 'n'? If so then.....
0258
0259 ; Yes, you did
0260 ; Have you won?
0261 ; Compare wrd ct. with no. of
0262 ; good tries. If equal, you win.
0263 ; Try again.
0264 ; Set up for message 3.
0265
0266 ; Get character.
0267 ; Kill parity bit.
0268 ; Obey rule for Zapple
0269 ; Echó it.
0270 ; Was it 'n'? If so then.....
0271
0272 ; Yes, you did
0273 ; Have you won?
0274 ; Compare wrd ct. with no. of
0275 ; good tries. If equal, you win.
0276 ; Try again.
0277 ; Set up for message 3.
0278
0279 ; Get character.
0280 ; Kill parity bit.
0281 ; Obey rule for Zapple
0282 ; Echó it.
0283 ; Was it 'n'? If so then.....
0284
0285 ; Yes, you did
0286 ; Have you won?
0287 ; Compare wrd ct. with no. of
0288 ; good tries. If equal, you win.
0289 ; Try again.
0290 ; Set up for message 3.
0291
0292 ; Get character.
0293 ; Kill parity bit.
0294 ; Obey rule for Zapple
0295 ; Echó it.
0296 ; Was it 'n'? If so then.....
0297
0298 ; Yes, you did
0299 ; Have you won?
0300 ; Compare wrd ct. with no. of
0301 ; good tries. If equal, you win.
0302 ; Try again.
0303 ; Set up for message 3.
0304
0305 ; Get character.
0306 ; Kill parity bit.
0307 ; Obey rule for Zapple
0308 ; Echó it.
0309 ; Was it 'n'? If so then.....
0310
0311 ; Yes, you did
0312 ; Have you won?
0313 ; Compare wrd ct. with no. of
0314 ; good tries. If equal, you win.
0315 ; Try again.
0316 ; Set up for message 3.
0317
0318 ; Get character.
0319 ; Kill parity bit.
0320 ; Obey rule for Zapple
0321 ; Echó it.
0322 ; Was it 'n'? If so then.....
0323
0324 ; Yes, you did
0325 ; Have you won?
0326 ; Compare wrd ct. with no. of
0327 ; good tries. If equal, you win.
0328 ; Try again.
0329 ; Set up for message 3.
0330
0331 ; Get character.
0332 ; Kill parity bit.
0333 ; Obey rule for Zapple
0334 ; Echó it.
0335 ; Was it 'n'? If so then.....
0336
0337 ; Yes, you did
0338 ; Have you won?
0339 ; Compare wrd ct. with no. of
0340 ; good tries. If equal, you win.
0341 ; Try again.
0342 ; Set up for message 3.
0343
0344 ; Get character.
0345 ; Kill parity bit.
0346 ; Obey rule for Zapple
0347 ; Echó it.
0348 ; Was it 'n'? If so then.....
0349
0350 ; Yes, you did
0351 ; Have you won?
0352 ; Compare wrd ct. with no. of
0353 ; good tries. If equal, you win.
0354 ; Try again.
0355 ; Set up for message 3.
0356
0357 ; Get character.
0358 ; Kill parity bit.
0359 ; Obey rule for Zapple
0360 ; Echó it.
0361 ; Was it 'n'? If so then.....
0362
0363 ; Yes, you did
0364 ; Have you won?
0365 ; Compare wrd ct. with no. of
0366 ; good tries. If equal, you win.
0367 ; Try again.
0368 ; Set up for message 3.
0369
0370 ; Get character.
0371 ; Kill parity bit.
0372 ; Obey rule for Zapple
0373 ; Echó it.
0374 ; Was it 'n'? If so then.....
0375
0376 ; Yes, you did
0377 ; Have you won?
0378 ; Compare wrd ct. with no. of
0379 ; good tries. If equal, you win.
0380 ; Try again.
0381 ; Set up for message 3.
0382
0383 ; Get character.
0384 ; Kill parity bit.
0385 ; Obey rule for Zapple
0386 ; Echó it.
0387 ; Was it 'n'? If so then.....
0388
0389 ; Yes, you did
0390 ; Have you won?
0391 ; Compare wrd ct. with no. of
0392 ; good tries. If equal, you win.
0393 ; Try again.
0394 ; Set up for message 3.
0395
0396 ; Get character.
0397 ; Kill parity bit.
0398 ; Obey rule for Zapple
0399 ; Echó it.
0400 ; Was it 'n'? If so then.....
0401
0402 ; Yes, you did
0403 ; Have you won?
0404 ; Compare wrd ct. with no. of
0405 ; good tries. If equal, you win.
0406 ; Try again.
0407 ; Set up for message 3.
0408
0409 ; Get character.
0410 ; Kill parity bit.
0411 ; Obey rule for Zapple
0412 ; Echó it.
0413 ; Was it 'n'? If so then.....
0414
0415 ; Yes, you did
0416 ; Have you won?
0417 ; Compare wrd ct. with no. of
0418 ; good tries. If equal, you win.
0419 ; Try again.
0420 ; Set up for message 3.
0421
0422 ; Get character.
0423 ; Kill parity bit.
0424 ; Obey rule for Zapple
0425 ; Echó it.
0426 ; Was it 'n'? If so then.....
0427
0428 ; Yes, you did
0429 ; Have you won?
0430 ; Compare wrd ct. with no. of
0431 ; good tries. If equal, you win.
0432 ; Try again.
0433 ; Set up for message 3.
0434
0435 ; Get character.
0436 ; Kill parity bit.
0437 ; Obey rule for Zapple
0438 ; Echó it.
0439 ; Was it 'n'? If so then.....
0440
0441 ; Yes, you did
0442 ; Have you won?
0443 ; Compare wrd ct. with no. of
0444 ; good tries. If equal, you win.
0445 ; Try again.
0446 ; Set up for message 3.
0447
0448 ; Get character.
0449 ; Kill parity bit.
0450 ; Obey rule for Zapple
0451 ; Echó it.
0452 ; Was it 'n'? If so then.....
0453
0454 ; Yes, you did
0455 ; Have you won?
0456 ; Compare wrd ct. with no. of
0457 ; good tries. If equal, you win.
0458 ; Try again.
0459 ; Set up for message 3.
0460
0461 ; Get character.
0462 ; Kill parity bit.
0463 ; Obey rule for Zapple
0464 ; Echó it.
0465 ; Was it 'n'? If so then.....
0466
0467 ; Yes, you did
0468 ; Have you won?
0469 ; Compare wrd ct. with no. of
0470 ; good tries. If equal, you win.
0471 ; Try again.
0472 ; Set up for message 3.
0473
0474 ; Get character.
0475 ; Kill parity bit.
0476 ; Obey rule for Zapple
0477 ; Echó it.
0478 ; Was it 'n'? If so then.....
0479
0480 ; Yes, you did
0481 ; Have you won?
0482 ; Compare wrd ct. with no. of
0483 ; good tries. If equal, you win.
0484 ; Try again.
0485 ; Set up for message 3.
0486
0487 ; Get character.
0488 ; Kill parity bit.
0489 ; Obey rule for Zapple
0490 ; Echó it.
0491 ; Was it 'n'? If so then.....
0492
0493 ; Yes, you did
0494 ; Have you won?
0495 ; Compare wrd ct. with no. of
0496 ; good tries. If equal, you win.
0497 ; Try again.
0498 ; Set up for message 3.
0499
0500 ; Get character.
0501 ; Kill parity bit.
0502 ; Obey rule for Zapple
0503 ; Echó it.
0504 ; Was it 'n'? If so then.....
0505
0506 ; Yes, you did
0507 ; Have you won?
0508 ; Compare wrd ct. with no. of
0509 ; good tries. If equal, you win.
0510 ; Try again.
0511 ; Set up for message 3.
0512
0513 ; Get character.
0514 ; Kill parity bit.
0515 ; Obey rule for Zapple
0516 ; Echó it.
0517 ; Was it 'n'? If so then.....
0518
0519 ; Yes, you did
0520 ; Have you won?
0521 ; Compare wrd ct. with no. of
0522 ; good tries. If equal, you win.
0523 ; Try again.
0524 ; Set up for message 3.
0525
0526 ; Get character.
0527 ; Kill parity bit.
0528 ; Obey rule for Zapple
0529 ; Echó it.
0530 ; Was it 'n'? If so then.....
0531
0532 ; Yes, you did
0533 ; Have you won?
0534 ; Compare wrd ct. with no. of
0535 ; good tries. If equal, you win.
0536 ; Try again.
0537 ; Set up for message 3.
0538
0539 ; Get character.
0540 ; Kill parity bit.
0541 ; Obey rule for Zapple
0542 ; Echó it.
0543 ; Was it 'n'? If so then.....
0544
0545 ; Yes, you did
0546 ; Have you won?
0547 ; Compare wrd ct. with no. of
0548 ; good tries. If equal, you win.
0549 ; Try again.
0550 ; Set up for message 3.
0551
0552 ; Get character.
0553 ; Kill parity bit.
0554 ; Obey rule for Zapple
0555 ; Echó it.
0556 ; Was it 'n'? If so then.....
0557
0558 ; Yes, you did
0559 ; Have you won?
0560 ; Compare wrd ct. with no. of
0561 ; good tries. If equal, you win.
0562 ; Try again.
0563 ; Set up for message 3.
0564
0565 ; Get character.
0566 ; Kill parity bit.
0567 ; Obey rule for Zapple
0568 ; Echó it.
0569 ; Was it 'n'? If so then.....
0570
0571 ; Yes, you did
0572 ; Have you won?
0573 ; Compare wrd ct. with no. of
0574 ; good tries. If equal, you win.
0575 ; Try again.
0576 ; Set up for message 3.
0577
0578 ; Get character.
0579 ; Kill parity bit.
0580 ; Obey rule for Zapple
0581 ; Echó it.
0582 ; Was it 'n'? If so then.....
0583
0584 ; Yes, you did
0585 ; Have you won?
0586 ; Compare wrd ct. with no. of
0587 ; good tries. If equal, you win.
0588 ; Try again.
0589 ; Set up for message 3.
0590
0591 ; Get character.
0592 ; Kill parity bit.
0593 ; Obey rule for Zapple
0594 ; Echó it.
0595 ; Was it 'n'? If so then.....
0596
0597 ; Yes, you did
0598 ; Have you won?
0599 ; Compare wrd ct. with no. of
0600 ; good tries. If equal, you win.
0601 ; Try again.
0602 ; Set up for message 3.
0603
0604 ; Get character.
0605 ; Kill parity bit.
0606 ; Obey rule for Zapple
0607 ; Echó it.
0608 ; Was it 'n'? If so then.....
0609
0610 ; Yes, you did
0611 ; Have you won?
0612 ; Compare wrd ct. with no. of
0613 ; good tries. If equal, you win.
0614 ; Try again.
0615 ; Set up for message 3.
0616
0617 ; Get character.
0618 ; Kill parity bit.
0619 ; Obey rule for Zapple
0620 ; Echó it.
0621 ; Was it 'n'? If so then.....
0622
0623 ; Yes, you did
0624 ; Have you won?
0625 ; Compare wrd ct. with no. of
0626 ; good tries. If equal, you win.
0627 ; Try again.
0628 ; Set up for message 3.
0629
0630 ; Get character.
0631 ; Kill parity bit.
0632 ; Obey rule for Zapple
0633 ; Echó it.
0634 ; Was it 'n'? If so then.....
0635
0636 ; Yes, you did
0637 ; Have you won?
0638 ; Compare wrd ct. with no. of
0639 ; good tries. If equal, you win.
0640 ; Try again.
0641 ; Set up for message 3.
0642
0643 ; Get character.
0644 ; Kill parity bit.
0645 ; Obey rule for Zapple
0646 ; Echó it.
0647 ; Was it 'n'? If so then.....
0648
0649 ; Yes, you did
0650 ; Have you won?
0651 ; Compare wrd ct. with no. of
0652 ; good tries. If equal, you win.
0653 ; Try again.
0654 ; Set up for message 3.
0655
0656 ; Get character.
0657 ; Kill parity bit.
0658 ; Obey rule for Zapple
0659 ; Echó it.
0660 ; Was it 'n'? If so then.....
0661
0662 ; Yes, you did
0663 ; Have you won?
0664 ; Compare wrd ct. with no. of
0665 ; good tries. If equal, you win.
0666 ; Try again.
0667 ; Set up for message 3.
0668
0669 ; Get character.
0670 ; Kill parity bit.
0671 ; Obey rule for Zapple
0672 ; Echó it.
0673 ; Was it 'n'? If so then.....
0674
0675 ; Yes, you did
0676 ; Have you won?
0677 ; Compare wrd ct. with no. of
0678 ; good tries. If equal, you win.
0679 ; Try again.
0680 ; Set up for message 3.
0681
0682 ; Get character.
0683 ; Kill parity bit.
0684 ; Obey rule for Zapple
0685 ; Echó it.
0686 ; Was it 'n'? If so then.....
0687
0688 ; Yes, you did
0689 ; Have you won?
0690 ; Compare wrd ct. with no. of
0691 ; good tries. If equal, you win.
0692 ; Try again.
0693 ; Set up for message 3.
0694
0695 ; Get character.
0696 ; Kill parity bit.
0697 ; Obey rule for Zapple
0698 ; Echó it.
0699 ; Was it 'n'? If so then.....
0700
0701 ; Yes, you did
0702 ; Have you won?
0703 ; Compare wrd ct. with no. of
0704 ; good tries. If equal, you win.
0705 ; Try again.
0706 ; Set up for message 3.
0707
0708 ; Get character.
0709 ; Kill parity bit.
0710 ; Obey rule for Zapple
0711 ; Echó it.
0712 ; Was it 'n'? If so then.....
0713
0714 ; Yes, you did
0715 ; Have you won?
0716 ; Compare wrd ct. with no. of
0717 ; good tries. If equal, you win.
0718 ; Try again.
0719 ; Set up for message 3.
0720
0721 ; Get character.
0722 ; Kill parity bit.
0723 ; Obey rule for Zapple
0724 ; Echó it.
0725 ; Was it 'n'? If so then.....
0726
0727 ; Yes, you did
0728 ; Have you won?
0729 ; Compare wrd ct. with no. of
0730 ; good tries. If equal, you win.
0731 ; Try again.
0732 ; Set up for message 3.
0733
0734 ; Get character.
0735 ; Kill parity bit.
0736 ; Obey rule for Zapple
0737 ; Echó it.
0738 ; Was it 'n'? If so then.....
0739
0740 ; Yes, you did
0741 ; Have you won?
0742 ; Compare wrd ct. with no. of
0743 ; good tries. If equal, you win.
0744 ; Try again.
0745 ; Set up for message 3.
0746
0747 ; Get character.
0748 ; Kill parity bit.
0749 ; Obey rule for Zapple
0750 ; Echó it.
0751 ; Was it 'n'? If so then.....
0752
0753 ; Yes, you did
0754 ; Have you won?
0755 ; Compare wrd ct. with no. of
0756 ; good tries. If equal, you win.
0757 ; Try again.
0758 ; Set up for message 3.
0759
0760 ; Get character.
0761 ; Kill parity bit.
0762 ; Obey rule for Zapple
0763 ; Echó it.
0764 ; Was it 'n'? If so then.....
0765
0766 ; Yes, you did
0767 ; Have you won?
0768 ; Compare wrd ct. with no. of
0769 ; good tries. If equal, you win.
0770 ; Try again.
0771 ; Set up for message 3.
0772
0773 ; Get character.
0774 ; Kill parity bit.
0775 ; Obey rule for Zapple
0776 ; Echó it.
0777 ; Was it 'n'? If so then.....
0778
0779 ; Yes, you did
0780 ; Have you won?
0781 ; Compare wrd ct. with no. of
0782 ; good tries. If equal, you win.
0783 ; Try again.
0784 ; Set up for message 3.
0785
0786 ; Get character.
0787 ; Kill parity bit.
0788 ; Obey rule for Zapple
0789 ; Echó it.
0790 ; Was it 'n'? If so then.....
0791
0792 ; Yes, you did
0793 ; Have you won?
0794 ; Compare wrd ct. with no. of
0795 ; good tries. If equal, you win.
0796 ; Try again.
0797 ; Set up for message 3.
0798
0799 ; Get character.
0800 ; Kill parity bit.
0801 ; Obey rule for Zapple
0802 ; Echó it.
0803 ; Was it 'n'? If so then.....
0804
0805 ; Yes, you did
0806 ; Have you won?
0807 ; Compare wrd ct. with no. of
0808 ; good tries. If equal, you win.
0809 ; Try again.
0810 ; Set up for message 3.
0811
0812 ; Get character.
0813 ; Kill parity bit.
0814 ; Obey rule for Zapple
0815 ; Echó it.
0816 ; Was it 'n'? If so then.....
0817
0818 ; Yes, you did
0819 ; Have you won?
0820 ; Compare wrd ct. with no. of
0821 ; good tries. If equal, you win.
0822 ; Try again.
0823 ; Set up for message 3.
0824
0825 ; Get character.
0826 ; Kill parity bit.
0827 ; Obey rule for Zapple
0828 ; Echó it.
0829 ; Was it 'n'? If so then.....
0830
0831 ; Yes, you did
0832 ; Have you won?
0833 ; Compare wrd ct. with no. of
0834 ; good tries. If equal, you win.
0835 ; Try again.
0836 ; Set up for message 3.
0837
0838 ; Get character.
0839 ; Kill parity bit.
0840 ; Obey rule for Zapple
0841 ; Echó it.
0842 ; Was it 'n'? If so then.....
0843
0844 ; Yes, you did
0845 ; Have you won?
0846 ; Compare wrd ct. with no. of
0847 ; good tries. If equal, you win.
0848 ; Try again.
0849 ; Set up for message 3.
0850
0851 ; Get character.
0852 ; Kill parity bit.
0853 ; Obey rule for Zapple
0854 ; Echó it.
0855 ; Was it 'n'? If so then.....
0856
0857 ; Yes, you did
0858 ; Have you won?
0859 ; Compare wrd ct. with no. of
0860 ; good tries. If equal, you win.
0861 ; Try again.
0862 ; Set up for message 3.
0863
0864 ; Get character.
0865 ; Kill parity bit.
0866 ; Obey rule for Zapple
0867 ; Echó it.
0868 ; Was it 'n'? If so then.....
0869
0870 ; Yes, you did
0871 ; Have you won?
0872 ; Compare wrd ct. with no. of
0873 ; good tries. If equal, you win.
0874 ; Try again.
0875 ; Set up for message 3.
0876
0877 ; Get character.
0878 ; Kill parity bit.
0879 ; Obey rule for Zapple
0880 ; Echó it.
0881 ; Was it 'n'? If so then.....
0882
0883 ; Yes, you did
0884 ; Have you won?
0885 ; Compare wrd ct. with no. of
0886 ; good tries. If equal, you win.
0887 ; Try again.
0888 ; Set up for message 3.
0889
0890 ; Get character.
0891 ; Kill parity bit.
0892 ; Obey rule for Zapple
0893 ; Echó it.
0894 ; Was it 'n'? If so then.....
0895
0896 ; Yes, you did
0897 ; Have you won?
0898 ; Compare wrd ct. with no. of
0899 ; good tries. If equal, you win.
0900 ; Try again.
0901 ; Set up for message 3.
0902
0903 ; Get character.
0904 ; Kill parity bit.
0905 ; Obey rule for Zapple
0906 ; Echó it.
0907 ; Was it 'n'? If so then.....
0908
