2/1

PERSONAL COMPUTER WORLD

50p U.S.$2.00

SMALL COMPUTERS FOR SMALL ORGANISATIONS

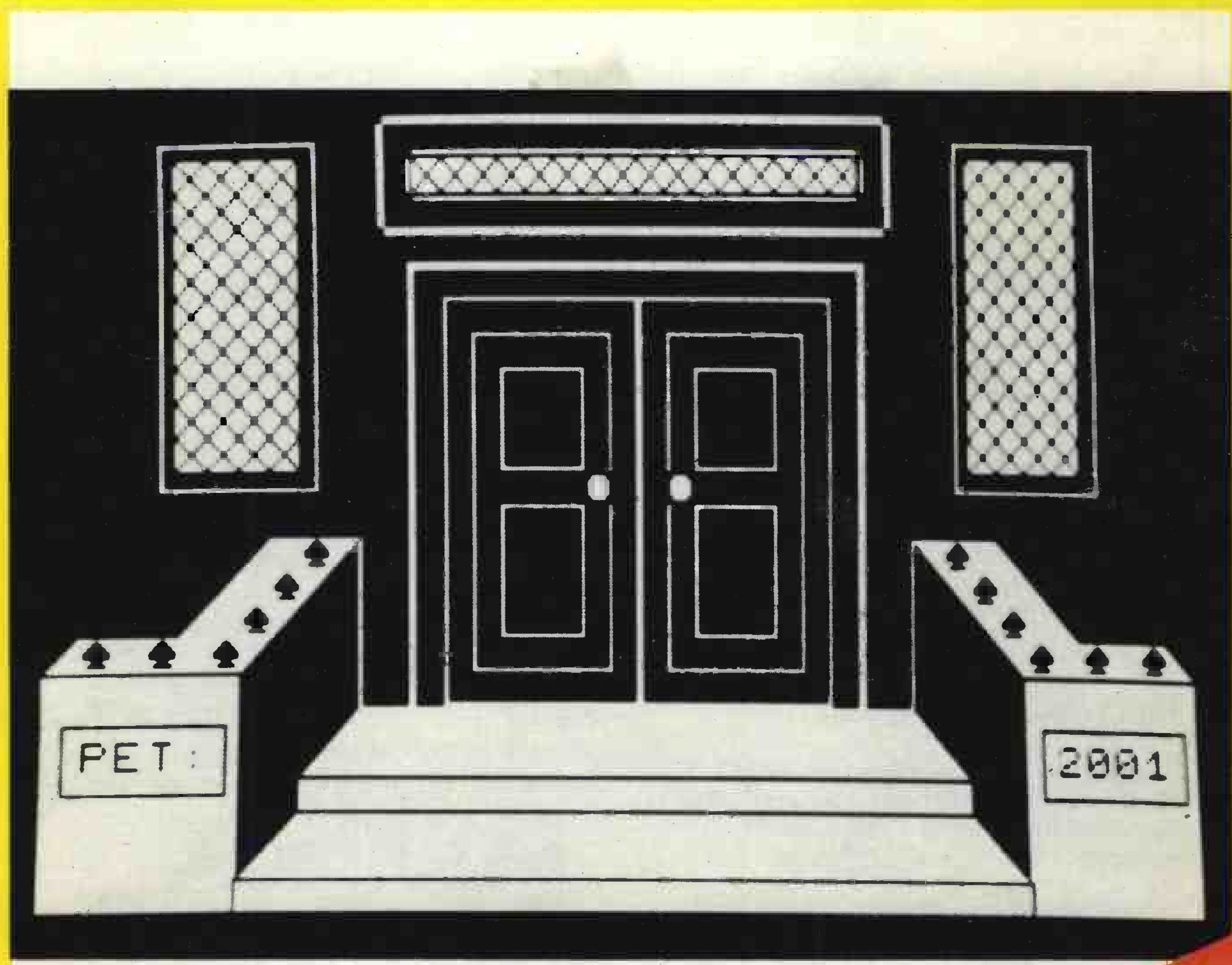
MAY 1979

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

Publishing and the Damned

The Sorcerer's Wand

The Buyer Bites Back



'In Defence of Pet'

PCW Chess Championship
- £1500 PRIZE

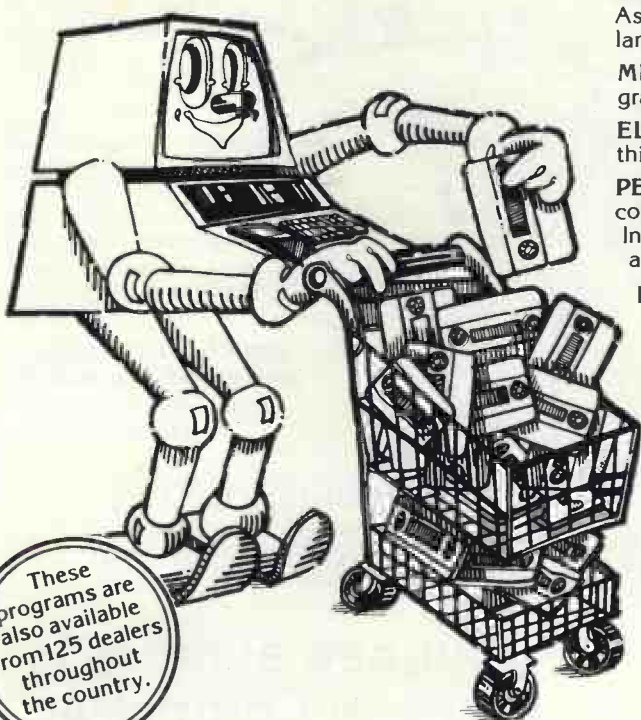
Pet program supermarket

At last a whole menu of programs to feed your PET at prices which knock the bottom out of traditional software costs.
Our 16 page catalogue lists nearly 130 programs from £3 to £50 (including VAT).
These cover Business Routines, Programming Aids to help you make the most of your PET and some super games to play with it. Here are just a few examples.

MAILING LIST—£15. Stores and prints names and addresses; carries out labelling etc.

INVOICING—£20. Stores pricing information and prints numbered invoices. Handles differing VAT rates, and at least 50 items on 8k PETS; more with additional memory.

WORD PROCESSOR—£25. Composes, edits and prints letters, articles, leaflets etc. Script directives include line length, left margin, centering and skip. Edit commands allow user to insert, delete and move lines, change strings, save and load from cassette, move up and down, print and type. Addresses any RS-232 printer through special adaptor, available separately.



These programs are also available from 125 dealers throughout the country.

V.A.T. PACK—£17.50. Two programs, one for output VAT and one for input VAT. Allows data to be entered, processed and stored on tape. Information, including VAT return, can be displayed at any time. 'VAT SCHEME D' is also available for retailers, price £15.00.

STOCK CONTROL—£12. Stores up to 150 items per tape file, under stock number, description, suppliers name and address, and stock level. Auto re-order warning.

MERGE—£8. Joins any program or section to another.

PET BASIC TUTORIAL—£15. Let your PET teach you to program in BASIC. A complete course.

ASSEMBLER/EDITOR—£25. One and two pass Assemblers, Editor and Executor translates assembly language programs into machine code for direct execution.

MICROCHESS—£14. Play chess against PET. Excellent graphics. A superb piece of programming.

ELIZA/DOCTOR—£10. PET converses "intelligently" in this famous simulation of a consultation, with a psychiatrist.

PET WORKBOOKS—£15. Five workbooks provide a comprehensive introduction to PET programming. Includes many examples and exercises. Now Volume 6 available, price £3.00.

For further details of these and the other 110 programs in our free catalogue, complete the coupon or call us today.
We also accept credit card orders over the telephone.

COMING SOON Programs for Apple and TRS 80

Petsoft Specialists in personal computer programs,

5-6 Vicarage Road, Edgbaston, Birmingham B15 3ES.
Tel: 021-454 5348 Telex: 339396.

Petsoft A member of the ACT Computer Group.

PET is the trade mark of Commodore.

To Petsoft 5-6 Vicarage Rd., Edgbaston, Birmingham B15 3ES.
MY PET NEEDS FEEDING Please rush me a copy of your free 16 page catalogue of PET programs without obligation.

My name is

My PET lives at:

PERSONAL COMPUTER WORLD

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

UK 50p
US $ 2.00

Vol. 2 No. 1
May 1979

ISSN: 0142 0232

CONTENTS

PUBLISHER'S LETTER 7

EDITORIAL 7

LETTERS 9

TIDBITS 12

THE FACES BEHIND THE PLACES 15

PUBLISHING AND THE DAMNED *John Kemp*
The microprocessor: miracle or disaster? 17

SMALL COMPUTERS FOR SMALL ORGANISATIONS *Mervyn J. Axson*
First part of a user's practical application 19

ON YOUR MARK, GET SET *Clifford Clark*
A diagnostic for the MK14 23

CHESS PROGRAMMING *David Levy*
An international chess master's hints 25

THE BUYER BITES BACK *Brian W. Haines*
PCW's Contribution to a sane dialogue between seller and buyer 29

THE SORCERER'S WAND *Dr. R.J. Beynon*
The wonderful wizard of Graph 31

SMART 1 *Mike Dennis*
A clever add-on 35

IN DEFENCE OF PET *Tyrone Crudis*
Tyrone claims that this little computer has done him a power of good. 42

ON THE LINE *David Hebditch*
Continuing the series on computers and personal communications 47

PCW OPEN PAGE *Mike Lord*
The most popular page of its kind 49

MICROS AND SMALL BUSINESSES *D.R. Worsley*
Another sober assessment of the impact of small computers on small businesses 51

TIDY IT UP *I. Addinsell* 54
A TRS-80 renumbering program ....

PARKINSON'S REVAS *D.W. Parkinson*
From object back to symbol 59

MUGS AWAY *Sheridan Williams*
Here's another excuse for going down to the pub 67

STATPACK *Colin Chatfield* 70

3D NOUGHTS AND CROSSES *Steve Thomas*
Another program for a very popular game 72

Buzzwords and the Book Review page will appear in the next issue.