0909 ; Yes, you did
0910 ; Have you won?
0911 ; Compare wrd ct. with no. of
0912 ; good tries. If equal, you win.
0913 ; Try again.
0914 ; Set up for message 3.
0915
0916 ; Get character.
0917 ; Kill parity bit.
0918 ; Obey rule for Zapple
0919 ; Echó it.
0920 ; Was it 'n'? If so then.....
0921
0922 ; Yes, you did
0923 ; Have you won?
0924 ; Compare wrd ct. with no. of
0925 ; good tries. If equal, you win.
0926 ; Try again.
0927 ; Set up for message 3.
0928
0929 ; Get character.
0930 ; Kill parity bit.
0931 ; Obey rule for Zapple
0932 ; Echó it.
0933 ; Was it 'n'? If so then.....
0934
0935 ; Yes, you did
0936 ; Have you won?
0937 ; Compare wrd ct. with no. of
0938 ; good tries. If equal, you win.
0939 ; Try again.
0940 ; Set up for message 3.
0941
0942 ; Get character.
0943 ; Kill parity bit.
0944 ; Obey rule for Zapple
0945 ; Echó it.
0946 ; Was it 'n'? If so then.....
0947
0948 ; Yes, you did
0949 ; Have you won?
0950 ; Compare wrd ct. with no. of
0951 ; good tries. If equal, you win.
0952 ; Try again.
0953 ; Set up for message 3.
0954
0955 ; Get character.
0956 ; Kill parity bit.
0957 ; Obey rule for Zapple
0958 ; Echó it.
0959 ; Was it 'n'? If so then.....
0960
0961 ; Yes, you did
0962 ; Have you won?
0963 ; Compare wrd ct. with no. of
0964 ; good tries. If equal, you win.
0965 ; Try again.
0966 ; Set up for message 3.
0967
0968 ; Get character.
0969 ; Kill parity bit.
0970 ; Obey rule for Zapple
0971 ; Echó it.
0972 ; Was it 'n'? If so then.....
0973
0974 ; Yes, you did
0975 ; Have you won?
0976 ; Compare wrd ct. with no. of
0977 ; good tries. If equal, you win.
0978 ; Try again.
0979 ; Set up for message 3.
0980
0981 ; Get character.
0982 ; Kill parity bit.
0983 ; Obey rule for Zapple
0984 ; Echó it.
0985 ; Was it 'n'? If so then.....
0986
0987 ; Yes, you did
0988 ; Have you won?
0989 ; Compare wrd ct. with no. of
0990 ; good tries. If equal, you win.
0991 ; Try again.
0992 ; Set up for message 3.
0993
0994 ; Get character.
0995 ; Kill parity bit.
0996 ; Obey rule for Zapple
0997 ; Echó it.
0998 ; Was it 'n'? If so then.....
0999
1000 ; Yes, you did
1001 ; Have you won?
1002 ; Compare wrd ct. with no. of
1003 ; good tries. If equal, you win.
1004 ; Try again.
1005 ; Set up for message 3.
1006
1007 ; Get character.
1008 ; Kill parity bit.
1009 ; Obey rule for Zapple
1010 ; Echó it.
1011 ; Was it 'n'? If so then.....
1012
1013 ; Yes, you did
1014 ; Have you won?
1015 ; Compare wrd ct. with no. of
1016 ; good tries. If equal, you win.
1017 ; Try again.
1018 ; Set up for message 3.
1019
1020 ; Get character.
1021 ; Kill parity bit.
1022 ; Obey rule for Zapple
1023 ; Echó it.
1024 ; Was it 'n'? If so then.....
1025
1026 ; Yes, you did
1027 ; Have you won?
1028 ; Compare wrd ct. with no. of
1029 ; good tries. If equal, you win.
1030 ; Try again.
1031 ; Set up for message 3.
1032
1033 ; Get character.
1034 ; Kill parity bit.
1035 ; Obey rule for Zapple
1036 ; Echó it.
1037 ; Was it 'n'? If so then.....
1038
1039 ; Yes, you did
1040 ; Have you won?
1041 ; Compare wrd ct. with no. of
1042 ; good tries. If equal, you win.
1043 ; Try again.
1044 ; Set up for message 3.
1045
1046 ; Get character.
1047 ; Kill parity bit.
1048 ; Obey rule for Zapple
1049 ; Echó it.
1050 ; Was it 'n'? If so then.....
1051
1052 ; Yes, you did
1053 ; Have you won?
1054 ; Compare wrd ct. with no. of
1055 ; good tries. If equal, you win.
1056 ; Try again.
1057 ; Set up for message 3.
1058
1059 ; Get character.
1060 ; Kill parity bit.
1061 ; Obey rule for Zapple
1062 ; Echó it.
1063 ; Was it 'n'? If so then.....
1064
1065 ; Yes, you did
1066 ; Have you won?
1067 ; Compare wrd ct. with no. of
1068 ; good tries. If equal, you win.
1069 ; Try again.
1070 ; Set up for message 3.
1071
1072 ; Get character.
1073 ; Kill parity bit.
1074 ; Obey rule for Zapple
1075 ; Echó it.
1076 ; Was it 'n'? If so then.....
1077
1078 ; Yes, you did
1079 ; Have you won?
1080 ; Compare wrd ct. with no. of
1081 ; good tries. If equal, you win.
1082 ; Try again.
1083 ; Set up for message 3.
1084
1085 ; Get character.
1086 ; Kill parity bit.
1087 ; Obey rule for Zapple
1088 ; Echó it.
1089 ; Was it 'n'? If so then.....
1090
1091 ; Yes, you did
1092 ; Have you won?
1093 ; Compare wrd ct. with no. of
1094 ; good tries. If equal, you win.
1095 ; Try again.
1096 ; Set up for message 3.
1097
1098 ; Get character.
1099 ; Kill parity bit.
1100 ; Obey rule for Zapple
1101 ; Echó it.
1102 ; Was it 'n'? If so then.....
1103
1104 ; Yes, you did
1105 ; Have you won?
1106 ; Compare wrd ct. with no. of
1107 ; good tries. If equal, you win.
1108 ; Try again.
1109 ; Set up for message 3.
1110
1111 ; Get character.
1112 ; Kill parity bit.
1113 ; Obey rule for Zapple
1114 ; Echó it.
1115 ; Was it 'n'? If so then.....
1116
1117 ; Yes, you did
1118 ; Have you won?
1119 ; Compare wrd ct. with no. of
1120 ; good tries. If equal, you win.
1121 ; Try again.
1122 ; Set up for message 3.
1123
1124 ; Get character.
1125 ; Kill parity bit.
1126 ; Obey rule for Zapple
1127 ; Echó it.
1128 ; Was it 'n'? If so then.....
1129
1130 ; Yes, you did
1131 ; Have you won?
1132 ; Compare wrd ct. with no. of
1133 ; good tries. If equal, you win.
1134 ; Try again.
1135 ; Set up for message 3.
1136
1137 ; Get character.
1138 ; Kill parity bit.
1139 ; Obey rule for Zapple
1140 ; Echó it.
1141 ; Was it 'n'? If so then.....
1142
1143 ; Yes, you did
1144 ; Have you won?
1145 ; Compare wrd ct. with no. of
1146 ; good tries. If equal, you win.
1147 ; Try again.
1148 ; Set up for message 3.
1149
1150 ; Get character.
1151 ; Kill parity bit.
1152 ; Obey rule for Zapple
1153 ; Echó it.
1154 ; Was it 'n'? If so then.....
1155
1156 ; Yes, you did
1157 ; Have you won?
1158 ; Compare wrd ct. with no. of
1159 ; good tries. If equal, you win.
1160 ; Try again.
1161 ; Set up for message 3.
1162
1163 ; Get character.
1164 ; Kill parity bit.
1165 ; Obey rule for Zapple
1166 ; Echó it.
1167 ; Was it 'n'? If so then.....
1168
1169 ; Yes, you did
1170 ; Have you won?
1171 ; Compare wrd ct. with no. of
1172 ; good tries. If equal, you win.
1173 ; Try again.
1174 ; Set up for message 3.
1175
1176 ; Get character.
1177 ; Kill parity bit.
1178 ; Obey rule for Zapple
1179 ; Echó it.
1180 ; Was it 'n'? If so then.....
1181
1182 ; Yes, you did
1183 ; Have you won?
1184 ; Compare wrd ct. with no. of
1185 ; good tries. If equal, you win.
1186 ; Try again.
1187 ; Set up for message 3.
1188
1189 ; Get character.
1190 ; Kill parity bit.
1191 ; Obey rule for Zapple
1192 ; Echó it.
1193 ; Was it 'n'? If so then.....
1194
1195 ; Yes, you did
1196 ; Have you won?
1197 ; Compare wrd ct. with no. of
1198 ; good tries. If equal, you win.
1199 ; Try again.
1200 ; Set up for message 3.
1201
1202 ; Get character.
1203 ; Kill parity bit.
1204 ; Obey rule for Zapple
1205 ; Echó it.
1206 ; Was it 'n'? If so then.....
1207
1208 ; Yes, you did
1209 ; Have you won?
1210 ; Compare wrd ct. with no. of
1211 ; good tries. If equal, you win.
1212 ; Try again.
1213 ; Set up for message 3.
1214
1215 ; Get character.
1216 ; Kill parity bit.
1217 ; Obey rule for Zapple
1218 ; Echó it.
1219 ; Was it 'n'? If so then.....
1220
1221 ; Yes, you did
1222 ; Have you won?
1223 ; Compare wrd ct. with no. of
1224 ; good tries. If equal, you win.
1225 ; Try again.
1226 ; Set up for message 3.
1227
1228 ; Get character.
1229 ; Kill parity bit.
1230 ; Obey rule for Zapple
1231 ; Echó it.
1232 ; Was it 'n'? If so then.....
1233
1234 ; Yes, you did
1235 ; Have you won?
1236 ; Compare wrd ct. with no. of
1237 ; good tries. If equal, you win.
1238 ; Try again.
1239 ; Set up for message 3.
1240
1241 ; Get character.
1242 ; Kill parity bit.
1243 ; Obey rule for Zapple
1244 ; Echó it.
1245 ; Was it 'n'? If so then.....
1246
1247 ; Yes, you did
1248 ; Have you won?
1249 ; Compare wrd ct. with no. of
1250 ; good tries. If equal, you win.
1251 ; Try again.
1252 ; Set up for message 3.
1253
1254 ; Get character.
1255 ; Kill parity bit.
1256 ; Obey rule for Zapple
1257 ; Echó it.
1258 ; Was it 'n'? If so then.....
1259
1260 ; Yes, you did
1261 ; Have you won?
1262 ; Compare wrd ct. with no. of
1263 ; good tries. If equal, you win.
1264 ; Try again.
1265 ; Set up for message 3.
1266
1267 ; Get character.
1268 ; Kill parity bit.
1269 ; Obey rule for Zapple
1270 ; Echó it.
1271 ; Was it 'n'? If so then.....
1272
1273 ; Yes, you did
1274 ; Have you won?
1275 ; Compare wrd ct. with no. of
1276 ; good tries. If equal, you win.
1277 ; Try again.
1278 ; Set up for message 3.
1279
1280 ; Get character.
1281 ; Kill parity bit.
1282 ; Obey rule for Zapple
1283 ; Echó it.
1284 ; Was it 'n'? If so then.....
1285
1286 ; Yes, you did
1287 ; Have you won?
1288 ; Compare wrd ct. with no. of
1289 ; good tries. If equal, you win.
1290 ; Try again.
1291 ; Set up for message 3.
1292
1293 ; Get character.
1294 ; Kill parity bit.
1295 ; Obey rule for Zapple
1296 ; Echó it.
1297 ; Was it 'n'? If so then.....
1298
1299 ; Yes, you did
1300 ; Have you won?
1301 ; Compare wrd ct. with no. of
1302 ;
```

# THE PCW COMPETITIONS



## THE PCW MICROCHESS CHAMPIONSHIP

An Open Event for small machines with  
up to 32K of memory.

Full details from the address below.  
Entries accepted from all over the world.



**Best Software: £200 Prize**



**Best Homebrew System: £200  
Prize**



**Best School Application: £200  
Prize**



**Best Home Application: £200  
Prize  
plus The PCW Trophy in each  
category.  
plus Donated Prizes.**

FURTHER DETAILS FROM  
**COMPETITIONS  
PCW  
62A WESTBOURNE GROVE  
LONDON W2**

### TALKING TERMINALS?

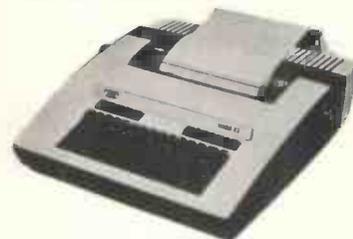
Talk to SAS Computer Products.  
We'll help solve your terminal needs.

#### SOROC IQ 120



Assembled \$825

#### TELEPRINTER 43



Model 4320 KP (RS-232) \$1377

#### LEAR SIEGLER ADM-3A



Kit \$695 Assembled \$795

#### SELECTRA-TERM



The ideal hard copy device with ex-  
panded features and greater reliabil-  
ity for your microcomputer system.

Assembled \$1750

For detailed information contact:

#### **SAS COMPUTER PRODUCTS, INC.**

7742 Redlands St., No. H3021  
Playa Del Rey, CA 90291 USA  
Phone(213)821-0642 Telex18-1059

#### **European Sales and Service**

SAS Computer Products, Inc.  
Badenerstrasse 281  
CH-8003 Zurich, Switzerland  
Phone 01/241 49 05 Telex 58738

OEM and dealer discounts. All  
prices FOB Los Angeles, CA, USA

# Getting It Together

MIKE BANAHAN

This is the first of a series of articles about assembly language and assemblers. In this month's article we take a look at the reasons for having assemblers, and see one form of this type of language; later articles will go more deeply into how an assembler works and will describe the construction of an assembler for the Z-80. 6800 users don't despair! The techniques are common to all assemblers, so although the Z-80 assembler won't run on your machine there's no reason for not using the principles outlined to translate the major parts into code for almost any processor. Indeed I hope very much that there will be people interested enough to start writing their own assembler. Make no mistake, it isn't an easy task and will not be a small program. The Z-80 assembler took 2 months of evenings to knock up and needs nearly 7K to run in, but it's great fun to have all the freedom that your *own language* permits. What's more you can modify it to suit your own requirements as and when you please. For example, there are the Z-80 "hidden" instructions that the manufacturer doesn't tell anyone about — but nonetheless happen to work. If you want to use them then no problem — just stick the necessary "mods" into the assembler and there they are.

First off though, we'll start right at the very beginning and ask just what is assembly language, and see how it differs from other computer languages.

## What is Assembly Language?

Assembly language results from the problems to be found playing the "low level" programmer. (Not that low level is a derogatory term used to describe the quality of the poor soul doing the programming: it's used to show that the programming is being done at the processor instruction level rather than in a language like Fortran or Basic.)

Low level programming has both advantages and disadvantages when compared with a high level language and there's no point in starting the arguments all over again here. The differences are that ideally you should never know what computer is running a program written in a high level language and the program should run just as well on a different machine. That's one of the reasons for having a high level language. Assembly language, on the other hand, is specifically for one computer and will depend entirely on the instructions the machine is capable of executing. Very few hobbyists will be writing in much other than assembler code, so it helps to know something about it!

The processor will be working with instructions and data which are strings of binary ones and noughts, this being fine for the processor since it isn't easily bored, but not so good for the human being who has to write it all down. The first step away from utter mental distraction is to turn the binary representation into a more familiar form — though still just a string of numbers. At first glance it might seem sensible to convert the binary into its decimal equivalent, but in practice two other forms of representation are used: Hexadecimal and Octal.

It's often important for the low level programmer to know whereabouts the ones and noughts are and by using hexadecimal (base 16) or octal (base 8) this can easily be done. For example, the 16-bit binary number 1101001010111111 can be converted into hex (the usual abbreviation for hexadecimal) by breaking it into 4-bit chunks and counting.

0-9, A-F:

1101 = d  
0010 = 2  
1011 = b  
1111 = f

giving D2BF.

The less common octal form is found by taking 3-bit slices:

1 = 1  
101 = 5  
001 = 1  
010 = 2  
111 = 7  
111 = 7

giving 151277

It's quite possible to write programs in hex or octal; indeed that's the only form that some programmers use, but as soon as programs get fairly long then problems crop up.

Imagine, if you will, that you have just finished writing a program in hex which contains a number of jumps from one part of the program to another. Since the *address to which* the processor *jumps* is an essential part of the instruction, you will have had to keep count of the addresses used as you went. Well yes, it can be done, fair enough. But then horrors! There's a mistake — an extra instruction has to be inserted in the middle. All the addresses (i.e. numbers) above it *change* and all of the jumps to instructions above the insertion have to be *changed* . . . . So you go and take up woodwork instead. Or write a program to do it for you. This is one of the main jobs of an assembler: KEEPING TRACK OF ADDRESSES.

Your program could now become

```
INSTRUCTION  1
              2
              3
              300
              LABEL 1 301
              JUMP TO LABEL 1
```

And you no longer have to worry about addresses if the program needs altering. "LABEL 1" does the job for you.

Although writing programs in hex or octal is an improvement on binary, it's still not easy to remember whether subtraction (say) is represented by D6 or C6. So the program that already looks after those addresses for you gets given a *new* job. It's *told to convert* SUBTRACT into D6. This is the other major job of an assembler: it allows the use of MNEMONICS (coming from the Greek for impossible to pronounce).

Suddenly everything has become much easier. Instead of writing (for example) 3A071EC32142, we can write

```
LD A,1E07 (load accumulator from address 1E07)
JP 4221 (jump to address 4221)
```

```
or, using labels,
LD A LABEL 1
JP LABEL 2
```

as long as Label 1 and Label 2 are *somewhere else* in the program. It's usual to describe the LD and JP as *opcodes*, because they describe the *operation* of the instruction; the *operands* (A, LABEL 1, etc) are what follow the opcode. (Note: 3A = LD and 071E is the way of writing the address 1E07; C3 = JP and 2142 is the way of writing the address 4221).

There then are the rudiments of an assembly language. Its purpose is to make programming at machine code level easier, by providing mnemonics for instructions and to allow the use of labels so that the programmer doesn't have to worry about addresses for

jumps and data storage. Most assemblers provide more than just these basic facilities, but they are only icing the cake — not putting in the nuts and raisins.

**How does it do it?**

I hear you cry. Well, as long as you will let the assembler have *two goes* at getting the right answer, then it's easy. Or, put another way, the most common approach is to read the input text (often called the "source code") twice. The first time through, the assembler builds up a table of all the labels used in the program and keeps a note of the address they occur at. When it reaches the end of the source code it starts again, but this time it produces the output (object code). Every time a label is referenced, the assembler *looks for the label in its symbol table*, and inserts the required address.

Dealing with labels is the easy part of the job though, and it's the other stuff — those mnemonics — that lead to heartache and a reawakening of interest in chisels and glue in our budding assembler writer. If the mnemonic is simple and unambiguous in its definition then that's easy to deal with. If on the other hand, it's anything like the Z-80 LD instruction, hours of amusement/self-torture result from trying to unravel the problem. Depending on the operands following the opcode, the instruction may turn out to be between one and four words (of memory) long, and there are 130 different combinations of operands! The flowchart for that little lot is of astonishing proportions and looks very like a plate of spaghetti — which has nothing to do with the fact that the designer of the Z-80 is Italian!

As a byeline, it's interesting to note that even Zilog seem to be embarrassed by LD. Quite rightly too, in my opinion. After all, what is LD A,B supposed to mean? In the programming manual they feel obliged to explain it by writing underneath the instruction  $A \leftarrow B$  which at least begins to suggest movement of something from B to A. (In fact it means load the accumulator from register B). We have just decided that the whole point of having an assembler is to make things easier for the programmer — and then Zilog recommend that an almost meaningless mnemonic is used. If  $A \leftarrow B$  describes the operation, then why not use  $\leftarrow$  as the mnemonic for loading? It's certainly possible to do just that, as the last part of this article demonstrates!

**Ice that Cake**

Apart from dealing with labels and instructions, the assembler can be made even more useful by providing extra facilities, notably the allocation of space for data, and the ability to build up tables, text strings and the like. These extra facilities are called up by "assembler directives", which are really instructions to the assembler. Don't confuse these directives with instructions to the processor because that's not what they are.

**The Real Thing**

Now, the reward for anyone patient enough to have read so far! Some real assembly language, in this case for the Z-80. It isn't in the standard Z-80 format, but that doesn't prevent it from demonstrating several of the principles mentioned before. It's unusual in some respects — notably in that it allows the use of several instructions on one line of text. There's no reason why conventional assemblers should not do the same, except laziness on the part of whoever writes them, and in some cases it provides a useful bonus of being able to write short loops in the program all on one line. This is especially easy in the case of the particular assembler in question, for there is an implied label '.' at the start of every line. The example should make that a little clearer.

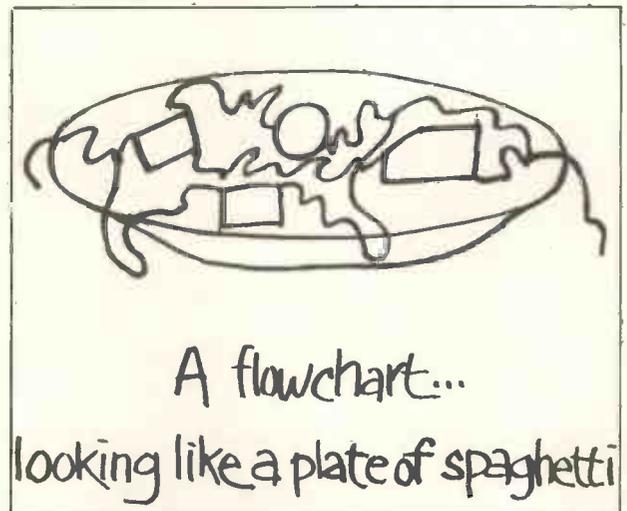
One final point — which usually confuses beginners to assembly language programming — is the distinction between an address and the *contents* of that address.

They're both just numbers, and easily mixed up; especially if you're not used to the difference. An address specifies a place in memory — the *contents* of the address are what is to be found *at* that location. Think about it. It would perhaps be easier if addresses came like the instructions for finding something on a row of shelves. "Third on the right, fifth from the bottom", maybe. They don't and that's that, so hard luck.