Editorial and Advertising Office:
62A Westbourne Grove, London W2
Phone: 01-229 5599 (Publisher)
01-727 8758 (Editorial)
01-229 5589 (Advertising & Subscriptions)

Publisher:
A Zgorelec
Editor:
Meyer N. Solomon
Editorial Assistant:
David Bannister
Policy Adviser:
Peter Crofton-Sleigh, FRAS

Consultants
John Coll, Mike Dennis, Neil Harrison, Charles Sweeten, Patrick Sutton, Michael James, R. W. Davy, David Hebditch, Sheridan Williams, Stephen Castell, Dr. D.J. Hand.

Art: Sauveur Laurent Sant, Kathryn Hamme
Secretarial:
Vanessa Blackburn Kiddle
Layout Consultant:
David G. McFarlane
PCW Photography: Yoshi Imamura, Peter McGee
Typesetting & Artwork: Gilfillan
Cover Photo: Kaveh Khan

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 withthe policy outlined above.

Published monthly by Intra Press, 62A Westbourne Grove, London W2. Phone 01-229 5599. 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 Keats Printing Limited, 12 Fouberts Place, London W1V 1HH. Sole UK Distributors Seymour Press Ltd., 334 Brixton Road, London S.W.9., England.
Distribution to specialist shops by Intra Press.

Editorial

Whose side are we on?

In our *Tidbits* and *Faces Behind the Places* pages, we put products and the people in the business in front of the reader's eye, to personalise and, quite frankly, to support and publicise computing in the home and small business.

In the Letters page, in articles on small business computing, and in articles like *The Buyer Bites Back*, we show that we do not wear rose-tinted spectacles.

Very plainly, we are on the highway of good sense and co-operation.

Acknowledgements, etc.

Photographs of those people at stands in "Faces Behind the Places" were taken at Microsystems '79, organised by *Iliffe Promotions.*

The article in PCW, January 1979, p.12, is by *Barry G. Woollard*, the one on p.19, same issue, by *Ron Wallace*.

Photographs of MK.17., p.25, PCW February 1979 is by *Paul Bannon*. Also *Clifford Clark* has pointed out the following errors in the same article: Letter S = Figure 5 = 6D. Letter Z = Figure 2 = 5B, figure 2 (Diagnostic Program) – KEY DOWN 7F, BF, EF, DF.

In March (Vol. 1, No. 11). "Turning the Tables".Incorrect listing. Corrections: INT; =; STO 0; X; 12; +; 1; = ; int; STO1; INV SUBR.

MINE FIELD listing March Issue line 650; end of line should read – 'goto 1070'

PUBLISHER'S LETTER

Dear Reader,

We are thinking of taking a party of at least fifty people to visit one of the best American Shows – the National Computer Conference & Personal Computing Festival, being held in New York on June 4 - 7 this year. The plan is to charter a plane and hotel accommodation at a rate below that which an individual would have to pay. Everything depends on how many people respond, and how soon. If you're interested, send a letter to me at my office at 62A Westbourne Grove, London W2.

I did ask you to watch this space!

Subscriptions

When PCW started publication, we had a special six-issue offer. When these subscriptions expired, we sent out reminders.
The renewal rate was 70%!
PCW reader loyalty is becoming a byword in publishing. If you're having difficulty in obtaining PCW at your newsagent, take our subscription. You can find the details at the foot of P.3.

OHIO SCIENTIFIC

Microcomputers from the world's largest full-line manufacturer

The C2-4P Mini Floppy

20K RAM
Basic + Assembler
Personal, Games, Small Business & Educational Disks
90K Mini Floppy Storage
Printer Interface
OS 65D V·30 Operating System
Only £1595.00 Complete + VAT.

Economic expandable systems with good disk based software, available now.
See your nearest dealer for full price list and catalogue.

Abacus Computers Limited

Other systems available include the C3 OEM with 32K RAM, 512K of disk storage and BASIC as standard, £2950.00 + VAT. (FORTRAN and COBOL available as extras.) All dealer enquiries direct to Abacus Computers Limited.

62 New Cavendish Street
London W1 Tel: 01-580 8841

Mutek
Quarry Hill, Box Corsham
Wiltshire SN14 9HT
Tel: 0225-743289

Thames Personal Computers
13 Wilmot Way Camberley
Surrey Tel: 0276-27860

Linn Products
235 Drakemire Drive
Castlemilk Glasgow
G45 9SZ Scotland
Tel: 041-634 3860

U Microcomputers
PO Box 24 Northwich
Cheshire CW8 1RS
Tel: 0606-75627

Letters

PUZZLE DAZZLE – No. 3
My last puzzle (PCW Nov 78) was purely academic. This one is of important practical value and to some extent reveals a weakness of many microprocessor designs.

Using the 8080 for illustration, if one wishes to pass control to a label ALTB if the contents of the accumulator A are less than the contents of B where A and B contain unsigned numbers one merely writes

```
CMP B        ; Compare A with B i.e. set the flags to the
               result A-B
JC   ALTB ;  ; Jump if A<B i.e. if carry flag set.
```

If however, A and B are regarded as signed numbers this does not work. The problem is: produce minimum code for an 8080 (or Z80 if it helps) that sets or resets one of the flags if and only if the contents of A are less than B, where A and B are signed numbers. A,B and all the other registers must keep their original values and only the stack memory may be used for temporary storage. My own solution, in 9 bytes, is enclosed.

David C. Broughton
38 Westbury Road, Northwood, Middlesex, HA6 3BX.
PCW *Readers, send your solutions to David Broughton. First three correct solutions receive £5 each. Result of Puzzle Dazzle No. 2 in next issue* PCW

The Tea-break to terminate further correspondence
With reference to Mr. I.W. Morrison's letter in the March issue of P.C.W., I find that my tea-break is a mere 26 seconds, just enough to stir my tea! Seriously, though, I have tackled the problem in what appears to be a different way. The program takes 48 steps on my PR 100, and takes, to find the square root of 99,999, just 26 seconds. This program also uses the Newton-Raphson method and is more accurate than my square root key!
Stephen Gwilliam (14)
22 Manning Road, DROITWICH, Worcs.

The Case of the Missing N
In the Contents of your February 1979 issue, you announce an article "THE BYTE CONSERVER . . . 49", but on page 49 the article is headed "THE BYTE COSERVER." Well, I suppose that's one way of doing it.

On a more serious note, may we have a "Diary" feature in PCW – forthcoming exhibitions, conferences etc? I only know of the existence of Microsystems 79 because of a passing reference (with no dates) in the February "Publisher's Letter". Thanks to the helpfulness of your staff when I telephoned, I discovered it is actually on today. Otherwise I should almost certainly have missed it.
E.W. Huggins,
1 Hampden Close, Grays Park Road, Stoke Poges, Bucks. SL2 4JF
PCW *A large list of exhibitions will be published in the next issue.* **PCW**

Will the computer of the future be made of Jello?
The front cover of the March issue (deliberately or by inadvertence?) produced a rather disconcerting optical illusion for the early-morning reader-in-bed: if you shake the magazine around a little in a dim light, the green letters "On The Line" seem to wiggle around on the page, like a bit of lime Jello – presumably the eyes fix on the black "unravelling" and blue "Minefield".
Langdon Proctor,
Johannesvej 26, 2920 Charlottenlund, Danmark.
PCW *The ultimate compliment to a magazine – bedtime reading.*

A rash of designs?
I feel I must comment on Mr. B.A. Martin's letter in your February issue. In it Mr. Martin says that he would like to see a Z80 based system with an 80 characters per line TV modulated output. The bandwidth restriction when putting characters on a TV screen is not at the modulator (as suggested by Mr. Martin) but is instead at the i.f. stage. The simple solution to this is to obtain a second hand TV set and bypass the i.f. stage feeding the signal from the computer's display board directly into the video stage of the TV set. This can be dangerous due to the high voltages present in a TV set and therefore should only be carried out by someone with TV experience. Your local TV repair shop will probably be able to help with this modification.

The choice of the Z80 for a home system at this time is perhaps a little rash because of the range of 16 bit processors making their appearance. A better choice would probably be the Z8000 which is due to be available in the next few months. This processor has such features as sixteen general purpose 16 bit registers, 32 bit manipulations and calculations (including multiplication and division) and eight different addressing modes. Furthermore 8080 and Z80 programs are easily adapted to work on the Z8000.

Perhaps the letters pages of PCW could act as the centre in which the specifications for a personal computer of the next few years are drawn up by collating readers' personal opinions and suggestions.
P.F.T. Tilsley,
51, Atherstone Rd., Loughborough, Leics. LE11 2SH.
PCW *Good idea – but space is at a premium* PCW

A micro mystery
I own a NASCOM 1 which has the B.BUG on board. Accidently I discovered that it was possible to store and retrieve text but, from a *limited* amount of memory.
Mr. K.S. Swainson,
9 Brayton Crescent, Highbury Vale Estate, Bulwell, Nottingham NG6 9DZ Telephone: (0602) 751742
PCW *So far, the writer has been unable to unravel this mystery and appeals to readers for help* PCW

And Babbage begat . . .
Could I make an appeal for co-operation on a project through your magazine? I am, as an amateur genealogist, working in some of my spare time on transcribing and indexing source material. The indexing system which I have used involves copying the relevant information on to slips which are subsequently sorted into alphabetical order and then typed out. Clearly a laborious process in these days of personal computers! I am currently working on the surname index of the 1851 census of various North Devon parishes and the population of these places varies from a few hundred to about five thousand, although I do know of others who are working on indexes for cities which are clearly much larger.

Would any of your readers who have a microcomputer with sufficient storage and printing facilities be interested in helping? Clearly somebody living in or near West London would be most convenient, but I will answer any letters which are sent to me at my address.
David M. Squire, 6 Downside Crescent, London W13 0BQ.

FERMAT'S LAST THEOREM – YOU HAVE BEEN WARNED!
D.C. Chown refers to "the enormous problem of finding integral solutions of $x^n + y^n = c^n$", PCW March 1979. I hope that the following notes will prevent readers from many hours of fruitless experimentation.

Conjecture: $x^n + y^n = z^n$ has no solutions in integers other than those for which $xyz = 0$.

It is shown in Chapter 29 of Diophantine Equations, L.J. Mordell, Academic Press 1970 that only prime values, p, of the exponent need be considered.

J.L. Selfridge, C.A. Nicol and H.S. Vandiver. Proc. Nat. Acad. U.S.A. 41(1955), 970-973 gave a proof for $p < 4002$; this was extended by J.L. Selfridge and B.W. Pollack, Notices Amer. Math. Soc. 11 (1967), 97 (Abstract 608-138) to $p < 25000$.