```
[COMMENT IN THE INPUT TO THE
[ASSEMBLER IS INDICATED BY OPEN
[SQUARE BRACKETS LIKE THE ONES AT THE
[BEGINNING OF THESE LINES.
[END OF COMMENT IS EITHER CLOSE
[BRACKET]
[OR ACTUAL END OF LINE.
[THE ACTUAL INSTRUCTIONS START HERE
.= 400; [SET CURRENT ADDRESS. THIS DIRECT-
[IVE
[IS AN INSTRUCTION TO THE ASSEM-
[BLER NOT THE Z-80.
BEGIN: SP<- .; [LOAD STACK POINTER WITH
[VALUE OF '.'
HL<- TABLE; [LOAD HL REGISTER
[PAIR WITH START
[ADDRESS OF TABLE
CLA; [CLEAR ACCUMULATOR
B<- 100; [LOAD REGISTER B WITH
[NUMBER 100
(HL)<-A; INC HL; DBNZ.;
[MOVE CONTENTS OF A TO LO-
[CATION
[WHOSE ADDRESS IS IN THE HL
[REGISTER, INCREMENT THE
[ADDRESS, DECREMENT THE B
[REGISTER AND BRANCH TO '.'
[IF THE RESULT IS NOT ZERO
TABLE: .BLKW 100; [TWO THINGS HERE:
[A LABEL AND AN ASSEMBLER DIREC-
[TIVE
[.BLKW 100 ALLOCATES 100 WORDS OF
[STORAGE
.END; [ANOTHER DIRECTIVE. THIS ONE
[TELLS THE ASSEMBLER THAT IT HAS
[READ ALL THE INPUT TEXT.
```

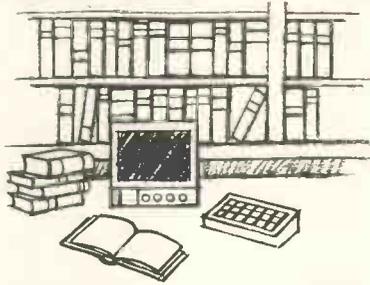
There it is. Real assembly language. There isn't a lot of it, but there's much more to come next month and there is enough above to give you some idea of what it looks like. If you think you're going to enjoy writing your own assembler, then my only suggestion (except that you keep on reading this series) are to cancel your holidays this year and start buying memory now. You'll need all the time and space you can find!

(PCW Next issue: more explanation before going into the hairy details. PCW)



# PCW Book Review

MICHAEL JAMES



## SCELBI "6800" SOFTWARE GOURMET GUIDE AND COOKBOOK.

R. Findley, 1976; 210 pages.  
(Scelbi Computer Consulting Inc, 5½" x 8½" £7.95).

One of the interesting features of this book is the copyright notice: "Other than using the information detailed herein on the purchaser's individual computer system, no part of this publication may be reproduced . . . etc. etc." As some of the programs to be found in this cookbook are very elementary, eg. a simple memory clear program, enforcing the copyright might be a little difficult. It's rather like copyrighting a BASIC "FOR" or a FORTRAN "DO" loop!

This said, the Scelbi's 6800 cookbook is very useful for anyone doing assembly language programming at either an elementary or advanced level, which is something of an achievement in itself. The first two chapters comprise the usual introduction to programming the 6800. Following a boring but necessary explanation of the instruction set, chapter two deals with some obvious but perhaps too often overlooked techniques of programming. Chapter three is the first of the cookbook chapters with general purpose routines for clearing memory, delays, random number generation, etc. . . Chapter four is more specialised, dealing with code conversion, ASCII to BAUDOT, and the first of the numerical routines BINARY to and from DECIMAL (BCD). Chapters five and six are the most useful to the advanced programmer. Chapter five gives details of a floating point maths package, and chapter six a complete decimal fixed point package. For these two items alone the book pays for itself. A floating point package for the 6800 costs between £10 and £12 and the Scelbi cookbook provides this and a lot more. Following the maths packages, chapter seven on input/output, and chapter eight on search and sort routines, come as something of an anticlimax. It's almost impossible to write anything useful on I/O processing because once the ideas of "handshaking" and interrupt control are understood, general principles are a bit thin on the ground. This is presumably why I/O problems are not a favourite area for academic computer scientists. However chapter seven manages simple and adequate descriptions of I/O from the software point of view. Conversely, searching and sorting is an academic's paradise with more ways of doing the same job than in any other area of programming. In fact some programmers will tell you that programming *is* searching and sorting! The Scelbi cookbook gives a good practical introduction to this subject without being at all academic.

Generally the text is well written and the programs well laid out. Each program is explained using a clear flow diagram which would make the book useful even for non-6800 owners.

As an instructional book the Scelbi cookbook is good. As a piece of software it's very good.

## PCC's HOME REFERENCE BOOK of PERSONAL and HOME COMPUTING,

D. McCabe, 1977; 248 pages.  
(Peoples Computer Company, paper 7" x 9" £4.95).

The idea of a home computer reference book is, of course, a good one. However the PCC Reference Book is not quite as useful as it seems at first glance. Only about one third of the book is dedicated to references, the rest being general articles

and a large number of advertisements. The useful reference section is reduced even further when it is realised that most of the addresses of computer clubs and stores are on the wrong side of the Atlantic for the average UK reader. There are only fifteen pages of references that are of general use. However, they are so useful that they nearly justify the book's existence in themselves. A general bibliography and a computer music bibliography are included and form a good guide to further reading. The most important pages contain a well-arranged index to all the articles in the major computer magazines to about March 1977. These make finding the location of all the information on a specialised topic relatively easy if the appropriate magazines can be found. (Does anyone know of a library taking BYTE, etc?)

The articles section contains some very enjoyable essays, along with a short glossary of computer terms and a listing of Tiny Basic for the 8080 or the Z80. For anyone thinking about writing an interpreter or just interested in how one works, this well-commented 3k BASIC listing makes slow but educative reading. (As computer hobbyists read programs as well as books perhaps PCW should review program listings as well?)

The rest of the articles are easy-to-read introductions to and discussions of various aspects of computing. None of them are "how to do it" articles and hence they are enjoyable rather than useful.

It is surprising to find so many advertisements in a book. Indeed it is surprising to find any! But the PCC home reference is full of them. There are two ways to react to this — either complain vigorously about the lost text, or consider it a section as valuable as an other sort of reference. My reaction is that knowing what is for sale in the US and at what price is something which is *not* out of place in a reference book and I enjoyed reading the ads. as much as the articles.

In conclusion, the PCC Home Reference is more of an enjoyable read than an informative reference book. Although one or two of the articles are basic introductions to home computing, the absolute beginner would do better with a more complete (i.e. lengthy) start.

The PCC reference book is reasonable value in terms of printed page per penny, but there is still room for a good reasonably priced computer reference book.

## TV TYPEWRITER COOKBOOK

D Lancaster, 1976; 256 pages. (H.W. Sams, paper, 5½" x 8½" £7.40).

This is a book for hardware experts and electronics addicts. It covers a wide range of topics not indicated by its title, eg. serial interfaces, UARTS, modems, cassette interfaces, colour TV, Selectric typewriters, Teletypes © etc., etc. Because of this diversity it has become one of my most used books.

Don Lancaster writes well, and is easy to read. His casual style makes the operation of his circuits seem easy to understand, but the reader may find that a certain amount of working out of the 'fine detail' is necessary to understand them fully.

To be able to construct a TV Typewriter (TVT) from this book, the reader would have to know something about electronics beforehand (TTL/CMOS logic design in particular). Most of the circuits are more than design notes but a complete TVT circuit diagram is not given and this some interpretation is necessary.

Some ingenious methods are explained and quite a few CMOS design 'tricks' can be learned by examining Don Lancaster's circuit diagrams. Whether they are all worth using is something that is left to the reader. For example, I was well into the construction of my own TVT when I came across this book, and as a result I redesigned the timing generator along the lines suggested by the phase locked economy circuit on page 96 (using TTL), but I did not use the McFadden cursor on page 111. By making such choices and modifications the reader should find it possible to construct a TVT to suit his pocket and application.

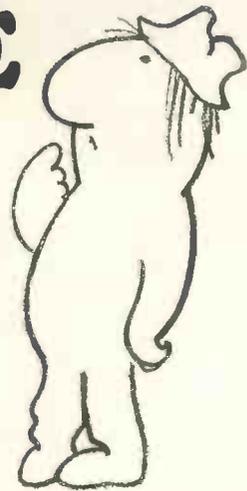
There are a number of ways that a TVT can be acquired by the amateur. The most expensive is to buy a commercial unit (always referred to as a VDU). The next is to buy a kit. The cheapest is to construct a TVT using either published designs (at least three have appeared in UK electronics magazines) Designing from scratch is not an easy project and certainly should not be attempted without this book. Even if a TVT is acquired by one of the easier methods, the TVT Cookbook would make a good companion when making a choice between the different designs, kits or even models.

The TVT Cookbook can be recommended on all counts but for a small paperback £7.40 is a bit expensive.

# PCW OPEN PAGE

## The Amateur Computer Club View

Mike Lord



### Meeting Points

In response to popular request, all amateur computing groups known to the author are listed below, plus the names and addresses of people who have offered to organise new groups during the last few months. Although an attempt has been made to collate as comprehensive a list as possible, inevitably some will have been overlooked, in which case please accept our apologies and send details to be included in a future list.

**Thames Valley Group;** started recently, they have been organising meetings at Reading University. Further details can be obtained by sending an s.a.e. to Bob Cottis, 'Pippins', Boulsters Lane, Maidenhead, Berks SL6 8JT, or by calling him on Maidenhead 22445.

The Cambridge University Processor Group is open to anyone, whether or not they are members of the university, and details of their activities can be obtained from Tim Hopkins, Cambridge University Processor Group, Magdalene College, Cambridge CB3 0AG.

Two people have recently offered to set up groups in the South West of England; Mr. G.V. Barbier, Palmers Mill, Calverleigh, Tiverton, and Mr. D. Carne of 44 George St., Exmouth (telephone 039 52 74479).

Coventry boasts a very active ACC group, and details may be obtained from Ray Diamond at 27 Loweswater Rd., Coventry CV3 2HJ (telephone Coventry 454061).

Not connected with the ACC, but still worthwhile (!) is the Vauxhall Community Computing Centre. For further details write to 132 South Lambert Rd., London SW8

A meeting point for Cornish bit bashers is the Personal Computing Club, c/o The Micro-B Computer Store, 22 Lemon St., Truro. Tel Truro 78487

Mr. P.A. Gibson-Daw of 479 Wellingborough Rd., Abingdon Park, Northampton NN3 3HN (tel; 714821) has offered to help set up a new Computer Users Group for enthusiasts in that area.

PET owners living in the North East should take note of Mr. T. Turnbull's offer to start an ACC branch to help people disseminate software and hardware information for this machine.

Mr. John Peacocks, of 22 Wheatfield Gdns, Belfast 14 would like to contact other amateur computing enthusiasts in his area, with the intention of setting up a regular local group.

Mr. A. Aylward, of 194 Balmoral Road, Gillingham, Kent, would also like to start a local microprocessor group.

The ACC Northwest Group is now well established and holding regular meetings at the National Computing Centre, Oxford St., Manchester. Contact man is Ken Horton, 50 Lymfield Drive, Worsley (061-799-0192)

Computer Hackers in Central Scotland interested in meeting people with similar interests should get in touch with Stewart Stevenson 'Lindisfarne', New Well Wynd, Linlithgow, West Lothian (telephone Linlithgow 2657).

For some reason Cheltenham seems to have an unusually high number of ACC members. Whatever the reason, the local group there can be contacted via Mr. M.P. Pullin at 45 Merestones Drive, The Park, Cheltenham. tel; 25617.

The Southampton University Computer Club have a NASCOM and are also building a 9900 based system; further details may be obtained from Paul Maddison, 47 Westridge Rd., Portswood, Hants.

For any readers who are contemplating starting a local group, may I offer a few tips passed on by those who have taken the plunge;

— Do it. You've nothing to lose and a lot of pleasure to gain.

— If you're stuck for a meeting place, try your local college. Most of these have well equipped lecture rooms of various sizes, and many now have computer access facilities that you may be able to use. It helps in some cases if you can present the educational value of the group and, in theory at least, you may be able to persuade the authorities to set up a series of weekly 'evening classes' with a paid lecturer. But do it quickly, before the education committees finalise their arrangements for the next educational year (beginning in September).

— For the first couple of meetings at least ask people to come along with their systems and give a brief talk about them; a series of short semi-formal presentations is much better than just letting people find their own way around a number of systems scattered about the room.

— Advertise widely to get as many people as possible along to the first few meetings; attendance is bound to drop off later. This means planning ahead and informing the monthly press (e.g. PCW) at least 8 weeks before the event. For those with the audacity, a call to the local press is guaranteed to bring a reporter round eager to write up the latest local eccentricity.

— Enjoy yourself.

Details of the Amateur Computer Club (now the oldest living amateur computing club in the world!) may be obtained from Mike Lord, 7 Dordells, Basildon,

Essex. He would also welcome news of new and existing local groups for mention in this column.

#### U.K. PET USERS CLUB NEWSLETTER

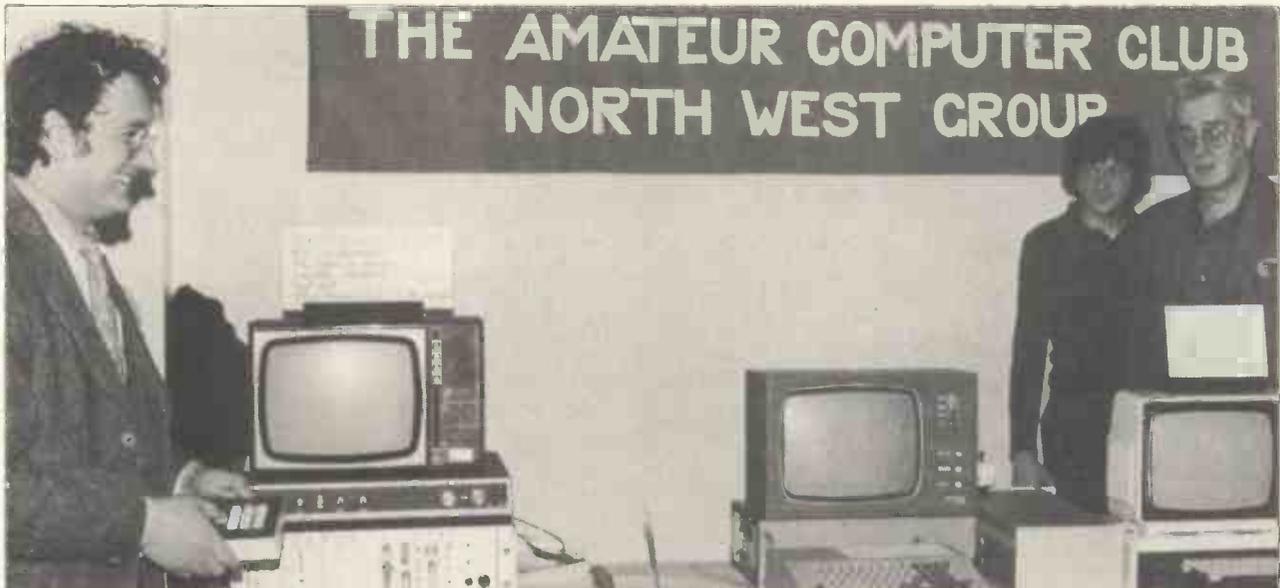
Welcome to the U.K. PET Users Club.

Printed in all issues of the PET newsletter will be a list of software available through the Users Club. The original list is small and Commodore Systems generated, but we hope to add many programmes in the near future. We intend to form a "Common Library" which will contain programmes submitted by PET users and we hope you will contribute. The contents of both Libraries will be printed in each issue.

For each of your programmes accepted, you will be given the choice of up to three from the Common Library, free of charge or you may purchase any programme from the Common Library for £1.00 per programme plus 30 pence for p+p. We also hope that you will write to us telling us of your applications for and your experiences with the PET. We will endeavour to publish as many of your letters and comments as we can and if anyone feels like submitting a small article, this will be most welcome. If you require a personal reply this will be guaranteed by enclosing a self addressed, stamped envelope.

The address to which you send your correspondence or orders is:

The Software Manager,  
Commodore Systems Division,  
360, Euston Road,  
London, N.W.1.



The N.W. group of the ACC is thriving. Photo is of a stand they had at a recent radio rally at Manchester's Belle Vue. Thanks to D. Wade, 26 Wolsey Close, Radcliffe, Manchester, for his energy in send us this photo. Readers of the Open Page: it's for you!

## SIRTON PRODUCTS SP



### KEYBOARD

7 Bit Parallel Output and Strobe  
TTL Compatible — requires 5V @ 100mA  
Qerty or Alpha/Numeric Block Face  
+ Kit price (including case) £ 38.50

### BOARD KITS

8080 CPU Board with jump on reset complete with sockets etc. £106.75  
8K RAM Board low power 450 n Sec. £118.50  
(21L02-1—  
Cassette Interface Board 187 to 540 Bytes/Second — built & tested £121.55  
PROM Board for 8K EPROM 2708 with Integral Programmer £118.50  
S.100 Motherboard 10/11 slot with Active Terminations and 4 edge connectors £ 58.95  
NEW Video Interface Board with Case & Power Supply £ 85.50

### HARDWARE

S.100 edge connectors gold plated solder tail £ 5.45  
Transformer Pri 110/240 V; sec. 8 V @ 10 Amp & 25 V CT @ 2 Amp £ 12.25  
Bridge Rectifier with integral bracket rating 18 Amp £ 3.95

Ask also for our Lists of 74, 74LS, TTL and CMOS IC's, Microprocessors and Support Chips.

+ Items thus market also available ready built and tested at £20 extra.

Please add 8% VAT.

**NEW PRODUCTS BEING ADDED TO OUR RANGE EACH MONTH — WRITE OR PHONE FOR LATEST LISTS, PRICES AND DISCOUNTS**

### SIRTON PRODUCTS

13 Warwick Road, Coulsdon Surrey CR3 2EF.  
Tel: 01-660 5617 (day and evening)

### Post & Packing

Kits:	80p per kit	Keyboard:	£1.00
Transformer:	£1.00	Hardware:)	30p per order
		IC's: )	



# Owner's Report

## THE TRS-80

A.J. HARDING

The TRS-80 is one of the first of the new breed of microcomputers which is now becoming available to the home and small business user, known as "appliance" systems because they are delivered complete with a built in firmware language and are ready for use immediately the power is switched on.

The TRS-80 is manufactured by the Tandy Corporation, an American company which trades as the "Radio Shack" in the United States. The first 200 units arrived in England in April 1978 and I understand that this first batch was sold very quickly. Various articles have appeared reviewing the computer but it was thought that one which relates an owner's experience of it may be of assistance to people who are considering its purchase.

In England, the TRS-80 (Figure 1) is delivered complete as four separate pieces of equipment: the keyboard, which also contains the computer board itself, a video monitor, a cassette tape recorder for programme and data storage and the power supply. In the United States

the monitor and the cassette need not be purchased, enabling the user to save money by using any such equipment which he already owns. Although this may be possible in England, there is no doubt that, at least initially, the sales drive is towards selling the complete system.

Although it is rather paradoxical in an appliance system to buy only parts of the system, there is little doubt that the monitor is over-priced. In the U.S. where 12" portable televisions can be readily bought for under \$100, it costs \$199.96, so substantial savings can be made. It is not known whether Tandy will go the separate route in England but if it does, it is certainly one which should be considered. The other side of this particular coin of course, is that one of the advantages of buying all of the parts together is that it is *guaranteed* as a system. What Tandy's policy is when, say, only the keyboard is purchased and a fault develops in it which may be the result of a malfunctioning cassette already

owned by the customer, one hates to imagine! In any event, so far as I know, the piecemeal approach is not available here as yet, so the point is academic.

As a matter of interest the prices in the U.S. are: TRS-80 complete is \$599.95, the keyboard including the computer is \$399.95, the cassette is \$49.95 and the video display is \$199.95. Hence the customer saves about \$50 by buying the complete unit, as compared to the sum of the individual prices. In England the price of the system is £499. At an exchange rate of \$1.90 to the pound, this means a differential of about £180, which is not as bad as some in this day and age, when it is not uncommon to see the same figure price on both sides of the Atlantic for the same article, even though one is in dollars and one in pounds. If it is any consolation, the price differential between the U.S. and Canada is roughly \$300, which is large when one considers that there is only about 12% difference on the different dollars.

The system comes in three cartons, one containing the cassette, the second the keyboard and finally one containing the video monitor and power supply. Also included are all necessary connecting cables and an Operating Manual. For some reason best known to Tandy, at least one of the cartons had been flown to England from the United States *via* a stop in Belgium but despite this somewhat circuitous route no transit damage was evident, no doubt due to the entirely adequate plastic foam packing used. Interconnection between the four parts of the system proved to be no problem at all. Even without the manual instructions it is fairly obvious where everything goes.

It is a pity that Tandy have sought to economise in the type of connectors used. Five pin DIN plugs are used for the three connections to the keyboard unit (Figure 2) and if Murphy's law is to be believed, someone, sometime will insert them incorrectly. Probably exchanging the tape and video connections would not do much harm but connecting the power supply to the video output might well cause some trouble! However, at least these connections are firm and instil some confidence in the user but unfortunately the same cannot be said of the connections to the cassette. As a normal production audio cassette is used, the 3.5 mm and 2.5 mm miniature plugs usually associated with lower cost cassettes are incorporated. The smaller plug is the remote connection to the tape, by which the computer switches it on and off.

On my unit the connection is intermittent and the plug has to be in just the right position for a proper connection. Unfortunately, this is the plug which has to be removed and reinserted when it is required that the tape be manually rewound, a function which the computer cannot perform. This results in the troublesome plug being frequently in use. Checking that proper con-

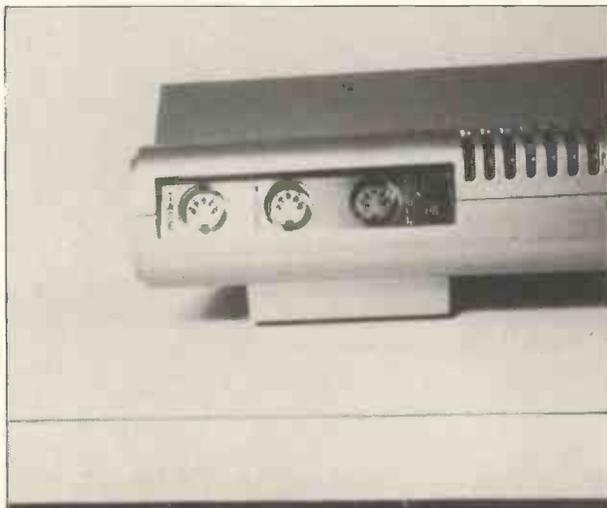
tact is being made every time that a rewind is performed, is to say the least, inconvenient. The solution is the small modification shown in Figure 3 and consists of a miniature toggle switch which takes over the job of the jack switch, which is activated when the plug is inserted or removed. One word of warning however, the installation of this switch probably voids the warranty on the cassette, although one would hope, not on the rest of the system.

To install the switch remove the back of the cassette by unscrewing the three obvious screws plus the two hidden away in the battery compartment. The printed circuit can be removed but it is not essential because if you look along an inch or two from the set of jacks, two white wires will be seen connecting to the component side of the board. One goes to a ground island on the printed circuit pattern and the other to a lug, the other side of which is connected to a yellow wire. An SPDT switch is used for the modification. Before mounting the switch, remove the yellow wire mentioned and solder it to the centre switch lug. Drill a hole in the plastic case and mount the switch with about four inches of fine wire attached to each outside lug of the switch. After mounting, one of these wires is soldered to the point from which the yellow lead was unsoldered and the other goes to the ground island. The switch will now take over the action from the jack switch and it is no longer necessary to remove the remote control plug at all.

A better modification of course, would be to install the switch as described and then *permanently* wire the three leads which go to the cassette and do away with the plugs and jacks altogether. Tandy are going to introduce a disc storage system to replace the cassette however, so if it is anticipated that this will be purchased, the permanent modification on the cassette would mean that it could not be used as an ordinary cassette later.

Whether or not a user likes the four individual system parts, as compared to other computers, such as the PET, wherein everything is one enclosure, is purely a matter of taste. The interconnecting cables can be a nuisance but on the other hand, the single enclosure is inevitably quite large and cumbersome for desk use. Personally, I think that the advantage of being able to move the keyboard aside easily to make room for the inevitable scribing chores associated with programming, far overrides any aesthetic considerations. Incidentally, a spot of emery cloth rubbed over the rubber feet on the keyboard bottom, works wonders for its stability.

Once one has re-checked the interconnecting cables and plugged into the mains, after changing the U.S. and Continental plugs supplied over to 13 amp ones (but using 3 or 5 amp fuses), the system can be switched on. Neither the switch on the monitor nor the one on the keyboard disconnect the transformer primary from the



mains, so it is better that all mains connections are made to switched outlets, in order that the outlet switch may be used to totally disconnect power, when the equipment is not in use. Otherwise the transformer will be continuously warm and, of course, a small amount of power will always be draining away. The same remarks apply to the cassette. After the application of power the word "READY" will appear in the top left hand corner of the monitor screen together with a prompt sign and the cursor underline. This means of course, that the BASIC is immediately available and the machine is, as it says, ready to go.

Although such efficiency and convenience is to be admired, it does have one very great drawback, namely that there is no way of dropping into machine language *before* the BASIC interpreter takes over. Parenthetically, what a hard to please lot we are! Ever since home computers became available we have complained (or rather some of us have) about manipulating machines with toggle switches in machine language, or having to load paper or magnetic tapes before we can use a higher level language. Now it is all laid on and the first thing I comment on is that machine language is not available!

It is not the purpose of this article to take the reader through all of the specifications of the TRS-80. These are available from Tandy or from previous review articles on the computer which have appeared in various magazines. However, a brief outline is necessary, so that we know what we are talking about.

The TRS-80 in its £499 version is supplied with 4K of ROM, mainly taken up with the BASIC interpreter and 4K of RAM, of which only 3583 bytes are available to the user for programming. At the present time, the RAM can be increased to 16K by the purchase of an additional 12K for £229. Also an improved BASIC, called Level 2, will be available in July 1978 for £79. The Level 1 BASIC supplied is similar to Tiny Basic and contains only the barest bones of the language. There is no doubt that Level 2 is almost essential for serious work. For instance, although two string variables (A\$ and B\$) are allowed, they cannot be manipulated in any way. It is a pity that at least a comparison of them is not included in Level 1; in other words the command IF A\$ = B\$ THEN ..... is illegal and results in an error. Most of the other important BASIC commands and statements are included. One array is available and 26 numerical variables.

An important trade off, bearing in mind the amount of RAM available, is that a Shorthand Dialect is permitted for most commands and statements, so that PRINT becomes P., GOTO G., NEXT N. and so on. In a fair length programme this can add up to a saving of many hundreds of bytes. Consequently if a decision of whether Level 2 or more RAM should be purchased arises, the user would be well advised to opt for the Level 2 first.

Unfortunately, nothing is yet known about Level 2 and Tandy are unable to supply even a list of the commands and statements included. As they insist on a deposit in order to assure delivery in July, this seems a little unfair and can be likened to buying a "pig in a poke". The market is certainly a seller's one at the moment! Local managers of Tandy (at any rate certainly mine in Eastbourne) are *very helpful*, within their capabilities in a new field, but apparently experience difficulty in extracting information from Head Office. It must be said in fairness however, that at least the TRS-80 *has been delivered to the market place*. If this necessitated some details getting lost on the way, then it is preferable to simply being on a waiting list! In the U.S.A. waiting lists of months are not uncommon for new micro computers.

The video display has a 12" screen (30 cm diagonal) which is a nice size and is preferable to the 9" screen which some micros have. The characters are legible al-

though sometimes some judicious spacing is advisable. Automatic scrolling from the bottom only is provided. It is regrettable that only one cursor shift direction (back-space) is supported by software at this time, but presumably this will be rectified in Basic 2. The TRS-80 is not as strong on graphics as the PET and other micro computers such as the Apple. Almost any graphic is possible but they are carried out by an instruction to "light" one of 128 horizontal by 48 vertical small blocks, each made up of 2 dots wide by 8 high. The command SET lights a specified block and RESET extinguishes it. This is not too bad with iterative designs. For instance the following programme will draw 22 vertical lines on the screen and can be written in under 60 bytes:

```
10 FOR X = 0 TO 26 STEP 6
20 FOR Y = 0 TO 47
30 SET (X,Y)
40 NEXT Y
50 NEXT X
99 GOTO 99
```

The everlasting loop in the final line ensures that the prompt sign does not appear on the screen. If however a non-iterative design, say the picture of a dog, is the subject of a programme, it gets to be a tedious business!

Incidentally, although the Manual does not mention it, the cursor controls, although not as yet effective in moving the cursor in three directions, may be used in their Shift position in PRINT or INPUT commands to draw arrows on the screen, in any of four directions. The text consists of 16 lines of 64 characters. Tandy say in their sales literature that this is software selectable to 32 characters per line, which presumably would result in larger characters but this is not mentioned in the Manual and despite a telephone call from the Eastbourne manager to Head Office, no information has been forthcoming.

The 233 page Operating Manual entitled "User's Manual for Level 1" will please some customers and horrify others. It is written in a "folksy, lets you me and the computer be friends together" style, common in the U.S. but which will surely grate on the nerves of some customers over here. It does however, succeed in its objective, which is to give a complete newcomer to programming a reasonable grounding in Basic. I personally found it instructive and amusing and am sure that it can teach someone with *no experience* of computers or programming, most of that which it is necessary to know, to get the most out of the TRS-80. Two and a half pages of Addendum are included and although it is a tiresome job, these mistakes should be rectified in the Manual before starting to read it. Unfortunately there are other errors but most of them are fairly obvious and a careful examination of a possible mistake leaves little doubt in the user's mind as to whether he or the manual is at fault.

One serious deficiency is the lack of an index. It is true that the mnemonics for the statements and commands are itemised at the back but they reference to chapter numbers rather than to page numbers. If this is intended to take the place of an index, then it is a poor substitute. One final criticism of the Manual is that the proportion of space allocated to one subject is not in balance with that allocated to another. For instance, over 10% of the Manual deals with the subject of graphics and yet only 8 pages deal with using the cassette for storage and 4 of these are said to be "optional" in that they deal with Data Files. Whilst it is true that due to the restrictions of ROM and the consequential shortage of BASIC commands used, the use of Data Files is very restricted, it is thought that they deserved more mention than an "optional" 4 pages in an Appendix!

In summary therefore, it is fair to say that the Manual does that which it is intended to do and a prospective buyer with no knowledge of computers or programming



puter, but reasonably strong F.M. stations will not be perceptibly effected. This is something worth bearing in mind when deciding where the computer is to be sited.

A Software Exchange for the TRS-80 has already been started in the U.S.A. and no doubt it will not be long before one is started here. Meanwhile anybody requiring the American address and membership details may obtain it from this magazine or the author.

Whilst this article was being written Tandy in the U.S.A. have announced the (hopefully) final form that their marketing of TRS-80 systems will take and on the assumption that these will cross the Atlantic in the same form, it was thought that a "preview" may be of interest. There are now five versions as shown in the list below. To get a rough estimate of what the systems may cost over here in pounds, divide the dollar amount by a factor of 1.25 but bear in mind that this is only a rough estimate and Tandy have not announced the actual prices, nor even that the equipment will be available in England.

System Name	Contents	U.S. Price
TRS-80 "Breakthru"	As described in this article	\$599
TRS-80 "Sweet 16"	As above but with 16K RAM	\$899
TRS-80 "Educator"	As "Breakthru" with a "Screen Printer" which reproduces the screen on 5-1/2" tape.	\$1198

TRS-80 "Professional"	As above but with 16K RAM, a disk drive, an expansion interface and Level 2 Basic.	\$2385
TRS-80 "Business"	As above but with 32K RAM, a line printer and two disk drives.	\$3874

Just as "dog bites man" is not news but the reverse is, so it is necessary to realise that the many matters I have not mentioned reflect the computer *working well*, and are therefore not news. The TRS-80 is aimed at that section of the public, whether hobbyist or business, who are interested in computers but know little or nothing about their use. If they have any sense, they are not therefore about to spend thousands of pounds to find out whether they either enjoy programming as a hobby or whether computer usage is viable in their business. For these people the TRS-80 is *ideal*. It is not heavily orientated towards graphics nor towards data processing,

It is a truly middle of the road, general purpose machine at a (comparatively) low price. Within these parameters *it is to be recommended* and Tandy *are to be congratulated* on attaining precisely, what I imagine their design specifications directed them to achieve. In North America alone Tandy have 6000 shops, so by the law of averages, there are going to be an awful lot of these computers around, which has to be beneficial for all owners.

(PCW Next issue: The Tandy TRS-80 is evaluated by Consultant Mike Dennis, PCW).

## THE NEWBEAR COMPUTING STORE



7 Bone Lane, Newbury, Berks.

Sales & Service: 7 Bone Lane, Newbury, Berkshire RG14 5SH. Tel No: 0635 49223

### Software & Literature Section



All books are Subject to availability.

#### COMPUTER DESIGNS

	£	P. & P.
77-68 6800 Microcomputer	7.50	.50
Spare diagram set for 77-68	1.50	.50
WB-1 TTL Microcomputer	6.50	.50
Spare diagram set for WB-1	1.00	.50

#### FROM ADAM OSBORNE ASSOCIATES

	£	P. & P.
Introduction To Microcomputers		
Volume 0: The Beginners Book	5.95	.50
Volume 1: Basic Concepts	5.95	.50
Volume 2: Some Real Products June 1977 Revision	11.95	1.00
8080A/8085 Assembly Language Programming	6.95	.50
Some Common Basic Programs	5.95	.50
6800 Programming for Logic Design	5.95	.50
8080 Programming for Logic Design	5.95	.50
Payroll with Cost Accounting in Basic	9.95	1.00

#### SCELBI

	£	P. & P.
Understanding Microcomputers & Small Computer Systems	7.56	.50
Scelbi 6800 Software Gourmet Guide & Cook book	7.95	.50
Scelbi 8080 Software Gourmet Guide & Cook book	7.95	.50
8080 Standard assembler	15.95	.75
The Scelbi Byte Primer (replaces Best of Byte)	9.95	1.00
The 8080 Programmers Pocket Guide	2.35	.30

#### ZILOG

	£	P. & P.
Z-80 Technical Manual	4.00	.50
Z-80 PIO Technical Manual	2.50	.50
Z-80 Programming Manual	4.00	.75

#### MOTOROLA

	£	P. & P.
Understanding Microprocessors	2.75	.30
M6800 Microprocessor Programming Manual	4.50	.50
M6800 Microprocessor Applications Manual	9.50	1.00

#### SYBEX

	£	P. & P.
Microprocessors C201	8.00	.50
Microprocessors Interfacing Techniques	8.00	.50

#### BASIC

	£	P. & P.
Instant Freeze-Dried Computer Programming In Basic	4.95	.75
By Jerald R. Brown		
My Computer likes me when I speak in Basic	1.65	.30
Computer Programs that Work	2.40	.75
Basic Software Library		
Volume 1:	17.50	.50
Volume 2:	17.50	.50
Volume 3:	26.95	.50
Volume 4:	7.95	.50
Volume 5:	7.95	.50
Volume 6:	TBA	.50
Volume 7:	26.95	.50
What to do after you hit Return	7.00	.75

#### SOFTWARE (Liable to 8% VAT)

#### GAMES

	£	P. & P.
The Bear Game	2.00	.50
The Well Tempered Microprocessor	2.00	.50

#### Z-80 SOFTWARE

	£	P. & P.
ZM-PT Z80 Control Monitor (Paper Tape)	15.00	.75
ZM-108 Z80 Control Monitor (PROM)	35.00	.75
CB-PT 3K Control Basic (Paper Tape)	15.00	.75
CB-308 3K Control Basic (PROM)	75.00	.75
ZA-PT Assembler/Resident 0. System (Paper Tape)	30.00	.75
ZA-808 Assembler/Resident 0. System (PROM)	175.00	.75
16KB-PT 16K Z-80 Basic (Paper Tape)	75.00	.75
16KB1608 16K Z-80 Basic (PROM)	370.00	.75

6800 Software - Phone for Availability Send for our up to date lists.

Our range of books is constantly expanding. Please write for latest list.

Goods are normally shipped within 24 hours subject to availability. Barclay card & Access VAT at 8% for Hardware Components. 30p postage and packing unless otherwise stated. Cheques to be made out to 'The Newbear Computing Store'.

# THE NEWBEAR COMPUTING STORE



## Hardware Components Section



### SYSTEMS \*

S-50 bus		S-100 bus	
MSI 6800 MICROCOMPUTER KIT	£375.00	CROMEMCO Z-2 kit (C.P.U + Card Frame)	£395.00
(2MHz clock 8K BYTES RAM)		BYTESAVER PROGRAMMING BOARD Kit (2708)	£95.00
FLOPPY DISCS, fully built and tested		TU-ART Interface Kit	£130.00
Smoke Signal Broadcasting		16K BYTE Dynabyte ram card (fully tested)	£415
Single Drive £522.00; Double £785.00; Treble £1,045.00		North Star Mini Floppy Kit	£490.00

### HARDWARE COMPONENTS

#### BEAR BAGS

1. 77-68 CPU P.C.B. Components and Edge Connectors	£ 49.50
2. 77-68 Leds and Switches	£ 14.95
3. 77-68 Power Supply.*	£ 17.95
4. 77-68 19" 5 u Rack and Backplane.*	£ 27.70
5. 77-68 4K Ram P.C.B. and Components	£ 75.00
6. 77-68 Mon 1 P.C.B. and Components	£ 50.70
7. 4K Ram Exorciser P.C.B. and Components	£ 71.50
8. 8K Ram Exorciser P.C.B. and Components.*	£160.00
9. Petitevid V.D.U. Kit	£ 85.00
10. Kansas City Cassette Interface.	£ 18.95

Unencoded Keyboard	£27.50	+ £1.50 p&p
Licon Assembled & Tested Keyboard		+ £1.50 p&p
Hexadecimal Keypad (19 positions)		£11.25
Keyboard Case	£20.33	+ £1.50 p&p
19" 5 u Rack	£23.40	+ £1.50 p&p
U.V. Prom Eraser		£56.00*
U.H.F. Modulator		£4.50
9368 7 SEG. Display Decoder		£2.45
FWD 500 1/2" 7 SEG. Display		£1.05
C 30 Cassette Memorex	£0.75	5" Floppy Discs £4.00
C 60 Cassette Memorex	£0.90	8" Floppy Discs £4.00
C 90 Cassette Memorex	£1.20	

Plus VAT 8% 50p p. & p.

\*£1.00 Postage & Packing.

#### MEMORIES

2102-1	£ 1.25
2102L-1	£ 1.35
2112	£ 3.04
2513	£ 6.50
SWATBUG	£16.00
4027	£ 4.50
2114	£10.00
4116	£29.50
2708	£10.50
MC6830L7 (MIKBUG)	£13.65

#### CRYSTALS

1 MHz	£3.50	1,008 MHz	£3.50
5 MHz	£2.60	10 MHz	£2.60
4 MHz	£1.50	(VAT 12½%)	

#### Scotchflex Connectors

26 way plug	£2.50	socket	£1.70
40 way plug	£3.10	socket	£2.70
50 way plug	£4.10	socket	£2.90

#### Motorola Microcomputing I.C.'s

MC 6800	£12.00	MC 8602P	£ 2.88
MC 6820	£ 6.20	MC 14536P	£ 3.69
MC 6850	£ 6.74	MC 3459	£ 2.53
MC 6810AP	£ 3.61		

#### Zilog Microcomputing I.C.'s

Z80 CPU 2.5 MHz	£15.50	Z80A CPU* 4 MHz	£20.50
Z80 P10 PS	£10.00	Z80A P10*	£11.55
Z80 CTC PC	£10.00	Z80A CTC*	£11.55

#### Microprocessors

SC/MP 11	£10.30; 6502	£14.93; 2650	£17.50
(INS 8060N [ISP8A/600N])			

F-8 and 8080 parts available soon.

INS 8154N £ 8.18

#### Low Power Schottky TTL Price List

Ref.	Price	Ref.	Price	Ref.	Price	Ref.	Price
74LS00	0.18	74LS55	0.21	74LS153	0.58	74LS241	1.85
74LS01	0.21	74LS63	1.48	74LS154	1.45	74LS242	1.80
74LS02	0.21	74LS73	0.34	74LS155	1.05	74LS243	1.72
74LS03	0.21	74LS74	0.38	74LS156	1.05	74LS244	1.85
74LS04	0.26	74LS75	0.55	74LS157	0.60	74LS245	4.10
74LS05	0.26	74LS76	0.34	74LS158	0.65	74LS247	0.96
74LS08	0.21	74LS78	0.34	74LS160	1.22	74LS248	0.96
74LS09	0.21	74LS83	1.05	74LS161	1.22	74LS249	0.96
74LS10	0.21	74LS85	0.97	74LS162	1.22	74LS251	1.48
74LS11	0.26	74LS86	0.38	74LS163	1.22	74LS253	1.05
74LS12	0.21	74LS90	0.71	74LS164	1.18	74LS257	1.09
74LS13	0.55	74LS91	1.20	74LS165	1.70	74LS258	1.09
74LS14	1.26	74LS92	0.92	74LS166	1.85	74LS260	0.45
74LS15	0.21	74LS93	0.72	74LS168	2.03	74LS261	4.95
74LS20	0.20	74LS95	1.45	74LS169	2.03	74LS266	0.38
74LS21	0.26	74LS96	1.88	74LS170	2.61	74LS273	1.91
74LS22	0.21	74LS107	0.38	74LS173	2.28	74LS275	2.52
74LS26	0.31	74LS109	0.38	74LS174	1.18	74LS279	0.85
74LS27	0.21	74LS112	0.38	74LS175	1.10	74LS280	1.85
74LS28	0.40	74LS113	0.38	74LS181	7.72	74LS283	0.96
74LS30	0.26	74LS114	0.38	74LS183	2.98	74LS289	1.05
74LS32	0.24	74LS123	0.95	74LS189	2.82	74LS352	0.96
74LS33	0.40	74LS124	1.10	74LS190	2.02	74LS353	0.51
74LS37	0.30	74LS125	0.50	74LS191	2.02	74LS365	0.51
74LS38	0.30	74LS126	0.50	74LS192	1.53	74LS366	0.51
74LS40	0.27	74LS132	0.85	74LS193	1.86	74LS367	0.51
74LS42	0.88	74LS133	0.40	74LS194	1.91	74LS368	0.51
74LS47	0.97	74LS136	0.38	74LS195	0.95	74LS373	0.92
74LS48	0.96	74LS138	0.65	74LS196	1.21	74LS374	0.95
74LS49	1.18	74LS139	0.65	74LS197	1.21	74LS386	0.38
74LS51	0.21	74LS145	1.30	74LS200	5.10	74LS670	3.04
74LS54	0.21	74LS148	1.52	74LS202	4.86		
		74LS151	0.96	74LS240	1.82		

All I.C.'s are 100% guaranteed and manufactured by Texas, Signetic, National, etc. VAT 8%

#### INTERFACING I.C.'s

SFF 96364	£16.20
MC 1488p	£ 1.40
MC 1489p	£ 1.40
75150p	£ 1.30
75150n	£ 1.20
75154	£ 2.30
4N33	£ 1.95
AY-5-1013	£ 5.30
6402	£ 4.50

#### BUFFERS

81LS95	£1.30
81LS96	£1.30
81LS97	£1.30
81LS98	£1.30
8T26	£1.84
8T95	£1.60
8T97	£1.60
74367	£1.30

Goods are normally shipped within 24 hours subject to availability. Barclay card & Access VAT at 8% for Hardware Components. 30p postage and packing unless otherwise stated. Cheques to be made out to 'The Newbear Computing Store'

Send for an up-to-date catalogue to:

## The Newbear Computing Store

7 Bone Lane, Newbury.

Tel. 0635-46898.

Callers welcome Monday to Saturday 9.00a.m.-5.30p.m.

The Newbear Computing Store is a division of Newbury Laboratories Ltd.

#### 24 HOUR TURNROUND ON ORDERS.

\*Securicard delivery advised; please phone for charges.

Please add 8% VAT to all Prices. P & P 30p unless otherwise stated. Barclaycard & Access welcome, overseas orders issued with pro-forma invoice. Send for catalogue to Newbear Computing Store, Bone Lane, Newbury, Berks. Callers welcome Mon - Sat., 9.00 - 5.30 p.m. but please phone us first on 0635 49223

# COLOUR IS AN

# APPLE II

## PCW REVIEW

JOHN COLL & CHARLES SWEETEN



The Apple II is a complete, assembled computer system. The system we had for evaluation loaned by Mike Sterland of Personal Computers Ltd., consists of a single box encasing the keyboard, the power supply, and the main board, together with a separate cassette recorder and a SONY portable colour TV set. There is also a carrying bag for the computer and two game paddles are included. All the necessary cables are present and attached.

The fundamental reason for buying an APPLE II must be for its colour graphics, for its extra high resolution graphics package, for its extreme portability and for its high quality construction. It is designed with expansion and extension in mind and, for example, you may increase the amount of memory simply by plugging chips into the sockets on the single circuit board.

The main printed circuit board and switched mode power supply is a model of good design. 66 integrated circuits are neatly laid out, with clear printed identification on the upper side. There is space for expansion RAM on board, so one can easily put extra RAM's in place. I/O connectors are a row of sockets near the back of the board. The absence of keyways or polarizing pins is unfortunate. The main board carries a small loud-speaker; the keyboard is well constructed.

The add-on UHF modulator, plugged into port #7 is of much lower standard — but performed adequately. Modulating a carrier with detail as fine as the HI-RES graphics offers is stretching the system to its limit as can be seen in the photographs. We would expect a dramatic improvement with a colour monitor.

The price for the computer itself with 16K of RAM is £1,250 plus £35 for the carrying bag.

A maintenance contract is available at 7½% of purchase every year. The software is free with the system we tested so the price is £1,285 plus VAT. The complete cost, including £220 for a TV and £15 for a cassette player, is £1,520 plus VAT.

The box is made of strong 5 mm plastic and is light, the interior has plenty of room and is very neatly laid out. For its sheer convenience we would recommend the carrying bag which makes transporting it very easy and safe. The keyboard is a delight to use and sets a standard of touch which other system manufacturers, who are not looking for rock bottom prices, would do well to aim at. We would like the Reset Key not to have been next to Return and = though. It is nice to have a Repeat Key that works with everything, and the form of screen editing that is available is particularly useful. Back space does not delete on screen, but is interpreted by the software, and forward space acts as if the character left has been typed. So that one can move the cursor to an offending line and repeat forward space, amend the incorrect character(s) and repeat forward space to the end of the required line. BASIC would re-format embedded blanks (or lack of them) to a standard format.

The display uses a UHF output and is 24 lines of 40 characters, with the standard ASCII set. It is memory mapped; that is to say that the contents of the screen are contained in a section of memory and the locations for each position of the screen are thus instantly addressable. This does give one obvious advantages in terms of speed and of course one can send the cursor to any position without having to wait. One can also have more than one page of text (or graphics) and can switch from one to another instantly.

Backing storage is cassette and the instructions for loading are very simple. The transfer rate is said to be 1500 bits per sec. which is five times as fast as many other cassette transfer rates. Applesoft extended BASIC takes 1½ mins. to load. Talking to Mike Sterland of Personal Computers Ltd., we gathered that APPLE are very much aware of the shortcomings of their cassette system as a backing store, notably that there is no way of finding the right file on a cassette except by listening to it. They are working on a floppy disk system rather than trying to improve the cassette handling. It will be interesting to see how APPLE cope with the file handling and to see whether they introduce an Editor and an Assembler.

Now, the prime reason that anyone buys an APPLE II must surely be for the colour graphics. In the low RES graphics mode, each character position of the screen is divided into two rectangles and each rectangle can be plotted in any of 15 brilliant colours plus black. Four lines of text can be retained at the bottom, which leaves a 40 by 40 grid for the colours. Very attractive pictures can be drawn and colourful games can be played.

Somebody must produce Startrek in colour soon! But one of the restrictions is that alphanumeric cannot be mixed with the colour display so that any labelling of diagrams has to be done in the bottom four lines. The instructions are nice and simple in both versions of BASIC, and one of the advantages of the memory mapped VDU is that complete lines (PIXELS) can be drawn *instantaneously*. But let us look first at the software that is at present available on this machine.

The software consists of a monitor, integer BASIC and extended BASIC and High-resolution graphics. The machine language environment is good. The firmware



An Apple Newton could have done with.

monitor provides an extensive set of instructions and binary data can be simply stored on cassette. However, the absence of an Editor, in the accepted sense of that word; or of an Assembler, prevent anything but the simplest machine language programs from being written.

The monitor commands enable memory to be examined and changed. A block of memory can be moved, and the move command can be easily used to fill a block of memory with any bit pattern. One block of memory can be verified (compared) with another block. Input and output ports can be easily changed.

Memory can also be examined directly in mnemonics using the resident dis-assembler. Dis-assembly can be of single lines or 20 lines at a time. That is a *very* useful feature. In addition, a program can be single stepped or "traced", potentially a most useful facility - marred only by the fact that the print-out is so fast that it is quite impossible to read if more than one page is displayed. The same criticism applies to BASIC listings, though obviously the latter can be controlled with the LIST command.

*This must be one of the most comprehensive monitors available for a micro.* It is a pity that, with such an outstanding monitor, there is no Editor/Assembler available. When a good Editor/Assembler is available the machine will be very easy to use in that environment.

Integer BASIC is in ROM and can be entered via control B, or via control C if you wish to retain any BASIC program that is in memory. Integer BASIC supports numbers up to 32767 and has a useful function MOD, which reduces a number to the remainder after a division. For example, 17 MOD 5 gives 2. There is an automatic line numbering command which is useful, but there is no renumber command (however, the on-screen editing is a feature that makes the lack of renumbering much less serious than would otherwise be the case). Multiple statements are possible but need to be watched when used with an IF statement. Usually when an IF statement is not satisfied the rest of the line is not processed. In this BASIC it *is* processed. For example,