There have certainly been extensions of these theoretical results over the past twelve years, so experimentation is unlikely to be valuable.
M.R. Mudge,
Lecturer – Department of Mathematics, The University of Aston in Birmingham.

Ah, the miseries of miskeying!
I have only recently been seduced into the computer sphere and therefore I am a little behind in my reading. An interesting article in the July '78 issue of PCW on Pseudo-Random Number generation by C. CLARK has prompted me to write concerning miskeying either within the infinite loop or within certain other programs after interruption.

Anyone using the Sinclair Cambridge Programmable for statistical calculations will know the anguish of miskeying that results in a blank display that refuses to reappear unless the calculator is turned off and on again. After many such errors on my part I decided to indulge in one of my many vices, to whit, press every key in sight and see what happens!

Please find attached therefore a table containing some new interrupts to be used in event of miskeying after normal interruption. I say *some* new interrupts, because the more I play around with the calculator (and I can hear the purists yelling 'Press more than one key at a time?!!!') the more interrupts there appear to be. For example, if the interrupt (+) key followed by 1 is pressed, and the display goes blank, either (–) or (7,8 and + simultaneously) or (1,2, and – simultaneously) will restore the status quo – provided the correct keys are pressed subsequently, i.e., C/CE and whatever is instructed.

I have had great fun designing perfectly useless programs for my SCP, including one which enables long division to be carried out to the nth decimal place, and hope soon to own a simple computing system, finances permitting.

Although much of what I read in PCW is above my head (Chris Ward, I know how you feel) I still enjoy what I can understand; roll on the publication of PCW's own dictionary!

The normal interrupt key is the division (+) key. If any key other than C/CE or O is then pressed, the display goes blank, and the program enters a loop which, in all but two cases, cannot be interrupted. The exceptions are marked 'unchanged' in the table; this means that the normal interrupt is still operative.

Key Pressed in Error	Alternative Interrupt
0	Equivalent to pressing C/CE
1	—
2	Unchanged
3	+
4	x
5	4,5 and x pressed simultaneously
6	=
7	—
8	Unchanged
9	+, but hold the key down until display reappears
*/EE/—	x =

More interrupts do exist, but they appear to become more complex; these are the simplest alternatives I can find.

Peter G.Q. Brooks
20, Brasenose Driftway, Cowley, Oxford OX4 2QX

An offer young people shouldn't refuse

I am a keen computer enthusiast and an avid reader of your fine magazine. I own a Science of Cambridge MK14 and I was wondering whether any other *young* computer hobbyists would like to get together with me in the hope of forming some kind of Organisation.

The organisation I had in mind would be run through the post and telephone and would be an exchange of ideas in both the hardware and software fields. I have access to adequate printing facilities and I know B.A.S.I.C.

Nigel Sutcliffe (Age 16)
1 Suncliffe Road, Higher Reedley, Near Burnley, Lancashire.

Texas Story

I find the letter from L.R. Carter (January '79) a little hard to believe. I ran a program for the "square roots by addition" algorithm on my TI 58 for square root 99999. It took only 4 min 21 sec to evaluate. Even at six digit level (square root 999999) run time is only 14 minutes.

Furthermore, since the loop of this program is traversed 317 times for square root 99999, his calculator is taking approximately 37 seconds per loop, which for a thirty step program is more than 1 second per step — hardly a practical running speed. If the timing given is correct perhaps someone could explain why there is such a disparity in the speeds of two programmables of the same make and approximately the same vintage.

On a different subject, TI 58/59 owners may be interested to know that the Dsz instruction may be used on any data registers, not just memories 0-9 as stated in the manual. To do this it is necessary to substitute whichever key or 2nd function has the same op-code as the required memory address. For those addresses for which there is no equivalent code, e.g. 21, it is necessary to use the sequence Dsz STO nn (where nn is the required address), then backstep to delete the STO.

Paul Kelly,
63 Corona Road, Lee, Lewisham, London, SE12 9LU

R.E.L. Ferguson replies

Several readers have written to comment on my article, 'An Agile Algorithm', which appeared in the November issue. I appreciate the humour in the letters of Mr. Carter and Mr. Morrison and the efficiency of Mr. Marsh's algorithm, but it occurred to me that all three may have missed the point of the program. This was expressed in the first sentence which indicated that the only operation used throughout was addition, allowing an awkward calculation to be undertaken without multiplication and division sub-routines. Even the more logical approach of subtraction was rejected, as this required extra corrections of the 'decimal adjust' instruction for working in BCD.

Mr. Marsh's system (very similar to 'square root by long division', which used to be part of the arithmetic syllabus) requires subtraction and multiplication; the latter could of course be reduced to a 'shift' operation since it only involves powers of ten.

The program made no attempt to rival the Newton-Raphson iteration, which offers both speed and accuracy. Rather was it an exercise in 'applied maths', the idea originating some years ago in the School Computer Society. Using decade counters and some TTL gates a piece of hardware was built which produced (very slowly!) a two-digit root by subtracting the odd numbers. On acquiring our own micro it was suggested that the same method be tried, using software. We were all surprised to find how FAST it was!

R.E.L. Ferguson,
Cults Academy, Cults, Aberdeen AB1 9SA.

Square Roots

Concerning the square root algorithm given by Mr. Marsh in your March issue, you may be interested to know that it is essentially identical in the way which I was taught to find square roots using an abacus some 18 years ago. Although I don't remember the details, there is a similar method for finding cube roots, involving the use of another register. A good abacus instruction book may give the method.

R.F. Berthelsdorf,
59 Stoke Fields, Guildford, Surrey. GU1 4LT.

Making the Conserver Compatible

Your article, *The Byte Conserver,* by Graham Trott in the February issue, would have done well to point out that users of SWATBUG or SMARTBUG would need to modify the BADDR routine. Although this is MIKBUG compatible, at address E047, XHI is stored at A000 and XLOW at A0.0E. Unless line IC39 is changed to 'CE A0 0D', the start address will always be 'OONSB'.

Otherwise very good.

J.D. Mittell, Chemical Crystallography Laboratory,
9 Parks Road, Oxford OX1 3PD.

Every house should have one — but for what?

I have been reading your magazine since last Autumn. Writers keep using this expression "Home Computing", but no-one seems to have properly thought out what a home needs a computer for. A lot of what we are offered are either business applications, or else all those silly games.

May I suggest:

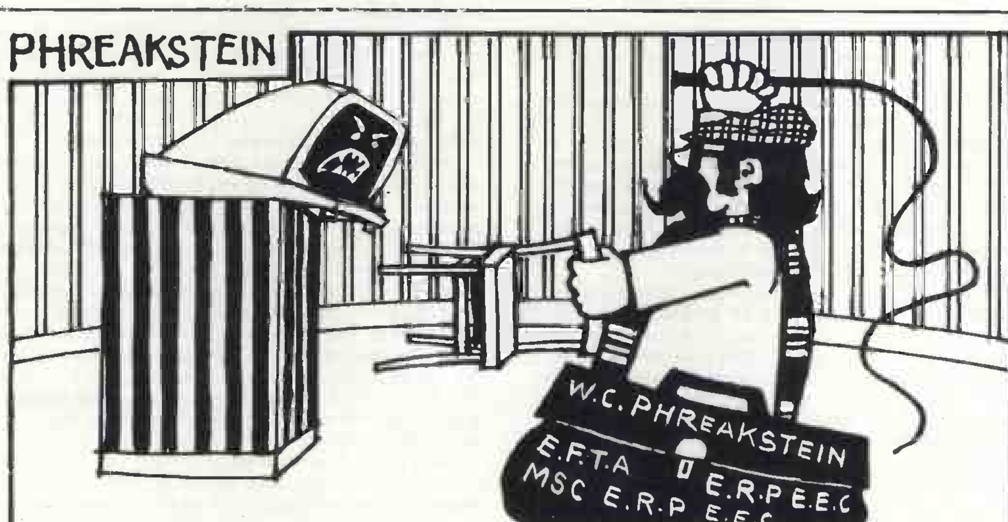
i) a home computer should be designed to be left running all the time,

ii) it should have a teletext decoder (could this be just software?)

iii) there should be a range of cheap, sinple peripheral gadgets so that the computer can run the central heating, administer fire and burglar alarms, log the telephone calls and so on.

This way it can really be a home computer and not a toy computer.

D. Simpson
64 Cromwell Road, Stretford, Manchester 32.



Tid Bits PRODUCTS . . . . COMPANY NEWS . .

Microdigital of Liverpool keeps making news. It has now announced Microdigital (Hire) Ltd., which has been set up to hire out Apples, Sorcerers, Pets, TRS-80s and (possibly) the NASCOM.

Microdigital found the best man to head the hire company in the person of Mike Maughan, who was once the NW technical representative of a large company. Mike has been given autonomy, makes his own decisions, and turned down a five figure salary to head the new division.

Mike Maughan of Microdigital (Hire) Ltd.

The hire company aims to serve an area within a fifty mile radius of Liverpool. The estimate is that "there's enough business to keep us busy for a while". For a one day hire or less, customers have to collect. For weekly hire, systems can be delivered.

One motive for hiring out machines was to give people a genuine chance for hands-on experience. Sales of systems might be delayed as a result, but Bruce Everiss, principal of Microdigital, feels that this is more than compensated for by long range customer satisfaction. "We aim to be as flexible as possible. We are offering a service, and that means the customer is always right".

Enquiries to:
Mike Maughan at Microdigital (Hire) Ltd.,
25 Brunswick Street, Liverpool L2 0BJ. Tel: 051 236 0707.

Small Software House

Tridata Micros Limited specialises in low cost Commercial packages such as sales ledger and purchase ledger, payroll, stock control and any tailored software requirements for the small businessman. It states that, subject to the needs of individual clients, the average cost of these packages is £500.

Tridata also have an advisory service for intending users in the small business and technical applications field.

The staff at Tridata is fully engaged in the business — and includes three working Directors. Enquiries to:
TRIDATA Micros Ltd.,
81 Grange Way, Rushden, Northants N10 9EW.
Tel: (093 34) 53105/59678

Megapalm open to enquiries

New company Megapalm produces and sells software for the Pet. Its director is J.R. Mace, a qualified accountant and Senior Lecturer in the Department of Accounting and Finance at the University of Lancaster.

Packages are specialised for accounting and finance. Current offer is an Incomplete Records Accounting System for accounting firms in public practice. **Enquiries to:**
J.R. Mace,
Downderry, Halton Road, Nether Kellet, Carnforth, Lancashire, LA6 1EU

Retailing the Microcomputer

An international study of the problems and opportunities in the retailing of microcomputers by non-specialist electrical retailers is being carried out by **Allteck (Technology Initiatives) Ltd.,** of 55/58 Pall Mall, London S.W.1. Tel: 01-839 3143. The research is to be carried out in Japan, North America and the EEC countries including the United Kingdom.
Contact: Peter Matthews.

New Software House's payroll package

Compsoft Ltd., specialises in MITS Extended Basic running on 8080 and Z80 based processors. It has developed a "general purpose, comprehensive" payroll system. The eight main features listed are indeed impressive. Two of them are: payments may be accumulated by department, pay point, cost code and job; up to 26 adjustment codes may be supplied per payroll. These may be applied to the whole payroll, an individual until further notice, or to an individual for any one period. Reducing balances may be supplied and payment ceases on a zero balance.

The one off price for the package is £600 including payslip format modification. A £50 p.a. maintenance contract covers the user against statutory changes and brings, free, improvements in the basic system.

Full details from:
P.N. Horgan,
Compsoft Ltd., Old Manor Farm, Chilworth, Guildford, Surrey.
Tel: (0483) 39665

This word processing machine, the TES 401, is to be marketed by British Olivetti through a selected dealer network. It is meant for the Secretary's desk, and is basically a microprocessor controlled programmable electronic typewriter. Prints at 350 words per minute, with changeable typeface. Stores 7,500 characters; above this storage is available on an erasable magnetic minidisk; there is a plasma display.

Full details from:
Jeremy Biggs,
British Olivetti Limited, 30 Berkeley Square, London W1.
Tel: 01-629 8807 Ext. 121.

Fringe Benefits

Peripheral Hardware specialises, as its name implies, in peripherals. A recent press release emphasises the advantages it offers, and gains, through specialisation. It has come up with a combination of a dumb VOU (Lear Siegler ADM 3A, "rugged, convenient, compact") and the SCT High Speed Rotary Printer (output of 2200 characters/sec, "up to the standard of those costing ten times as much"). Price is £1395. The combination gets a full screen of data (24 lines x 80 characters) printed in under one second.

Full details from:
Keith Searle,
Peripheral Hardware, Link House, Pool Close, West Molesley, Surrey. Tel: 01-941 4806

A magnificent monitor manual

A.J. Harding of Bexhill offers a Monitor (Version 3) TRS-80 monitor program written by Hubert S. Howe Jr.

A free and exceptionally well written manual comes with the Monitor 3 TRS-80 monitor program. Details of the Monitor 3 program from:
A.J. Harding, 28 Collington Ave., Bexhill-on-Sea, East Sussex.
Tel: 0424 220391

Intel Extends

Intel has well developed systems in its *Intellec* range. Now it has introduced a Disc Extended BASIC Interpreter for Intellec disc-based systems of 48K read/write memory and the ISIS-11 operating system (Version 3.4 upward).

The Interpreter is known as BASIC-80 and has a list of features and a manual one would expect from a company of Intel's standing.

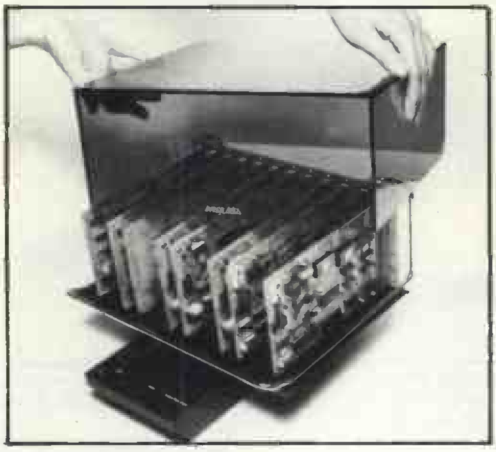
Full details from:
Intel Corporation (UK) Ltd.,
4 Between Towns Road, Cowley, Oxford OX4 3NB.
Tel: (0865) 771431 Telex: 837203

The right kind of GBH

GBH Data Services Ltd., has a highly experienced staff in data processing, strict professional control of its own work, and states that customer satisfaction is highly important to it. It can "produce software packages suited to requirements, and on time". It quotes without cost or obligation.

Address is:
GBH Data Services Ltd.,
Dumfries Chambers, 91 St Mary Street, Cardiff, CF1 1JY
Tel: (0222) 561807

Microcomputers made to a new Image
Image Data, a British owned company, has opened headquarters and a manufacturing facility in Bristol. It has its own-design microcomputer, the Image Data 8, built around a modular processor based on the 114mm edge connector, a 43/86 way bus structure, and can be configured around a Motorola 6802 and Intel 8085 (capable of being co-resident). Other PCB boards offer features such as RAM in 4K and 8K byte units, CRT display, keyboard, teletype I/O cassette from 2400 to 300 bits/sec, lineprinter output.



Full details are available from:
H.P. James, Image Data Products Ltd.,
1-4 Portland Square, Bristol BS2 8RR. Tel: 0272-40248/9
(PCW The background to this press release is interesting. The fathers of the City of Bristol strained every nerve to obtain Image Data for their City. Their success is reflected in Image Data's satisfaction. Congratulations to the City Council's Industrial Development Unit PCW)

More about Etcetera
T & V Johnson (Microcomputers Etc. Ltd) say their trade has been increasing at 20% per month. Customers are educational establishments, hobbyists, scientific entrepreneurs, small businessmen, Industry etc.

Their range now includes the Sorcerer, ITT 2020, PET, Jade, Micropolis, Shugart . . . etc. Hardware developments include add-ons, and interfaces, designed by the Technical Director, Dr. R.V. King, for the Pet and TRS-80. Software includes accounts packages for the TRS-80 . . . etc.
Full details from:
TVJ Microcomputers Etc.,
165 London Road, Camberley, Surrey. Tel: 0276 62506

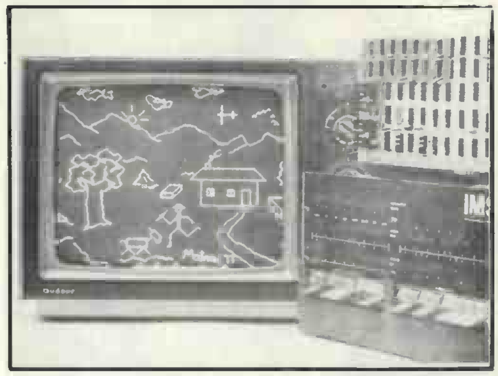
DEC — Digital Education Courses!
Digital Equipment Company now states it is able to give courses for people in organisations who are "upgrading or purchasing a computer for the first time". Courses are for beginners as well as those with previous experience.

DEC recently opened a training centre in Manchester, and says courses are immediately available.
For full details, organisations should contact:
Gerry Dabney, Training, DEC Ltd.,
Fountain House, The Bulls Centre, Reading, Berks.
Tel: (0734) 583555

Swedish System
The DataBoard 4680 is a microcomputer system for OEM (Original Equipment Manufacturers) and End-users. There are more than 60 boards available, of which 40 are interface boards. Boards are designed to accept, without modification, any one of the Z80, 8080, 2650 (from Signetics), 6502 and 6802 mpus.

Features are numerous, including Colour-Video RAM "fully Viewdata compatible" with support commands in BASIC. Can be expanded to a development system. Software is detailed and versatile. High Level Languages: A 24K, two-pass Basic, a true resident FORTRAN compiler, a PASCAL compiler.
Full details from:
Bengt E.B. Svensson,
Sattco AB, Dal vagen 10, 171 36 Solna, Sweden.
Phone 08-83 02 80. Telex: 11588

DIGITAL DESIGN GRAPHICS SYSTEM INCLUDES REFRESH MEMORY



The ALT-256**2, a 256 x 256 high resolution graphics device, plugs directly into Altair, Imsai or similar S-100 bus computers. The card contains all interface electronics, a TV sync generator and 65,536 x 1 bit refresh memory. Because of the built-in refresh memory, no CPU time is required to refresh the screen. The composite video signal output connects to any TV monitor or the video portion of a TV set. The ALT-256**2 board occupies a single S-100 bus slot and requires 4 output ports and 1 input port. Multiple ALT-256**2 cards combine to form graphic systems with grey scale or colour capability.
PRICE: £284 + VAT
Graphics boards are also available for Intel SBC80, PDP11, LSI11, and Motorola Exorciser busses with resolutions up to 1024 points. For further details of these and many other Matrox display devices contact:
SHELTON INSTRUMENTS LTD.,
22/24 COPENHAGEN STREET, LONDON, N1 0HD.
Telephone: 01-278 6273

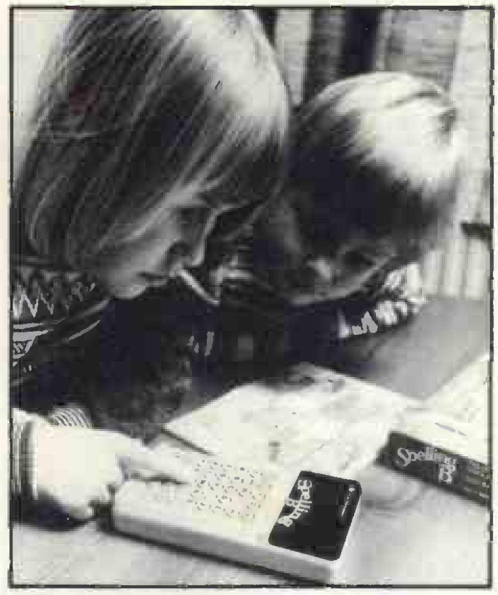
Newbear's New Bare Board: The Leader SS-50 8K RAM Board
NewBear Computing Store announce the availability of a new SS-50 bus compatible bare board. The board is thus designed to operate with the same bus architecture as that employed by the MSI-6800 or SWTP 6800 computer systems. This new product has gold plating not only on the edge connectors but over the entire board.