```
IF 2 = 3 THEN A = 3 : A = 5
```

will give A = 5 in this BASIC, but will give A = 3 in Apple extended BASIC, which is a more usual result. A null return to a string input is accepted and gives a null string. It is possible to LIST a section of a program and to delete a section. Naturally there are no trigonometric functions or such-like but PEEK and POKE are supported, and the manual is particularly helpful about using these generally.

Extended BASIC is called Applesoft and was written by Microsoft. It is loaded under integer BASIC and then a program is run to complete the loading and give a choice between graphics or REM and LET statements.

Applesoft is much like other extended BASIC's and it is only worth listing the differences. RENUMBER is again absent. The logical operators AND, OR and NOT are present. String "space" did not have to be allocated, and "full length" strings of 255 are available. The only restriction is the amount of memory that is free for the user's program. Furthermore, string matrices are available where each element is a string. When the Apple gets a disk this will be a very definite plus.

Incidentally, null returns to a string input are not accepted. Multiple statements with conditional statements work in the expected manner. There are also two excellent pages in the manual which give details of addresses to PEEK or POKE to do such things as cursor addressing, black on white letters, different margins, page lengths, and etc. There is a list here of various machine language subroutines that can be used for communicating with the game paddles and with the internal speaker.

There are some differences in language between the two versions of BASIC which we feel could be tidied up. For example, if one wants to list lines 300 through 400 one uses LIST 300, 400 in Integer BASIC and one uses LIST 300-400 in Applesoft. To enter a machine language subroutine one uses CALL 3000 or USR (3000) depending on which level of BASIC one is in. CON or CONT, TEX or TEXT; no, the three letter versions are not in the same level of BASIC. They are very minor differences but irritating. Slightly annoying also is the fact that a program that is syntactically correct in both versions of BASIC cannot be saved in one and loaded in the other. One feature that neither version has and that we would plead with producers to provide is the ability to remain on the same line after an input. This allows so much more freedom to produce an acceptable format for interactive data presentation.

The commands that Applesoft uses to control the colour graphics are PLTG (go to graphics), PLTCX (colour X), PLTH X1, X2, Y (draw horizontal line), PLTV Y1, Y2, X (draw vertical line), PLTP X, Y (plot a point), TEX (text mode). They are very easy to use and appear to have few vices. Text is limited to the lines at the bottom of the screen so that the graphics are used for drawing pictures rather than diagrams. Do not be misled into thinking that the Financial Times Index can be plotted on the screen in a useful way. First, you cannot label the vertical axis or the trend lines, and secondly, you can only store the data as a DATA statement in a BASIC program, which makes updating a manual process, which I imagine defeats the purpose of such a program.

But of course the scope for game playing is enormous. Most teenage boys can reveal the details of Battleships, but what about 5-a-side football with the pitch, players and ball in individual colours? Or golf with greens, darker shades for the less approved areas of the course, yellow for bunkers, and so on?

The system that we had contains 20K and with extended BASIC this gives 5K bytes of program space. With Integer BASIC there is more than 10K bytes available for user programs. Integer BASIC is also faster and thus it is more suitable for game playing on both counts. The Benchmark timings using the programs listed in the first issue of PCW are shown in Figure 1.

Benchmark	Integer BASIC	Extended (Applesoft)
BM1	1.5	1.3
BM2	3.2	8.5
BM3	7.3	16
BM4	7.2	17.8
BM5	8.9	19.1
BM6	18.6	28.6
BM7	28.2	44.8
BM8	-	10.7

Figure 1

Timings of Benchmark programs

These show that Integer BASIC is fast and that Applesoft is satisfactory. So there should be few problems with speed. Microsoft Co. wrote Applesoft and also the PET and Altair BASICs, which is why the timings are "rather similar".

We have said that one of the reasons for buying an Apple II would be for its high-resolution graphics and it is worth looking at this in detail. At this moment (May 1978), we are not aware of another complete machine in this price bracket that has this particular feature. High resolution graphics is a set of machine code sub-routines which allow the user to mark points and lines on a screen that is 280 points across by 160 points down.

Three slightly variable colours are available (green, violet, white) and also "black". The subroutines allow one to poke the co-ordinates of a point into three locations and CALL 3780 will plot the point on the screen. After another set of co-ordinates has been poked into the same locations, one has the option of joining the last two points by CALL 3786. The beauty of a real line on the screen has to be seen to be believed. Given enough points, or lines, a curve can be drawn which from a few feet away looks as smooth as one could wish for. What is more, it is in colour and different lines or curves can thus be distinguished on the same screen.

However, there is an oddity which users may be puzzled by, and that is that the colours do not come out consistently. It is possible to draw a horizontal line in any position but not a vertical line; vertical green lines can be drawn at odd co-ordinates only, and violet ones at even co-ordinates only. Every other green line (visible ones) is yellowish, and every other violet line is bluish. White vertical lines can be drawn at every position, but are in fact shades of violet, green, blue, green in that order. It is only if one fills in solid patches of a single colour that its true shade emerges, and that is not always what one wishes to do with high resolution graphics!

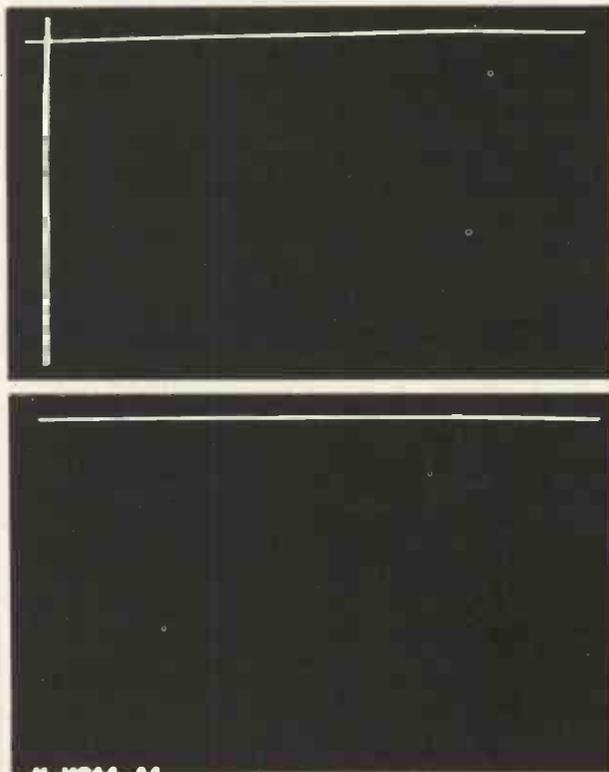
The disappearance of alternate vertical lines can result in moire fringe effects. Figure 2 gives the test program and Figure 3 shows the lack of vertical line at X = 11 with the colour set to violet. But one gets used to it and anyway no-one else has colour graphics like this at this sort of price.

```

10 CALL 3072
20 INPUT "COLOUR",C
30 POKE 812,C MOD 256
40 INPUT "X,Y:",X,Y
50 CALL 3086
60 X=XMOD279:POKE800,X MOD 256
70 POKE 801,X/256:POKE 802,0:CALL 3761
80 POKE 802,159:CALL 3786
90 POKE 800,0: POKE 801,0
100 POKE 802,Y MOD 159:CALL3761
110 POKE 800,23:POKE 801,1:CALL3786
120 GOTO 40

```

Figure 2 High resolution line test



When I (C.S.) was first let loose on a computer it was a vast machine working in FORTRAN and after two days of my course, I discovered that it had a graph plotter on it. I spent the rest of the course drawing a four petal flower on a long stem. It seemed quite difficult, largely, I think, because I could not understand the manuals. Anyway, I have always wanted to do it on a colour TV.

So, to demonstrate what the Apple high resolution graphics are capable of, I have written a polar graph program. The equations that I wanted to use need trigonometry and Integer BASIC naturally does not have such things. Happily there is a block of memory in the high resolution package which has a table of values of SIN(X) with the function evaluated between  $\phi$  and 255 and with the values of X going from  $\phi$  to 255. This table starts at 3840 and goes through to 4095 decimal. SIN=3840:PEEK(SIN + (TH)MOD256) gives SIN(TH) for positive TH with 256 units in a revolution. PEEK(SIN + (TH+64)MOD256) gives COS(TH).

```

100 SIN=3840
110 POKE812,170
120 CALL3072
130 CALL3086
140 POKE800,0:POKE801,0
150 POKE802,80:CALL3761
160 POKE801,1:CALL3786
170 POKE800,123:POKE801,0
180 POKE802,0:CALL3761
190 POKE802,159:CALL3786

300 LIST500
310 PRINT"85VIOLET,170GREEN,255WHITE"
320 INPUT"COLOUR(85,170,255)",C
330 INPUT"NO.OF REVOLUTIONS",N
340 LIST500
350 POKE812,C
360 GR:B=0
400 FOR TH = 0 TO 256*N

500 R= 2*(PEEK(SIN+(TH*4/3+64)
MOD256)-127)
510 R=R+256
550 IF ABS(R)>816THEN800
560 R=R*5/16
570 E=0

600 S=PEEK(SIN+(TH)MOD256)-127
610 C=PEEK(SIN+(TH+64)MOD256)-127
620 X=R*C/256+123
630 Y=R*S/256+80
640 IF X >279 THEN E=1
650 IF Y >159 THEN E=1
660 IF X<0 THEN E=1
670 IF Y<0 THEN E=1
680 IF E=1 THEN B=0
690 IF E=1 THEN 800

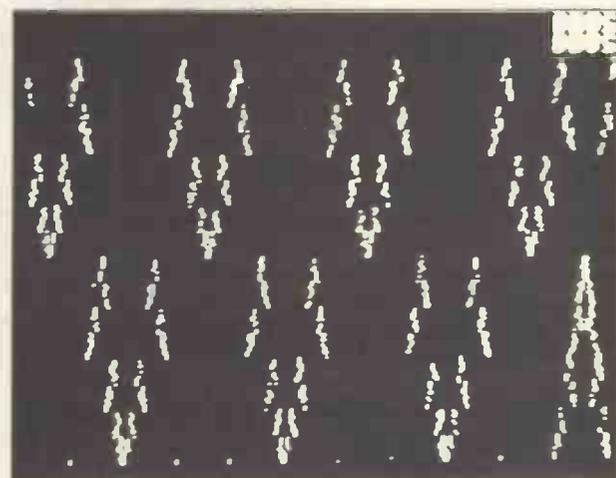
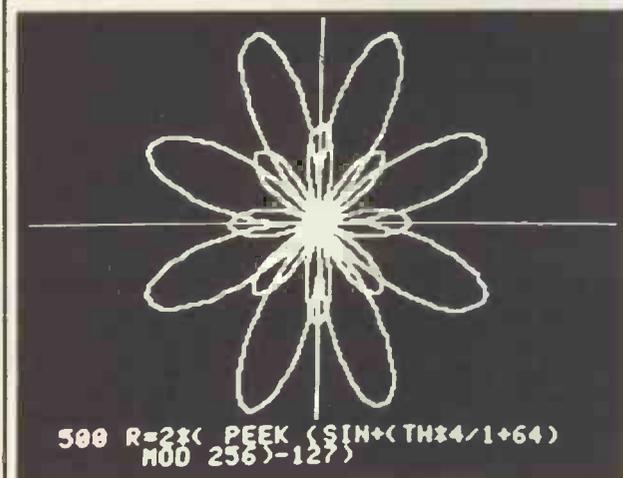
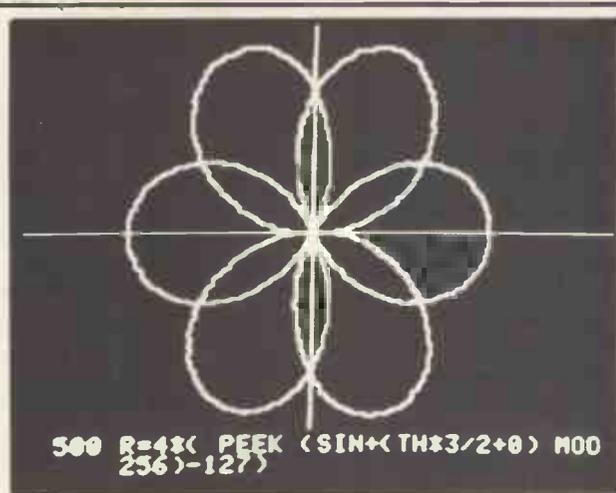
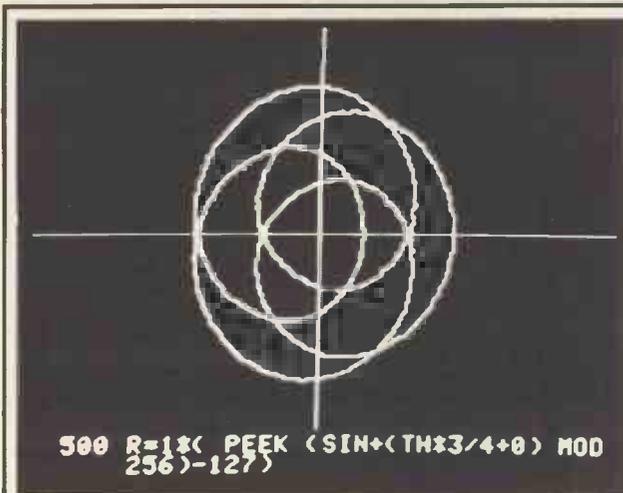
700 POKE800,XMOD256:POKE801,X/256
710 POKE802,Y
720 IF B=0 THEN CALL 3780
730 B=1
740 CALL3786

800 NEXT TH
900 END

```

The program contains the equation of the picture at lines 500 and 510; these can easily be altered using the edit facility. From 600 to 700 converts the co-ordinates into x and y values and joins successive values up on the display.

Lines 100 to 190 draw the axes and lines 300 to 400 control the colour and the number of revolutions needed to describe the curve. To draw one curve on top of another one merely starts at 300. To make solid flowers



one can join each point to the origin by adding the following lines:

```
750 POKE 800,123:POKE 801,0 : POKE 802,80:
CALL 3761
760 POKE 800,X MOD256: POKE 801,X/256:
POKE 802,Y: CALL 3786
```

The photographs illustrate the remarkable resolution that is available. Figure 4 is obtained with 500 R = PEEK(SIN+(TH\*3/4)MOD256)-127 and 4 revolutions. Figure 5 is obtained by a double plot; first 500 R=4\*(PEEK(SIN+(TH\*3/2)MOD256)-127) in white and with two revolutions; then 500 R=2\*(PEEK(SIN+(TH\*6+64)MOD256)-127) in violet with one revolution. Figure 6 is obtained with 500 R=4\*(PEEK(SIN+(TH\*4)MOD256)-127) in violet with one revolution; then follow with 500 R=2\*(PEEK(SIN+(TH\*4+64)MOD256)-127) in white with one revolution. Figure 7 shows the effect of filling in figure 6 and one can just make out the shot silk effect in the outer petals. Naturally it needs colour to do it full justice.

For those who would like to pursue this further I have listed a few shapes to try.

- Circle 500 R=2\*(PEEK(SIN+(TH+64)MOD256)-127)+256
- Ellipse 500 R=10\*(PEEK(SIN+(TH+64)MOD 256)/7  
510 R=32767/(R+256)
- Hyperbola as for the Ellipse but divide line 500 by 3 rather than 7
- Trisectrix 500 R=2\*(PEEK(SIN+(TH+64)MOD256)-127)+128

The manuals were in some ways *excellent* and included, for example, a very detailed discussion of the sub-routines available and a full listing of the monitor and board layout and circuit diagrams. But there are many

errors and though the scope is comprehensive, it is very obviously a first edition and even includes some handwritten pages. A point worth mentioning to users is in the section on program editing with Applesoft on p.58 etc., where the example is wrong. The key 'left arrow' is not identical to escape B as stated. You must use escape B to make the example work and p. 56 is correct.

As an example of the locations which can be manipulated and the sections of resident code that can be utilised within one's own programs, here is a selection of what is given many times throughout the manuals.

- POKE 32,L Set left margin of TV display
- POKE 33,W Set the width of TV display
- POKE 34,T Set top margin line of TV display
- POKE 35,B Set bottom margin line of TV display
- POKE 36,CH Move cursor to horizontal position CH+1 relative to left margin
- POKE 37,CV Move cursor to vertical position CV T less than CV less than B
- POKE 50,127 Set text mode to flash outputted characters
- X = USR(-958) Clear display from present cursor position to bottom of window and left of window
- X = USR(-912) Scroll window up one line and leave characters outside window unchanged
- PEEK(-16384) Examine keyboard input: if contents greater than 127 then a key has been hit whose value in ASCII is found from 127 AND the contents

The manual gives instructions for connecting peripherals via one of eight peripheral connectors near the back edge of the printed circuit board. Each of these has sixteen addresses associated with it, and all the signals required are fully described.

We mentioned earlier that it is easy to plug in more memory. What is also rather nice is that the manual tells you *exactly* how to do this. There are three memory select sockets, and these allow you to put in jumpers to place either 4K chips, 16K chips or both at any address

block you please up to 48K. There are a few restrictions naturally, but what we like is the approach that the consumer knows what he wants better than the manufacturer does, and here is one manufacturer who is anxious that we should get it.

All in all then the Apple II is a very promising machine which is nicely packaged, has minimal BASIC at the touch of a key, has very adequate extended BASIC, has colour graphics which are half way to being beautiful, has an excellent monitor, has well thought out expansion possibilities. The Apple would be even more of a temptation were its price slightly lower.

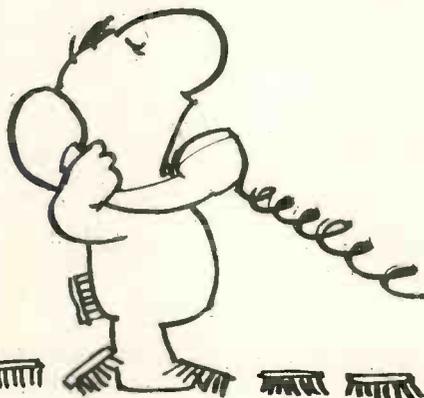
There is also considerable development to be done on the software side to make it comparable with what is available on many other systems; and the expansion possibilities will need to be catered for by the software.

We would like to say how much co-operation we received from Personal Computers Ltd and how much we appreciated their attitude — it bodes well for the future.

So for the moment, colour is an Apple II.

Memory Address Allocations in 4K Bytes

0000	text and color graphics display pages, OS/2 stack, printers, etc.	8000	
1000		9000	
2000	high res graphics display primary page	A000	
3000	"	B000	
4000	high res. graphics display secondary page	C000	addresses dedicated to hardware functions
5000	"	D000	"
6000	"	E000	ROM socket D0: spare
7000	"	F000	ROM socket D8: spare
			ROM socket F0: BASIC
			ROM socket F8: BASIC
			ROM socket F0: BASIC utility
			ROM socket F8: monitor



"Quick!  
Get me a microbiologist."

# Computabits System 8

A POWERFUL BRITISH-MADE COMPUTER SYSTEM AT A LOW PRICE.

- ★ 1 Megabyte of disc store
- ★ Up to 64K of programmable memory
- ★ Simple turnkey operation
- ★ User programmable in BASIC
- ★ Expandable, modular S-100 Bus design
- ★ Extensive range of business applications under development
- ★ Prices from £3,800

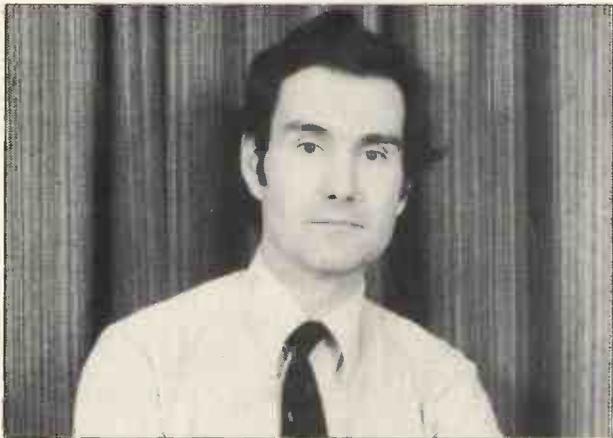
Computabits Limited can supply a wide range of other lower-cost computer systems such as the Commodore PET and Cromemco Z-2. Also memory, cpu, interface and disc controller boards.

For further information on these or the System 8, write or ring:

**Computabits Limited, 41 Vincent Street, Yeovil, Somerset.**  
(0935) 26522

# TOO HIGH A PRICE

Dr. JOHN DAWSON



Have you seen your doctor recently? Has your family been to the surgery? What did you tell him and what do you think he did with that information? Large main-frame computers will be used by the Department of Health and Social Security's Area Health Authorities to store and distribute medical information about all pre-school children from *conception* onwards if the Department manages to bulldoze its proposals into operation. The Child Health Computing Committee (CHCC) has now finalised system proposals after four years celebration and the Department of Health and Social Security (DHSS) intends to have the necessary software written by the Welsh Health Technical Services Organisation for pilot trials to be completed by late 1979. The software will then be put up for sale to individual Area Health Authorities (AHAs) for operation to start in 1980.

And that might be fine for the AHA payroll system, but medical information is different; not because the doctors say so, but because you think it is. What may appear trivial or innocent to you or me may be intensely embarrassing or important to someone else. The fact that a middle aged man has a hernia is not something that many of us would be concerned to keep an absolute secret if we were the person affected, but if the firm employing him were selecting some of their staff to be made redundant the fact of his temporary ill health might have a disproportionate influence on their decision as to whether or not he should be included in the group to be put out of work.

The doctors have never made rules about classes of information that are or are not to be regarded as confidential. Anything that a doctor hears in the course of his work is covered and the moral obligation to maintain confidential information that is given to the doctor in the normal doctor/patient relationship is clear, except when there is a question of an overriding duty to the community. One obvious example of an overriding duty occurs from time to time when a doctor discovers a patient to be carrying typhoid while he works at an occupation in which he has to handle food. The clear danger to other members of the community must be minimised in this instance as soon as possible.

Judges and magistrates, however, are the only people who have the power to direct the doctor to answer questions in a court of law and the direction will only be given when the questions are relevant and proper.

In contrast to the CHCC jerry built plans for the Pre-School Health Module of the Standard Child Health

System, there is a pilot scheme in the West Country of great tangible benefit for the patients involved. The Exeter complex is built on a real-time basis and uses VDUs in individual doctors' surgeries, out-patients and hospital wards linked to a machine in the grounds of the Royal Devon and Exeter hospital.

The Exeter system has produced a greatly increased speed of transfer of patients records between the GP's surgery and hospital, a more controlled approach to the necessary evil of repeat prescribing (including the production of printed, and hence legible!! prescriptions) and the instant recall of a large amount of pharmacological information to help with prescribing that the doctors involved would not previously have had access to because of the sheer difficulty of finding the information quickly. Additionally, from the general practitioner's point of view the system has made it possible to produce a micro-fiche of all the patients records in his practice, that the duty doctor can carry with him at night and over the weekend. The GP who is covering for the practice at the weekend has a portable micro-fiche reader in his car and is able to pick up significant events and past history from the patient's records before commencing his visit.

The Exeter computer is a big system. The benefits are indubitably there and are being used in a controlled way to produce a better quality of medical practice; so it seems that *size itself need not be a problem*. The project, incidentally funded by DHSS, does illustrate a basic idiocy in departmental thinking. During the time that we have been campaigning for adequate safeguards in relation to the pre-school computer proposals the DHSS has said that it is impossible to build ethical principles into a computer system. They say instead that the ethical principles can only be observed by the operators of a computer system. Obvious rubbish, because *their own* pilot system at Exeter has indeed managed to incorporate ethical principles into the software of the system!

The importance of ethical controls in hardware and software should not, however, distract attention from the necessity for agreed ethical codes of practice among all computer users. Operator ethics must relate to the nature and size of the machine they are using and the purposes for which it is being used. Small business users as a group would probably find it acceptable that records of the names and addresses of individuals coming into contact with the business should be used for in-house mail advertising; the automatic transmission of those records to a Government department, for example, might be unacceptable to the majority.

The Computer Users' Ethical Subcommittee at Exeter have developed accessibility matrices for groups of personnel who may require access to various categories of data stored in the system:

The general practitioner is allowed total processing on the data displayed at his VDU at the surgery; his receptionist is allowed 'total processing' access on the name and address and other registration data, 'read and add data' access only to the patient's current and past medication, but no access to the clinical data or entries such as letters from hospital consultants.

The local authority is allowed 'read only' access to the patients' registration data and *no access* to any other information stored in the system. Additionally, the GP's ability to process the data is modified if he uses a VDU terminal at a location other than his home surgery. There are other matrices for other groups of staff.

The principles used in building these matrices were very similar to those we are pressing the Child Health Computing Committee to agree.

The two most important principles are these:—

Identifiable information is to be regarded as held for the specific purpose of the continuing care of the patient, and should not be used without appropriate

authorisation or the consent of the patient (parent or guardian in the case of a child) for any other purposes.

Access to identifiable information held in medical records is to be confined to the author and to the person clinically responsible for the patient during the episode from which the data has been collected (or their successors); *unless* specifically authorised by the clinician in the clinical interest of the patient.

All good stuff. Unfortunately, the prevailing public image of a computer is either that of a line of endless squat cabinets, on each a panel containing a myriad lights, flashing in exciting pseudo-random sequence; or as a face shaped machine with two tape reels above a slot from which protrudes impertinently a tongue of paper on which is written the word 'Answer'.

An informed appreciation of the distinction between different types of hardware is becoming an essential part of the citizen's education, for, the different uses to which various bits of hardware can be put in practice can be a real threat to both the patient and his trust in his doctor if the system is developed insensitively; and generally to the whole of society.

In the present instance the two vital distinctions in terms of hardware are: that the Department's batch processing system will require groups of operators to code information for input to the machine, instead of allowing the general practitioner (or health clinic doctor) to input his own information; secondly, the Department's proposals lead inevitably to a hard copy output which is *inherently less secure* than the VDUs and selected print — out employed in the Exeter system. For medical information, there is not a shadow of doubt that the transient display of data on a VDU is vastly prefer-

able for the majority of uses to which the system will be put.

The central point is that the doctor must be sure that information given to him by the patient is held secure, for if it is not, he cannot guarantee confidentiality to his patient. By becoming a doctor's patient you do not believe yourself in any sense to be handing over to him any rights in your private information, beyond the matter for which you consult him. What do you think he did with the information you gave him the last time you saw him? After all, however good or bad you think your doctor is, there is never a shadow of doubt that what you tell him will be kept secret — but unless we can control them these machines will be in operation in the mid-1980s and it will then be impossible to dismantle the *ideas* on which they were built and, as the hardware is replaced, so the system will continue.

Finally, leaving aside the Child Health Computing Committee's bland indifference to the essential human need for privacy, the Department's whole hardware philosophy appears to be obsolescent by several years.

We know that if the general practitioners had their own micro-systems they could within the immediate future handle the quantities of data involved at the necessary processing speeds. With systems of this sort a group practice would have physical control over its own hardware and would be able to *guarantee* confidentiality to the patients, at the most fundamental level by pulling the plug out when the surgery is closed at night!

It seems to me that computers can be used constructively to bring great benefits to medical practice and patient care, but they must be adequately controlled by the doctors, who are in turn responsible to their patients. The DHSS proposals will extort too high a price from us all for the small advantages they may achieve.

## SC/MP MICROCOMPUTER SYSTEM £99.50 +VAT

If you're in engineering, software or both, are a student or hobbyist, or simply looking for a high-powered development system for National's SC/MP at a realistic price, read on!

### HARDWARE

- 7 segment display for every register (32 digits!)
- 20 key keypad
- Completely transparent operating system
- ¼K RAM, expands to 64K
- Buffered system bus interfaces with cassette/TV/TTY peripherals etc. via plug-in motherboard. (LCDS/ETI bus compatible)
- Single step, slow step facility

### SOFTWARE

- Operating system allows easy modification of registers and memory, move and copy data blocks and much more.
- Set breakpoint register will allow traps on any location, including data and ROM
- Software breakpoint

For full details of the system and supporting products and documentation write to—

**I.E.S. LTD**  
**35 Purleigh Road, Corsham, Wilts**  
or phone  
**(0249) 713515/712317**

## MEET YOUR COMPUTING NEEDS FROM PADMEDE

We supply APPLE IIs

*Whether your applications are for the home or business, we at Padmede would like to understand your needs*

If you want to buy an Apple II, or just have a demonstration, visit or contact Trevor Williams now at:

**PADMEDE LTD**  
**The Tuns, High Street,**  
**Odiham, nr. Basingstoke**  
Tel. Odiham (025-671) 2434

*Padmede is an Apple II agent for Surrey, Sussex, Hampshire, Dorset, Berkshire and Oxfordshire*

# Computing for Everybody

Magnus Magnusson

Computers have come a long way. Today's micro computers need cost no more than a few hundred pounds — about the price of a reasonably good hi-fi — and occupy as little room space. There's been a revolution in computers. And it's taken just thirty years.

In 1948, Joe Lyons started work on their first commercial computer, Leo 1. This was run on valves. Costing approximately — £5,000,000.

In 1958, transistor circuits took the place of valves, cutting the cost to £500,000.

Then, in 1968, the mini computer started using integrated circuits, which brought the price of a computer sharply down to £50,000. Still a little too high for most households, though.

And, so to the present day — where you can buy comparable fully operational computer systems using LSI large scale integrated circuits for as little as £2,500. Of course you would not need as expensive a system as this for a computer for home use.

A basic computer starts at £275 in kit form, or £353 if you prefer to buy the unit fully assembled. This unit comes with 4000 words of memory. You can build this up to a machine with 36,000 words of memory.

Right. Now you've got your computer — how do you speak to it? How do you get it to perform for you? Well, acting as a sort of interpreter between you and your computer is the Terminal. With this, you type in all the information and data you want to put into the computer system. You can link the system to your own television set for just £4.50 — or you can buy the TV monitor.

Now, all the information you feed into a computer has to be stored. And we do this on a system of discs called diskettes. As you can see, they look a bit like gramophone records, and in a way they do a similar sort of job. We call them Twin Minifloppy Discs.

Here's another way of storing computer data. It uses magnetic tape, the kind you use in a cassette recorder. In fact, you can use your hi-fi system as part of your computer unit by means of this Audio Cassette Interface. It costs £100 and makes data storage as easy as recording your favourite radio programme.

Now we come to the most expensive part of computing. The Printer. Expensive, because this is basically a mechanical device. The price is £250. But don't let the price frighten you. You only need one if your computer is to be used for your business — then it will pay for itself.

And there you have a complete computer system to fit your office or study.

But first there's a very important question that has to be answered. Why should you buy one?

The answer is all around you. We are living in the computer age. More and more of us are coming into contact with computers in our business and home lives. Whether it's computer-prepared bills and bank statements — or mailing addresses stored and printed by computer.

But there are still many people who are a little afraid of the computer. They don't know how it works. And, until now, they had no way of learning

about them at first hand.

Now, by buying a small computer, you can learn at your own pace and in your own home or office how to master basic computer technology. How to make computers work for you. And make sure you're not left behind in the computer race. Why should your children feel as worried about computers as we were about metrication? Let them grow up with them and not feel lost or afraid in a technological world.

But it's in schools that the revolution in small computers leaves its deepest impression. Up to now, few secondary schools have included computing in the curriculum. And when you consider that the computer industry is crying out for more and more manpower — and backing the demand with very high salaries — there's obviously a need for some form of computer education in the schools.

And if your school hasn't got round to investing in a small computer yet, the next best thing is to buy one yourself and teach your children computing at home. The basic kits can start from as little as £275, and it's a skill your children will come increasingly to rely on in their future careers.