A complete 8K RAM kit based on this board will be available in the near future. The bare board is priced at £25 plus 8% VAT. Full details available from NewBear.
Contact:— Jon Day, NewBear Computing Store,
7 Bone Lane, Newbury, Berkshire RG14 5SH.
Telephone: Newbury (0635) 49223

Interior of long established Byte Shop at Gants Hill, Ilford, Essex.

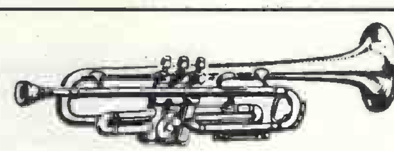
The new Byte Shop in Tottenham Court Road, London, with its first customer, John Andrews (Senior Course Tutor at the nearby London College of Data Processing).

Ithaca Audio products: among dealers in this country are Comp and Newbear Computing Store.



The 'Spelling B' electronic learning aid from Texas Instruments. This uses word/picture association techniques to help children progress in spelling.

Bazaar Gossip
Over 60 stands have already been taken for The Great British Electronics Bazaar, due to take place at Alexandra Palace on June 28, 29 and 30th. A further 28 stands have been tentatively booked. Those already committed include both large and small companies and organisations. *Personal Computer World* will be there, at Stand 31.



TRUMPET, VOLUNTARY

Responding to TIDBITS information or an advertisement?
Say you saw it in **PCW**
Blow your own trumpet.
Mention your magazine.

The Northend Keyboard, by Northend Office Supplies, "Northend", Mill Lane, Kelveton Hatch, Nr. Brentwood, Essex.

The QWERTY keyboard can be attached to the PET, requires no electrical modifications, and includes six Pet control keys with their related shift key. The control keys are in a contrasting colour.

You can bank on it
Micro-Computing in the Banking World has "finally arrived with the advent of the *Micro-Currency Dealer*".

Micro Software Systems of Stanford-le-hope, "being aware of the delays and frustration of main-frame data processing", have developed a real-time system to give same day results of foreign exchange dealings, "this giving the department Manager the information he requires to run his section efficiently".

After six months intensive trials in an International Swiss Bank Micro Software Systems feel satisfied that this suite of Programs along with its desk-top computer (the Compucorp 625) is now ready for release to the financial world.

Contact:
Micro Software Systems, 3, Stanhope House,
Stanford High Street, Stanford-le-hope, Essex.
Tel: 03756 - 41991/2

Maintaining a high profile
Rostronics, TRS-80 specialists, have grasped the simple but effective idea that business is generated and retained if customers are offered round-the-clock maintenance contracts. Beginners in small business computing have the option of seeing demonstrations "live" at the company's computer centre and receive advice. They are not only sold hardware but can order business packages "bespoke".
Rostronics are at: 118 Wandsworth High Street, London SW18.
Tel: 01-870 4805. Telex: 8813089 (Interprem)

"You've done it, — the jackpot!"

the Faces Behind the Places

Photographs *by Yoshi Imamura*

Richard Hinton *(foreground) and* **David Goadby** *of SEED, a company of vast experience. Richard Hinton looks after marketing, and David Goadby contributes his years of experience at IBM and almost unrivalled hardware knowledge. He is a* **PCW** *author (Vol 1, No. 3), a speaker at the Conference at the* **PCW** *Show, and has another article coming up in* **PCW**. *SEED deal in the Midwest Scientific Instruments (MSI) 6800 systems and are at Brownhills, Walsall. Tel: Brownhills 4321*

John Lamb, *Marketing Director of Comart. A shrewd and knowledgeable man, who, together with David Broad (Managing Director), is taking Comart steadily upmarket. Comart's range of products on offer is very wide indeed. It deals in the Cromemco and the Sol. In line with the quality of the computers, the software is extensive and includes Fortran, and Cobol. Comart are at PO Box 2, St. Neots, Cambridgeshire, PE19 4NY. Tel: (0480) 215 005.*

Mike Sterland *of Personal Computers Limited with friends (his staff). Under Mike, the Apple II has turned into a star, with numerous TV appearances. The Apple II is perhaps the most colourful of small computers and must be one of the few status symbols with usefulness to match. Well supported by its distributor and dealers. Documentation is good and getting better. Personal Computers is at 194-200 Bishopsgate, London, EC2. Tel: 01-283 3391*

From left to right, **Alan Barton, Dr Roy King, Terry Johnson** *and* **Steven Johnson,** *the directors of T. & V. Johnson (Microcomputers Etc.) Ltd. Trading as* **TVJ MICROCOMPUTERS ETC,** *they are currently retailing a very wide range of microcomputer hardware and software from their new showrooms at 165 London Road, Camberley, Surrey.*

In Dr. King, TVJ have a innovative designer who was formerly at Tandy. To be an entrepreneur means issuing orders to yourself to get in and do it. There are four entrepreneurs at TVJ.

apple II AT MICRODIGITAL

JOIN THE MICROCOMPUTER REVOLUTION BY BUYING THE BEST
APPLE PLUGS INTO A DOMESTIC TELEVISION AND CASSETTE RECORDER TO MAKE A POWERFUL, EASY TO USE COMPUTER SYSTEM

Apple II

Features

- A fast powerful integer BASIC is built into Apple II. The built-in assembler, disassembler and monitor will be appreciated by advance programmers in search of more speed or flexibility than BASIC can provide.
- Fifteen colour standard graphics in a 1,880 Point array for spectacular visual effects.
- High resolution graphics in a 54,000 point array for finely detailed display.
- Loudspeaker and sound capabilities that bring programs to life.
- Four hand control inputs for games and other human input applications.
- Internal memory capacity of 48K Bytes of RAM, 12K Bytes of ROM for big system performance in a small package.
- Eight expansion slots to plug in cards that give your Apple even more power.
- Superb, easy to follow documentation, so even a total beginner can use the machine.
- Fast (1500 baud) cassette interface.
- Proper typewriter style keyboard.

Apple II prices

Nett.	Vat.	Total
With 16K of RAM		
985.00	78.80	1063.80
With 32K Bytes of RAM		
1185.00	94.80	1279.80
With 48K Bytes of RAM		
1305.00	104.40	1409.40

Applesoft II Floating Point BASIC

An expanded version of Microsoft's popular floating point BASIC. Its 9 digit arithmetic and large library make it ideal for business and scientific programs. Applesoft II is supplied either with a cassette tape or a plug in ROM card. The tape version is supplied free with every Apple.

Apple II ROM card

110.00	8.80	118.80

Floppy Disk Subsystem

Gives your system immediate access to large quantities of data. The subsystem consists of an intelligent interface card, a powerful Disk Operating System and one or two mini-floppy drives.

Features

- Storage capacity of 116 Kilobytes/diskette.
- Data transfer rate 156K Bits/second.
- Individual file write protection.
- Powered directly from Apple II.
- Full disk capability with systems as little as 16K bytes of RAM.
- Fast access time — 600 m sec (max) across 35 tracks.
- Powerful disk operating software.
- Load and store files ;by name.
- BASIC program chaining.
- Random or sequential file access.

Floppy disk subsystem

425.00	34.00	459.00
Second disk drive and connecting cable		
375.00	30.00	405.00

Parallel Printer Interface Card

Allows you to connect almost any popular printer to your Apple. A BASIC program can produce hard-copy output as easily as it prints to the TV monitor screen. Command interpretation and printer control details are handled by the firmware built into the card, to eliminate user programming requirements.

Parallel Printer Interface Card

Nett	Vat	Total
110.00	8.80	118.80

Communications Interface Card

Allows your Apple to "talk" (though a modem) with other computers and terminals over ordinary telephone lines. Now you can load programs over the phone, send messages to remote terminals or access your office computer from the comfort of your home.

Communication Interface Card

Nett	Vat	Total
110.00	8.80	118.80

High Speed Serial Interface Card

Allows Apple to exchange data with printers, plotters and computers in serial format at up to 19.2 K Baud

High Speed Serial Interface Card

Nett	Vat	Total
110.00	8.80	118.80

Speechlab Voice Recognition Card

Allows the Apple to recognise a spoken vocabulary of up to 32 user-selected words. The computer can be programmed to perform any task desired upon recognition of a key word.

Voice Recognition Card

Nett	Vat	Total
165.00	13.20	178.20

Prototyping Card

Provides the User with a means of building up experimental circuitry for the Apple computer. The 2¾" x 7" double-sided board includes a hole pattern that accepts all conventional integrated circuits and passive components. Documentation includes a complete system bus description to aid the interface designer.

Prototyping Card

Nett	Vat	Total
18.00	1.44	19.44

Carrying Case

The Apple is truly portable and this padded vinyl, leather look case protects your Apple in transit and makes it easier to carry.

Carrying Case

Nett	Vat	Total
25.00	2.00	27.00

BARCLAYCARD VISA

accepted

Other Products

Apple maintains a 6 to 12 months technology lead over the competition. There is not sufficient space to give full details of all that is available, but the following is a sample to whet your appetite.

Light pen
Real time clock
Co-resident assembler on disk or tape
Programming aid ROM
Joysticks
PROM Burner

Apple Hire

The Apple is one of many machines from Microdigital (Hire) Ltd. For details ring Mike Maughan on 051-227 2535.

Software

We can supply application programs from a number of sources and advise you on your program requirements.

Our own software department has developed a Trade Counter program which keeps a round pounds debtors ledger in real time and advises trade counter staff when credit limits are reached.

This program is tried, tested and proven and helps reduce bad debts.

Nett	Vat	Total
25.00	2.00	27.00

Trade Counter Program (integer basic, needs 32K of RAM and a single disk)

NEWS FLASH

ITT 2020 SYSTEM

We are now dealers for the ITT version of the Apple at the following prices.

	Nett	Vat	Total
With 4K Bytes of RAM..........	827.00	66.16	893.16
With 16K Bytes of RAM..........	950.00	76.00	1026.00
With 32K Bytes of RAM..........	1114.00	89.12	1203.12
With 48K Bytes of RAM..........	1278.00	102.24	1380.24

PART EXCHANGE

Pet owners, trade up to an Apple at MICRODIGITAL. We can allow up to £300 for your old PET against the cost of a new Apple II

MICRODIGITAL

To provide a better service
we have re-organised ourselves.
The following remain the same:—
Our shop is at
25 Brunswick Street, Liverpool.
Telephone order number
051-236 0707
The following are changed:—
Mail order, software, engineering,
accounts, etc., are now at :—
14 Castle Street, Liverpool.
Telephone (except telephone orders)
051-227 2535/6/7.