Just as dad buys an electric train set for his son's first birthday, so dad will buy a personal computer for his children's educational advancement and his own fun.

Incidentally, research proves that computers are ideal teachers using programmed learning courses. A teacher at a blackboard can all too easily lose a child's attention. A computer can capture that attention and develop the child's growing intelligence.

Now we have to answer the second most important question. How does the ordinary household — your household — use a personal computer?

Well, right now, in the United States, you can actually go shopping by computer. And you can be sure it's an idea that won't be long in reaching British shores. All you need is a telephone and your own small computer. You telephone from home to the shop — link your own computer to the shop's computer via a device called an acoustic coupler — and simply key in your list of purchases. The shop's computer tells you whether the goods are in stock, and how much they're going to cost you. You then place the order, and leave your own computer to debit the amount from your weekly budget.

Computer shall speak into computer. In the future, that will be the key phrase. As more stores, garages, transport companies, hospitals go onto computer, it will be possible for you to link in your computer to almost any service you require for fast, accurate, up-to-date information.

On a more personal day-to-day level, computers have all sorts of uses in the home.

For instance, do you have difficulty remembering birthdays? Well, now you can record all important dates — your wife's birthday, your wedding anniversary, your last dental check-up — on a computer calendar. Make it a habit to start every week by calling for a print-out of all the events you have to deal with that week. Computers never forget — but take up less room than elephants.

But a home computer really comes into its own by helping you plan all your household budgeting. Can you afford a bigger house? A bigger car, or a holiday? Where can you make economies? Your personal computer can find the solutions by taking all your financial circumstances into account. All you do is feed in your monthly pay cheque, and your fixed expenses, and let the computer take out the drudgery and come up with the answers.

Now there's the audio cassette interface I mentioned before. Just as music can be stored on cassettes, so can computer data and programs. Similarly, just like an Ansafone in a tape recorder connection to a telephone, you can connect your personal computer to the telephone via this device.

In other words, you can let your computer dial important phone numbers; giving you, in effect, an automatic telephone system.

You would avoid the frustration of an engaged line, because the computer just keeps on trying until the line is clear.

And now for something that's really different. A music generator board which can turn your personal computer into a programmable electronic organ. You can listen to it through your hi-fi set and have your computer music beautifully amplified. The ultimate, perhaps, in electronic keyboard music.

In fact, virtually the only limits to the uses of a personal computer are your own interests. Statistics — mailing lists — word processing — a personal accounting system — personal tax computations — keeping a complex diary — designing crosswords — a switching system for central heating — storing recipes — even controlling model railways — and it's a perfect aid to learning about computing at first hand.

As we saw earlier, computers of one sort or another are going to affect our lives over the next few years — it makes sense to make friends with them now.

But it's in the field of small businesses that personal computers are enjoying their biggest success. The cost of a complete system for a small business is about £2,500. With this equipment, plus some programs, the computer takes care of all the invoicing and accounting the average small business needs. In addition, with some word-processing programs, it can take over the problems of typing and sales letters. A personal computer linked to an electric typewriter can become a fully automatic typewriter, typing hundreds of sales letters faster than any human typing pool — and with no margin for error necessary.

The personal computer has become an "exploding consumer product" in the USA. Already, there are more than 300 personal computer shops catering to the American home computer user. And to cope with a demand that is literally worldwide, nine major personal computer journals, including one published here in the UK, are in regular publication.

Personal computers have become a social activity. All sorts of people are clubbing together to exchange ideas, swap programs, and show off their various achievements. In some clubs, all the members chip in a weekly subscription to buy their personal computers. A lot of the fun lies in buying the computers in kit form and making them up.

All you need is a soldering iron and a bit of patience. The instructions are clear and easy to follow, and any do-it-yourselfer can create his own computer system from kits. Engineers will test the finished system and pinpoint any faults.

And those are two qualities tested to the full by the most popular of all computer activities — computer games.

Now, when we speak of computer games, most of us think of ultra-fast games of chess played on electronic chess-boards. Some of you will remember the scene in the film '2001' in which an astronaut challenged the spaceship's HAL 9000 computer — and was soundly thrashed for his presumption.

But there are many more games which can be played on a computer — all of them challenging —

all of them stimulating — and all of them developing your computer skills.

One of the most fascinating is CIVIL WAR — a simulation of the battles in the war between the American States. It can be played by one or two players.

The players are told by the computer which battle is about to commence, the original date, and the outcome. The computer also processes the statistics of the number of men at each player's disposal, the money available, and the rate of inflation. Though perhaps it's not surprising how few wars were ever subjected to economic cut-backs.

One player becomes the confederate general, the other is promoted to command of the Union army. Each general keys in how much he is going to spend on food, pay, and ammunition. The personal computer then calculates the morale of the troops — a very important and sometimes decisive factor.

The computer then advises the generals who is attacking and who defending — nothing as mundane as the toss of a coin — and asks for a strategy. This is where the human skill comes in. From a wide range of strategies, each general chooses the one which he thinks will win him the day. Once each general inputs his strategy, the computer computes the casualties.

Naturally, the side which sustains the least casualties is the victor. The computer — like some ancient Greek god — then moves to another battlefield.

There are literally hundreds of other games you can play on the computer. Some are merely adaptations of well-known board games. Others are designed specifically for the computer.

Games like STOCKS, or how to play the Stock Market without going bankrupt — or POLICY in which you get the chance to manage a country's economy; and the chance to start again when you've utterly ruined it.

Games of this type become a powerful educational tool in teaching practical economics or the workings of the stock market, whether it be in the secondary schools or colleges of further education. POLLUTION was a game specifically developed with students in mind. Here, the players create a water pollution situation, and are then asked to clean it up.

And that's where the importance and genuine satisfaction of computer games lie. They are simulated real-life situations which encourage imaginative and constructive responses. The results of the games are relatively unimportant — we take risks, tolerate uncertainty, and proceed with less than complete understanding. Our abilities to react creatively to new and unexpected situations are constantly changing. We develop our decision-making capabilities and problem-solving skills.

A computer waits patiently while we experiment with different lines of thought. It's possible to play any game we choose, as often as we'd like.

You leave the computer to take care of the complicated book-keepings, while you create the initial conditions, manipulate the parameters, and analyse the results. The effects of each parameter can be isolated; the simulation can be repeated as often as you wish.

But, above all — computer games are fun!

Finally, let's quickly look at the more practical uses of personal computers around the home. You'll see that computers can be a real investment that more than pay for their initial cost in terms of convenience and money-saving ideas.

First, it can actually help you reduce your energy consumption, optimise increases and decreases in the inside temperature by controlling the central

heating system. A computer can be programmed to open and close curtains or windows to use the sun's energy — or keep it out.

A computer can also be used as an effective security system, by monitoring the opening and closing of windows or doors — especially important if you're out of the house. If an intruder breaks in while you're away, the computer can automatically telephone the police with a recorded message.

Fire monitoring equipment can be located in many places around the home or office, to sound an alarm long before your nose detects the first trace of smoke. Again, the computer will automatically alert the fire brigade with a recorded message.

Computers can help reduce your food bills and ensure you always have a plentiful supply of fresh food by selecting foods on a seasonal basis. Programs can be used for selecting locally produced foods at the right times, or to guide you to the best and cheapest buys.

It can also be used to write menus in order to increase variety or maximise nutrition.

With a shopping guide helping you to take advantage of local food supplies by indicating the best of one or two markets from which to buy.

Remember that stock market game I mentioned earlier? Well, here's the real thing. Hook your personal computer into the Stock Exchange or the Financial Times' computer and use their stock data base to collect the share information you require. You could even program your computer to manage your portfolio, buying and selling at predetermined levels.

And why buy newspapers when you can hook your own computer into Fleet Street and get the

news as it happens? You just program your computer to search for the news you're interested in, and then print it out on your terminal.

You can link up with your local university or college computer and learn a subject in the comfort of your own home. The necessary information and study material will be fed in from the university's computer to appear immediately on your screen.

May I remind you of something I said at the beginning. These are not science-fiction dreams. These are possibilities which are happening now.

Let me finish this brief guide to the world of personal computers by quoting the words of the editor of "Personal Computer World". He said this . . .

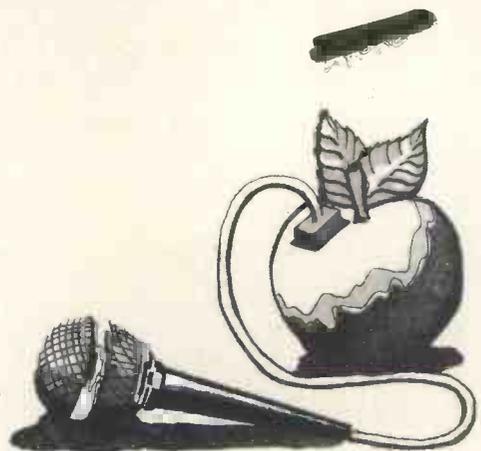
"The world is full of divisions: rich and poor, north and south, developed and underdeveloped, literate and illiterate. Soon, if we do not make a great effort, there will be another division — between those of us who are aware of the uses and the power of the personal computer, and those who are not. It will be the difference between surviving the information tidal wave and being drowned by it. If we cannot handle the dizzying rate at which the world is changing there will be those who will step forward and handle it for us — at a price which will increase at the same dizzying rate. At last science and technology have given us a product which isn't big, noisy, greedy for energy and expensive. It extends personal freedom, personal control, personal knowledge, personal enterprise — without depriving others."

Ladies and gentlemen — welcome to the world of personal computing.

## Apple would like you to say a few words!

Apple II, the only personal computer with colour,  
now has voice recognition!

Apple II comes in a portable, 10 lbs, zip case and is the ultimate personal computer, with powerful colour graphics, extended Basic, 81/0 ports, 1000 cps. display, 8K ROM, up to 48K RAM, 350ns memory access, mini assembler and much more!



Now available in the Midlands from Dr. Tim Keen, Tel: 0602 412214.

**Keen Computers Ltd.** 58 Castle Boulevard, Nottingham.



**NEW**

# 56-STATION ASCII KEYBOARD

## NOW IN STOCK — ASCII KEYBOARD MODEL KB756



**FANTASTIC VALUE AT £60**

+ £1.50 p&p + 8% VAT (mail order total £66.42)

<b>SELF-SCAN ALPHANUMERIC PANEL DISPLAY</b> 16/18 position display with 64 character repertoire, 5 x 7 dot matrix. Input 6-bit BCD-code, power requirements +5v, -12v. Character size 0.40" x 0.28". Overall dimensions 8 1/4" x 2 1/4" x 1 1/4". Supplied with full technical data. Price £55.00 + 75p P&P + 8% VAT (Mail order total £60.21).	<b>BALL MIRATEL VIDEO MONITOR</b> 9" diagonal P4 phosphor tube. Bandwidth 12 MHz (3dB). Input voltage 220V 50/60 Hz 24W. Output voltage +15V DC (short circuit protected) +12kV DC; 12.6V rms. Supplied complete with high & low voltage power supplies, amplifier, and attractive moulded plastic housing including space for keyboard. Price £95.00 + carriage + VAT.
--	--

### SPECIAL PURCHASE - MITE

## 123P Alphanumeric printer mechanisms



**BRAND NEW SURPLUS ONLY £75**  
+ £3.50 P & P + 8% VAT (Mail order total £84.78)

Solenoid-operated page printer using standard reversible typewriter ribbon. Prints standard 64-ASCII character set on 8 1/2" paper (80 characters per line, 6 lines to the inch). Maximum speed 11 cps. Power requirements 115VDC. Compact, light-weight unit 9 1/2 lbs, 12" x 9" x 2 3/4". Supplied complete with full technical manual.

We also specialise in: DEC minis - PDP8 and PDP11 processors, add-on memory, peripherals and spares. Hard copy terminals - ASR 33 and KSR 33 Teletypes, Data Dynamics 390, Texas Silent 700. Send for complete lists.

### HAZELTINE VISUAL DISPLAY UNIT

- \* Teletype Compatible
- \* 12" Diagonal Screen
- \* TTY Format Keyboard
- \* 12 lines of 80 characters
- \* 64 ASCII Character Set
- \* 5 x 7 Dot Matrix
- \* Switch-selectable Transmission Speeds up to 9600 baud
- \* Switch-selectable Parity
- \* Standard CCITT V.24 Interface



**MODEL H-1000 PRICE £350** + carriage and VAT

Also available: -  
**Model H-1200:** Specification as for H-1000 except 24 lines of 80 characters displayed. Price £425.00 + carriage + VAT.  
**Model H-2000:** Buffered/Editing model with direct cursor addressing, dual intensity video, and detachable keyboard with separate numeric and edit clusters. 27 lines of 74 characters. Price £495.00 + carriage + VAT.

A copy of trading conditions supplied on request

**ELECTRONIC BROKERS LTD.**  
 49-53 Pancras Road, London NW1 2QB.  
 Tel: 01-837 7781. Telex: 298694.

Main distributors for IMSAI products, reliable low cost, Micro Computers, cassette or multi-floppy disk based systems. Ideal for all business, industrial, medical and educational use. Simple and quick to learn programming with BASIC, FORTRAN, and ASSEMBLER languages.

IMSAI 8080 computer - 10,000 sold in 18 months - the world's fastest and most successful selling machine. Flexibility that everyone can afford.

IMSAI latest product, the BDP80 with 32K or 64K memory, 1 million alphabetic characters of disk memory and loaded with lots of super features.

Designed for you with thought.  
Join the successful people - buy IMSAI.

Powerful programmes are available - designed so that you can tailor them to your needs without programming skill.



Profit/Loss accounts, Balance Sheet, Sales Ledger, Stock Control, Invoicing, Purchase Ledger, Budgetary Controls, Postal Lists, Production Control, and Word Processing.

All products are competitively priced for the finest budget. For details call or write:  
 Mark Proudfoot  
 0723-63638 or 67027

We are looking for suitable local and area distributors. Join our successful team supported by successful products.  
 IMSAI  
 The standard of excellence in Micro Computers.

P.s. All parts are guaranteed for 6 months, we have a repair service to back up our products.



**Belvedere Computer Services (Scarboro) Limited**  
 9 Belvedere Place . Scarborough . North Yorkshire . YO11 2QX  
 Telephone 0723 63638 (67027 - 24 hours)

The Exhibition with the difference

# PERSONAL COMPUTER WORLD SHOW

AT THE WEST CENTRE HOTEL LONDON S.W.6  
ON 21, 22, 23 SEPTEMBER 1978

## FEATURING

Systems for small businesses  
Computers in Education  
Computers in the home  
Amateur computing

## ATTRACTIONS

Competitions  
Best home brew system  
Best software  
Best school applications  
Best home applications

## PLUS

Famous Personalities  
Innovators corner  
Video games  
PCW Micro chess championship  
Seminars, PCW consultants in attendance

Three days to remember, Seminars:

**Thursday 21st: Small Computers and Small Businesses**

**Friday 22nd: Small Computers in Schools and Universities**

**Saturday 23rd: Computing for the Beginner and Hobbyist**

Readers interested in exhibiting, attending the seminars or visiting the Exhibition, please write for further details to: **Will Martin**, Interbuild Exhibitions Ltd., 11 Manchester Square, London W1M 5AB. Tel: 01-486 1951

# compelec electronics

Welcome you  
to the  
Altair Age



THE MICROSYSTEM FOR THE PROFESSIONAL  
(System 9V illustrated — £6,781 to £7,026)

We supply a complete range of low-cost Altair systems and software for educational, industrial and business use.

#### **Main Frames**

Choice of front panel or Turnkey

Memory capacity to 64K bytes

#### **Bulk Storage Devices**

10 megabyte disc system; standard floppy disc system; mini floppy disc system or cassette.

#### **Interfaces**

A variety of plug-in analogue, digital and industrial control interfaces supplied from stock.

#### **Terminals**

Soroc VDUs, Centronics and Qume line printers

#### **Software**

FORTRAN, BASIC, multi-user BASIC, disc operating systems, assemblers, editors, monitors, business packages, word processing packages, etc.

We supply fully commissioned Altair computer systems complete with terminals and software.

Telephone 01-328 1124 for a Demonstration

## compelec electronics limited

Compelec Electronics Limited, 107 Kilburn Square, London NW6. Tel: 01-328 1124

# COMPUTER WORKSHOP

## BRITAIN'S LONGEST ESTABLISHED COMPUTER SHOP

### HARDWARE

COMPUTER MP-68 with 4K Memory & Control Interface Kit £275.00 Assembled £330.00  
 MEMORY MP-4K Memory Kit £70.00 Assembled £90.00 MP-8 8K Memory Kit £170.00 Assembled £195.00  
 DISC SYSTEM DMA1 1.2 megabyte twin system Kit £1450.00 Assembled £1525.00  
 MF-68 160K byte twin minifloppy Kit £800.00 Assembled £860.00  
 TERMINAL ASR43 Upper/Lower case 132 Char. 30 cps KSR Assembled £915.00  
 CT-64 Upper/Lower case visual display terminal Kit £230.00 Assembled £315.00  
 CT-VM Video Monitor for CT-64 Assembled £140.00  
 PRINTER RICOH RP-40 Daisy wheel printer 30 cps Assembled £1800.00  
 CENTRONICS 701 Dot Matrix printer 60 cps Assembled £1400.00  
 PR-40 40 column tally roll printer 55 cpm Kit £200.00 Assembled £250.00  
 CASSETTE AC-30 Cassette Interface for 2 audio cassettes Kit £65.00 Assembled £100.00  
 GRAPHICS GT-61 Graphics Terminal with P197 Power Supply Kit £92.50 Assembled £120.00  
 SUNDRIES MP-R EPROM Programmer for 2716's Kit £36.00 Assembled £43.00  
 MP-N Calculator Interface Kit £37.00 Assembled £44.00  
 MP-T Real Time Clock Kit £32.00 Assembled £40.00  
 MP-L Parallel Interface Kit £30.00 Assembled £37.00  
 MP-S Serial Interface Kit £30.00 Assembled £37.00  
 PPG-J Joystick Kit £32.00 Assembled £40.00  
 MOD1 UHF Modulator to link CT-64 to T.V. Kit £4.50 Assembled £6.00



PRICES DO NOT INCLUDE V.A.T. or carriage.



### SOFTWARE

Software includes 3K†, 4K and 8K Basic\*, Coresident Editor/Assembler, Text Editing System†, Text Processor†, TSC Assembler†, Games and Scientific packages.  
 \* Disk Based versions available.  
 † Full source listing included.  
 Prices vary between £5 and £20.

Both Disk Systems come complete with the comprehensive 'Flex' Disk Operating System. Commands include -

APPEND	COPY	LINK	RENAME
ASSIGN	DELETE	LIST	SAVE
BACKUP	EXECUTE	MONITOR	STARTUP
BUILD	GET	NEW DISK	VERIFY
CATALOG	JUMP	PRINTER	VERSION

**Applications include:** • Travel bookings • Word Processing • Questionnaire analysis •  
 Process control • Teaching • Accounts analysis  
**Existing customers:** Schools - Universities - hospitals - M.O.D. - business -  
 industry.

Please write for full details to:-

COMPUTER WORKSHOP, 38 DOVER ST., LONDON W1  
 Tel 01-491 7507