An operator working at an input keyboard. Under the Mirror Group Newspapers system, all text is normally keyed directly into the computer via input keyboards. These replace the functions of both the linotype and ludlow operators. Photo: Daily Mirror

Publishing and the Damned

John Kemp

Imagine it. A newspaper office without paper.

Journalists abandon typewriters to write at VDU terminals; their electronic words are switched to editors' and sub-editors' terminals to be knocked into shape; pages are designed and made up on other VDUs. A parallel system looks after type for advertisements. Finally a computer-controlled laser beam "cuts" a plate ready to be bolted onto the presses.

Or why bother with plates? The computer could, instead, control electrostatic or ink jet presses, where fine jets of ink are squirted on just the right areas of paper as the newsprint rushes past.

Or why bother with presses, delivery vans, newsboys? It would be simplicity itself to send the computer's coded message by telephone or radio to a small ink jet or Xerox-type printer in the customer's home.

It is a journalist's dream and nightmare all rolled into one. But it is no crystal ball dream. Microprocessor technology makes almost all this possible today. Parts of it are already in use; the rest is not far behind.

All of which makes it rather surprising that Chris Ward (Personal Experience, PCW January 1979) says "editorial halls ... are without doubt among the most technologically conservative of places."

Like Chris Ward, I am a journalist fascinated by micros. I spend hours (too many, my wife says) poring over PCW and other computer mags, including the American ones. I yearn for the day when the pennies stretch far enough to have one of my own. I am even beginning to understand what makes them tick (if a few million signals a second can be described as a tick).

The thought of them taking over the office both enthralls and appals me. And take over (yes, I believe those are the right words) they will, certainly within my working life.

But it is not technological conservatism that is holding up the process. It is a number of very real fears, mainly over the quality of the papers we produce and over job security. While journalists' jobs are not threatened directly — indeed more jobs could be created by the electronic revolution — thousands of print workers will no longer be needed.

The composing room, where copy is set into type and pages assembled, will vanish because computer-aided journalists are doing the job; electronic scanners will handle pictures and the platemaking, press room, publishing and distribution departments will be replaced by telephone lines and customers' "receiver presses".

The threat to composing room jobs was one of the factors which led to suspension of publication of The Times late last year. The Nottingham Evening Post's switchover to an electronic newsroom was accompanied by wide-ranging redundancies. Mirror Group Newspapers, which is using VDUs for its weekly paper Reveille, is using print workers to operate the equipment.

The Threat to Jobs

Already, computer-controlled typesetting has led to a loss of jobs in composing rooms throughout the country. Mainly this has been achieved through voluntary redundancy and natural wastage. But there are still tragedies to be seen in any composing room . . . highly-skilled craftsmen too old for retraining on new equipment, passing the long days to retirement operating a photocopier (proofing press), drying off wet bromide paper on which the type has been set, pushing it through a waxer so the compositor making up the page can stick it to his paste-up. Skilled men, with a six-year apprenticeship and half a century of experience behind them doing tasks so menial they would be beneath many a school-leaver.

At least, on a fully electronic newspaper, these jobs would vanish. So would most others between the journalist and the reader.

Which is why there are fears, too, about the quality of newspapers.

Journalists are human; they make mistakes. Cynics would say it is all they make but that's another story. But with traditional newspapers and photo-set papers run on traditional lines, there are checks on the journalist's work which could vanish.

At present, a reporter's copy will be read (at least) by a news desk man and two, probably three, sub-editors. Hopefully, the copy should, by then, be perfect in accuracy, grammar, spelling, office style and with any legal snags ironed out.

On an electronic newspaper, all these stages would, presumably, continue in some form.

But the next series of checks would not. For the compositor who turns the words into type, the proof reader, the paste-up man (and the best do read the type they are sticking up) and an editorial man who oversees the make-up of the pages, all would vanish. Yet at each of those four stages, queries are raised, spellings corrected, facts challenged. All of which leads to a better product for the reader. And all of which are threatened by electronics.

Of course it would be naive to pretend that errors were not sometimes introduced, too. It would be equally naive to pretend that there was never overmanning, pointless restrictive practices and other evils in the newspaper industry. Or any other industry.

But it must always be remembered that newspapers have, if they are to give the public the best possible service, to be geared up to handle *late* news breaks — their lifeblood — quickly and efficiently. And it is this need to cope with quite unpredictable production peaks that leads to at least part of the apparent overmanning.

Certainly an electronic newspaper, with printing in the customer's home, would be able to operate with much later deadlines. We might even beat radio or TV to a story! And in Nottingham, deadlines are quite impressive.

Is It Justified?

But generally, computer-controlled typesetting has led to *earlier deadlines and a loss of flexibility*. For while computer typesetting is far less labour intensive, far more efficient in producing the newspaper as a whole, as long as copy reaches the composing room in a steady even flow, it is much slower than traditional methods when it comes to throwing Page One in the air at the last minute because of some late news break.

There are also problems in getting editorial changes made. Often a story will develop after the type has been set. And while proof readers' corrections have generally been done from a print-out while the text is still in the computer, the one or two lines a journalist changed on a proof seem to take an age to get through the system.

Another problem — this time with the software of the computer — is over what is known as hyphenation and justification. At its simplest, this involves varying the space between the words (and sometimes between letters, too) to make the right hand margin straight, as in this magazine, and not "ragged" as on a typewriter. Sometimes computers will put a ludicrous amount of space between letters of a word just to fill out the line. The result wastes space and jars on the reader.

What it should do, and often does badly, is break the next word between two lines, with a hyphen at the end of the first line. There are rules on how to do this which can be incorporated in the software. But the English language is notorious for exceptions to the rules. Which is why you can now see such beauties in newspapers as new-spapers, che-que or unu-sual. Irritating. But there are hidden dangers too. Think about hyphenating the South Yorkshire town of Penistone.

It is these problems, not technological conservatism, that make journalists wary of the so-called new technology.

Electronic newsrooms could revolutionise the economics of newspapers. Labour costs could be cut, newsprint bills — and they are really frightening — eliminated, deadlines improved by several hours. Indeed, many towns and cities with just one paper now could support several with the drastically reduced cost of production and improved news service.

There are an awful lot of problems to be ironed out before changes are made. But made they will be. With the right safeguards, it could be an exciting time.

Except for the printers in the dole queues.

Small Computers for Small Organisations

Mervyn J. Axson

Computing for the Small Organisation

Small and relatively inexpensive computers are now a reality and a number of different makes have appeared on the market, Apple, Heathkit, PET and Tandy to name a few. Most are advertised as being suitable for hobby, personal and small business use. It was the last that caught my attention. Could a system of this type really be used to perform some of the many routine tasks inherent in all organisations and particularly troublesome in the very small where there are few staff? You must have read of the problems that VAT has caused to small businesses for example, and there are many other such tasks to be performed.

These tasks all basically consist of *storing*, *sorting*, *retrieving* and *presenting* information of various kinds. Whilst to a lot of people computers are devices that do complicated sums at very high speed, and some in fact do this, most of the world's computers are engaged in the more mundane task of data processing, so it would seem that small computers might be the answer to the small organisation's prayer.

A few enquiries were made, but the replies were not promising. Data handling needs memory and memory is expensive, so small systems do not have much of it. More can be bought but at a price which may equal the initial cost of the basic system. Even then, external memory will be needed and this will have to be of random access type e.g. disc drives etc. which are very expensive, and so the cost has rapidly escalated from under £1,000 to £4,000 or even more, which is way beyond the budget. Even worse, to have a tailor made system would require special, expensive, software, and then maintenance of the system would need considerable skill and expert knowledge.

Computer data processing is beyond the reach of the very small organisation, and the small computers are condemned to play games and operate very general standardised programs unless in the hands of an expert.

But the thought intruded that the people who had given the answers undoubtedly knew a lot about computers, but probably very little about small organisations. Were they perhaps adopting totally different standards to those required in this instance? To one who knows quite a lot about small organisations and a little about computers, it seemed that they might be.

In the *large* system time is vital. Computer time is costly and there are many conflicting demands on the system, so programmes must operate very quickly. It is worth spending a lot of man hours on programming in order to save computer minutes (or seconds). In the *small* system the converse is true. Computer time is cheap and man time is expensive in the sense that there are a limited number of hours available in which to perform the tasks. Suppose that without a computer a routine task takes one hour for the man to perform. If the provision of a simple computer enables him to set it working in five minutes, then he has saved

55 minutes in which he can carry on with some of his many other tasks. It does not even matter if the computer is no quicker than him, since the important fact is that the task has been done *for* and not by him.

Again in the large system, many people are involved at both the design and operating stages. Much effort has to be expended to meet this situation. For instance, documentation of programs for the benefit of other programmers, extensive data checks and verification since many people are concerned in providing and listing data, providing instructions for operators who actually run the programs, etc., etc. In the very small system one person will carry out these tasks. It does seem that different standards can be adopted.

Let me lay down some principles.

a) The prime judgement of any program will be "Does it work within an acceptable time?" and the second "Is it reasonably simple?"
b) I will not try to produce a whole system at once but will start with a very small part and make it work as simply as possible. This can be expanded slightly to include some further part, and so the system will be built-up gradually. No doubt, in the process, new ideas will emerge, but I can always go back and alter some earlier procedure, and the outcome, hopefully, will be a system tailored exactly to my needs! Moreover, since I have been concerned at every stage, I shall understand it thoroughly and so be my own expert when problems arise, as they surely will.

A Typical Situation

This seems alright so far, but will it work in practice? Let us do a case study of a typical situation. Consider the hon. secretary of a fairly small golf club. There are 430 members in two main categories, men and ladies.

These are further sub-divided into playing, social and junior members.

Men of 65+ and ladies of 60+ pay half the normal subscription.

The club has a number of lockers in which members may store their clubs, etc. and these may be rented at an annual fee.

This will suffice for our example, since it will give fourteen different categories of membership, viz:—

1. Male playing with locker
2. Male playing with locker pensioner
3. Male playing no locker
4. Male playing no locker pensioner
5. Male social
6. Male social pensioner
7. Lady playing with locker
8. Lady playing with locker pensioner
9. Lady playing no locker
10. Lady playing no locker pensioner
11. Lady social
12. Lady social pensioner
13. Junior with locker
14. Junior no locker

All of this data will be stored in a *membership register*, together with the member's address, telephone no. etc. This register may be in the form of loose leaf pages, cards, a bound book divided into sections, etc. However it is stored, an individual member's record will be something like this:—

Name	Mr. A. N. Other,
Address	"Dormy Cottage" 27 Acacia Avenue, Anytown, Some County.
Phone	Anytown 1234
Category	Male playing, locker, pensioner

Once the register has been set up, the hon. sec's task would not appear to be very difficult. He only has to keep it up to date by recording such matters as change of address, adding new members and deleting those who resign, etc.

But let us think a little further. During the course of any year there will be many calls for the data in the register. Sometimes all of it will be required, at other times only part.

Some requests (or commands!) will be:—

a) The steward is to be provided with a list of all the members so that he may check on those people using the club. Addresses are not needed, but 'phone nos. would be useful. Junior members should be noted so that they may be kept out of the bar.
b) Address labels are required for all playing members so that a letter may be sent to them.
c) The ladies section want a list of all lady members.
d) The annual horror! All members to be advised that subscriptions are now due and reminded of the total sum that they should remit. This will be one of *fourteen different amounts* according to their category!

It is easy to see how the hon. sec. manipulates the data. One example, that of a case a), the steward's list, will show this. He examines the data record for the first member and writes/types the first data item "NAME". He ignores the "ADDRESS" data, and then writes/types "PHONE NO". He now checks on the membership category and if it is "JUNIOR" he writes/types JUNIOR. He then repeats this another 429 times and he has produced the list. Very simple, but awfully time consuming! And just the sort of operation that a small computer can easily handle.

We provide the computer with a list of the four data items for each member, and tell it what to do.

"Look at data for first member. Write his/her name. Write his/her 'phone no. If category "JUNIOR" write JUNIOR. Proceed to second member. And so on until all members have been dealt with"

Expanding

This is simple and should work. Let us expand a little. We can obtain the answers to the four problems posed earlier by the manipulation of just four data items, but is this sufficient for all our requirements? The more the data is sub-divided, the more we can manipulate it. A simple example is the member's name. We have taken it as a single data item, e.g. "NAME" = "MR A N OTHER". We can only instruct the computer to output it in this form. Let us suppose that it is traditional in the club to personalise letters, and that one starting "DEAR MR A N OTHER" will bring immediate protests from pedantic members who will insist that the form should be "DEAR MR OTHER". We can meet this requirement by dividing "NAME" into three data items "STYLE", "INITIALS" and "SURNAME". Now in this case the computer can be instructed to write DEAR followed by the first and third data items, which will produce the desired result.

Just how far this sub-division of data should be carried is a matter for the individual hon. sec. to decide, but he should be aware of the danger inherent in going too far, for he may end up with a system that is too cumbersome to be of much use. Compromise must be the keynote. If the hon. sec. thinks about how he uses the data he will soon be able to sort out the key items. He may not know much about computers, but he is an expert on the manual system!

There is no need for us to go through this process in detail. Let us assume that the decision has been made to store the data for each member in ten data items, which is a reasonable number to handle.

Data item	
A	STYLE
B	INITIALS OR FORENAME
C	SURNAME
D	
E	4 LINES FOR
F	ADDRESS
G	
H	PHONE NO.
I	CATEGORY
J	ANY OTHER MEMBER RELATED?

The first nine items are straightforward, but why has the hon. sec. seen fit to add a tenth, labelled "relationship"? This arises from experience. As you have probably guessed, I was once the hon. sec. of just such a golf club, and in pre-microcomputer days, I set up a small addressing machine system. Each member's name and address was recorded on a plate and these were kept in alphabetical sequence of men, followed by ladies, then juniors and finally social members. This was fine up to a point, but if a general circular was to be sent out it meant that quite often two or more letters were sent to one address where husband and wife were both members, with possible child or children as junior members. This immediately produces complaints that the hon. sec. is wasting club funds on postage, etc. Hence data item J in the system.

We now have the membership data arranged in a form so that the computer will produce many different kinds of output when given suitable instructions. Obviously, all the tasks previously mentioned can be performed, but so can many others that haven't been considered. One example will make the point. We have seen how to satisfy the pedants by writing "DEAR MR OTHER", but suppose we have a member who is titled; e.g. SIR JOHN SMITH. On the instructions we have the computer will write "DEAR SIR SMITH" and the pedants will have a field day! But thanks to the way the data has been arranged, they can be forestalled by a further simple instruction — "If STYLE is SIR write DEAR followed by first data item and second data item" — the result will be "DEAR SIR JOHN". The system seems flexible.

From Paper to Program

At this stage I was, and I hope you are, convinced that it has possibilities. But this is just about as far as we can go on paper. We must try to put these ideas into practice on a microcomputer. Since I had also reached the same stage with a number of other projects connected with my business, the decision was taken to buy a Commodore PET.

How can we use Pet to implement "Membership"? The data can be stored on a cassette tape using the built-in recorder. Programs can then be written into PET's internal memory by typing them in through the keyboard and then used to arrange and output the data stored on the cassette. These operating programs can themselves be stored on other cassettes, so once they have been tested, they can be simply loaded in a few seconds as required.

In the same way that I am using a special language, English, to communicate to you, we use a special language to communicate to computers. Just as there are many human languages, so there are many computer languages. Like most small systems, PET's language is BASIC. This is fairly easy to understand, and it will only take a little study to gain mastery of it at the level which our simple system requires. One warning is required. Just as human languages have 'dialects', so do computer languages. For instance "I am" is standard English but in some dialects this becomes "I be" (or even "oi be") whilst in others it is "I are". Since I am using a PET my dialect of Basic will be "Pet Basic" which although very similar to other Basics will have some minor differences. Bear this in mind if you want to use the programs on some other computer.

In the next article we will write the actual programs required to operate "Membership" on PET.

comart

We've come a long way since the quill pen ....
microcomputers are now taking the drudgery out of office life ....
providing safe and constant control of industrial processes ....
helping further our knowledge by monitoring scientific experiments and .... well you may have yet another application.

Comart specialise in microcomputers — from single-card computers to multiuser systems. Analogue interfaces, floppy discs, digital, VDU and printer interfaces are all in our range. Software from Assembler, Basic, Fortran, Cobol to Word Processing and Data Base Management.

Why not discover the Comart Catalogue of Computers — NOW

Contact us at :-

**COMART LTD., P.O. Box 2, St. Neots,
Cambridgeshire. PE19 4NY.
Tel: Huntingdon (0480) 215005.
Telex: 32514.**

Comart Computers are also available from regional dealers :-

Microcomputermart Ltd., Manchester. Tel: 061 832 2269 The Byte Shop, Ilford, Essex. Tel: 01 554 2177
Cambridge Computer Store, Cambridge. Tel: 0223 68155 Xitan Systems Ltd., Southampton. Tel: 0703 38740
New Bear Computing Store, Newbury, Berks. Tel: 0635 49223 Holdene Ltd., Leeds. Tel: 0532 459459
Isherwoods (Personal Computer Systems) Ltd., Luton, Beds. Tel: 0582 424851

ON YOUR MARK, GET SET.

~MK14. Diagnostic Program

Clifford Clark

The Diagnostic

Programs written to examine or check the action of an electronic data processing system are generally called "diagnostic" programs. Diagnosis is specifically the art or act of recognising disease from its symptoms. In computer jargon this has come to mean looking for "bugs" in a "sick" system or checking out a "healthy" system. The diagnostic program is loaded and run. From the results obtained, indications are given of the section in the equipment of a fault or malfunction.

The simple program shown in Figure 1 checks the key action and address locations. It is of use in two ways:—

1. It checks the key action and associated circuitry. (Diagnostic program)
2. It enables one to understand the programming of key actions for the MK14 system. (Routine appreciation program)

Program Action — How it Works

The first six bytes (OF12—OF17) load the display address "ODOO" into the pointer register No. 1. The next two bytes (OF18+OF19) give the option of loading a single address to each of the display and key locations which are in the range "ODOO—ODO7" (eight separate memory address locations). Byte "OF19" is the location in the program which is changed as required from "00" to "07". This number is put into the extension register by the instruction at "OF1A" ready for use as the displacement to be added to the address in pointer register No. 1. Bytes "OF1B" and "OF1C" load the contents of pointer register No. 1. plus the displacement from the extension register into the accumulator (ODOO plus 00 equals ODOO). "OF1D" and "OF1E" the instruction at these locations connects the bits of the byte in the accumulator to the first display LED unit.

```
MK14. PROGRAM TO CHECK KEY
ACTION AND ADDRESS LOCATIONS

0F12  C4 }
0F13  0D }
0F14  35 } LOADS DISPLAY ADD-
         }   RESS "0D00"
0F15  C4 } INTO POINTER REGISTER
             #1.
0F16  00 }
0F17  31 }
0F18  C4 } LOADS DATA "00"
0F19  00 } INTO ACCUMULATOR
0F1A  01   TRANSFERS "00" TO
             EXTENSION REGISTER
0F1B  C1 } LOADS CONTENTS OF
         }   PTR. #1 PLUS
0F1C  80 } CONTENTS OF EXTEN-
             SION REGISTER. INTO AC
0F1D  C9 } STORE CONTENTS OF AC.
             INTO ADDRESS
0F1E  80 } CONTAINED IN PTR. #1
             (DISPLAY)
0F1F  90 } JUMP TO 0F11 & START
             PROGRAM
0F20  F1 } AGAIN AT 0F12 TO LOOP.
```

Figure 1

If the bits are set (1) the appropriate segment is illuminated. If the bits are not set (0) the segment remains unlit. The relationship between the byte and the seven segments plus the decimal point was illustrated in the previous MK14 article (PCW Vol 1, No. 11). At this point in the program with the display instruction the computer is made to "look inside itself". In other words, it is being instructed to show what bits are set in its own display memory location. These locations are in the "read only memory" (ROM). The last two bytes of the program (OF1F—OF20) are the essential loop instructions to effect the display continuously.

Running the program

On running the program as shown in Figure 1 with location "OF19" containing "00" the first display unit of the MK14 (first on the right hand) will show all seven segments and the decimal point illuminated. This indicates that all eight bits of the byte are set (1). On pressing key "0" the decimal point will be switched off. This means that key "0" in the down or closed position switches the most significant bit (MSB) of memory location "ODOO" from set (1) to not set or off (0). Looking at this action as a complete byte of memory in hexadecimal notation it means that in this case key up or off gives "FF" and key down or on gives "7F". The table at Figure 2 gives the relationship between address, key action and effect on byte in ROM. This information is essential to be able to control programs by the various keys.

MEMORY LOCATION ROM	KEYS			
0D00	0	8	A	
0D01	1	9	B	
0D02	2		C	GO
0D03	3		D	MEM
0D04	4			ABORT
0D05	5			
0D06	6		E	
0D07	7		F	TERM
KEY DOWN	7F	BF	EF	DF
LED DISPLAY	8	0.	9.	2.

* NO KEY
ALL LOC = FF
DISPLAYS
SHOW 8.

Figure 2

This program demonstrates that the display units and the control keys share common addresses in ROM but each key or display unit has its own single address.

In writing programs that require key actions it is necessary to include in the program instruction sequence a "sensing" of the state of the particular memory location related to the key or keys that are required for control. There are various ways of doing this and they will be explained in future articles on the MK14 microcomputer.

Chess Programming

Before You Begin

David Levy

Until comparatively recently one essential prerequisite for any budding chess programmer was the odd £million or so needed to pick up a suitable mainframe. Nowadays it is possible for any home computer enthusiast to write his/her own program for a microprocessor based system costing less than £1,000.

Indeed, so many personal computer users are becoming interested in this field that I thought it worthwhile to pen a few pages of advice and warning so that PCW readers can be steered along the right track. The potential of micro-programs can be judged from the fact that at the 9th ACM chess championships, held in Washington last December, two such programs performed very creditably alongside many of the world's strongest mainframe programs: SARGON (written by Kathy and Dan Spracklen) scored 2½ out of 4 and MIKE (Mike Johnson) scored 1½. (Also see my article in November 78 PCW).

The Game of Chess

Chess is an extremely complex game to play yet its rules are clearly defined, and the way that the pieces move can be learned within a few minutes. Despite this, many programmers do not take the trouble to program all the rules properly. *En passant* captures and some of the rules pertaining to castling are often overlooked, even in commercially available chess machines costing up to £200. There is no excuse for such sloppy programming and I would therefore suggest that before you write so much as one instruction of code, do make sure that you know all the moves of chess and the circumstances under which the special moves can be made. This information is readily available in some elementary textbooks on the game but it would do no harm to consult the official rules of chess, published by Pitman in Britain and by David McKay in the USA.

Once you are confident that you know the moves and their rules, it is time to learn something about the game itself. If you are an inexperienced player you could do with some help, so try to get some hints on the basic principles of the game from a player who knows his onions. You can learn a certain amount from books but a few evenings at the local chess club will pay greater dividends and besides, the club can always do with your membership fee. Remember, if you do not possess some understanding of the problem (i.e. what chess is about) you will never be able to solve it (i.e. write a strong program).

Chess Challenger

Computer Chess Literature

There exists a wealth of literature on the subject, mostly papers in learned journals and in books on artificial intelligence. Before you rush off and invent the wheel again it will be worthwhile to study some of the more important items in the literature. As an optional introductory paper, and certainly one of great historic interest, I suggest Claude Shannon's "Programming a Computer for Playing Chess", Philosophical Magazine volume 41 (7th series), pp. 256-275.

Certainly more readily available are three useful books on the subject. "Chess and Computers" by David Levy, published in Britain by Batsford and in the USA by Computer Science Press, is an elementary introduction to the subject. "Chess Skill in Man and Machine", edited by Peter Frey (Springer Verlag) is an outstanding work at a more advanced level. "Computer Chess" by Monroe Newborn (Academic Press) lies somewhere between the two. The first two titles contain extensive bibliographies.

Where to Begin Programming?

This is not an appropriate place to describe in detail how chess programs work. Suffice it to say that the three key items in all chess programs are:
(i) A move generator which lists the legal moves from any position and which can generate a tree structure representing the possible variations in play.
(ii) A position evaluator which assesses the merit of positions in the tree.
(iii) A tree searching mechanism which determines optimal play for both sides and thereby selects the move which the program makes.

Since the tree of possibilities in chess is enormous it is necessary to use various "tricks" to speed up the search process. By far the most powerful trick is a device known as the alpha-beta algorithm which can plough through a tree with N terminal nodes (or positions), and select a move after evaluating something of the order of $6 \times \sqrt{N}$ positions (optimally $2 \times \sqrt{N}$ but optimality in chess is a pipe-dream). Since the tree is enormous, so is the saving achieved by employing the alpha-beta algorithm.

This algorithm is described in many parts of the literature but possibly the best description, which comes complete with an Algol-like version of the algorithm, is that found in an article by Knuth and Moore, "An analysis of alpha-beta pruning", in "Artificial Intelligence", volume 6 (1975), pp. 293-326. Do not worry about overflowing memory during the tree search. The alpha-beta algorithm needs only to retain one node at each level of the tree.

Of crucial importance in tree searching is your choice of the quiescence criteria. Prune off too few moves and the combinatorial effects on the growth of the tree will cause your program to think forever over its moves. Prune off too many and it will overlook some important tactical possibilities. Most programs perform an exhaustive search to some fixed depth then a selective search along tactical paths, but why not try to grow intelligent trees, just as strong human players do?

Your position evaluator may be as primitive or sophisticated as desired, but remember that there is a trade off between sophisticated evaluation (which takes a lot of time) and the reduction in the size of the tree that can be examined (because of the time taken to evaluate each node). It has yet to be determined whether sophisticated evaluation and a small tree search is better than a primitive evaluation function and a larger search. You should experiment before making your final decision — begin with an evaluation function containing only one or two terms (material and mobility are the most important) and build it up from there until you are satisfied with the results. Remember, your position evaluator is the most frequently used part of the program so it must execute quickly.

One useful device in the evaluation mechanism is what Donald Michie calls "swap-off". This is described incorrectly on pages 45-47 of "Chess and Computers", the errors having been copied over from Michie's original article on the subject which was published in 1966. I am indebted to Helmut Richter of Hamburg for the following, more accurate, description.

Swapoff Values

The swapoff value of a square is the material gain that can be expected if the side on the move makes the most of a capturing or exchanging sequence on that square. The purpose of using swapoff values is to decide whether or not a particular capture is worthwhile without the necessity for lookahead. It can also determine the safety of a square.

The basis for calculating swapoff values is that the side on the move should either make no capture on a particular square or should capture with his least valuable unit so that any recapture by his opponent will have minimal value.

Swapoff values are calculated in the following way. Assume that a black piece of value v_0 is defended by n black pieces of values $v_1, v_2, \ldots v_n$ in ascending order of value, and attacked by N white pieces of values $u_1, u_2, \ldots u_N$ in ascending order of value.

(w = white; b = black)

$w_1 = v_0$

$w_2 = v_0 - u_1 + v_1$

$w_3 = v_0 - u_1 + v_1 - u_2 + v_2$

$w_4 = v_0 - u_1 + v_1 - u_2 + v_2 - u_3 + v_3$ etc.

and

$b_1 = v_0 - u_1$

$b_2 = v_0 - u_1 + v_1 - u_2$

$b_3 = v_0 - u_1 + v_1 - u_2 + v_2 - u_3$

$b_4 = v_0 - u_1 + v_1 - u_2 + v_2 - u_3 + v_3 - u_4$ etc.

These two series are calculated until one side or the other runs out of pieces with which to capture on the square occupied by v_0.

Let us assume that:

$w_1 = 2$ $b_1 = 0$

$w_2 = -4$ $b_2 = 10$

$w_3 = -12$ $b_3 = 10$

$w_4 = -10$ ($b_4 = 13$) . . . which is ignored, since the last player to capture does not lose the capturing piece.

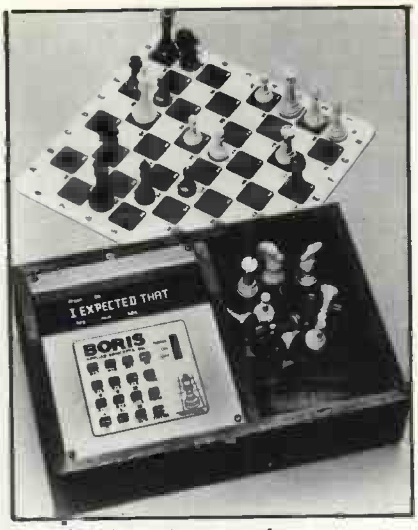
These are placed in a one-dimensional array: 2, 0, -4, 10, -12, 10, -10.

(Values of white and black alternately)

There are two reasons for stopping the sequence. One side may not capture when it is possible to do so; or both sides capture until no more captures are possible. White may stop after the even indexed elements of the array (after Black has captured) and Black can stop after the odd-indexed elements. In order not to treat the last value as a special case a zero is added to the array and it is repeated, so both sides have the same last value: 2, 0, -4, 10, -12, 10, -10, 0.0 . . . (1).

White is trying to reach the maximum of the even-indexed elements and Black the minimum of the odd ones in array (1).

The Videomaster Chess Champion



BORIS – Chess Computer from Optimisation

Max (0, 10, 10, 0) = 10	index = 6
Min (2, -4, -12, -10, 0) = -12	index = 5

(Index is the position of max or min element in array (1)).

Black cannot continue after -12, Therefore the sequence after -12 must be pruned off, leaving the new array:

2, 0, -4, 10, -12, -12
(this last value is a copy)

White now tries to find a new maximum and Black tries to find a new minimum. The process is repeated until White's maximum = Black's minimum. Here:

Max (0, 10, -12) = 10	index = 4
Min (2, -4, -12) = -12	index = 5

New array is 2, 0, -4, 10, 10 (copied)

Max (0, 10) = 10	index = 4
Min (2, -4, 10) = -4	index = 3

New array is 2, 0, -4, -4 (copied)

Max (0, -4) = 0	index = 2
Min (2, -4) = -4	index = 3

New array is 2, 0, 0 (copied)
Max (0) = 0
Min (2,0) = 0. Finally we have:

Maximum = Minimum = value of exchanging sequence = 0.

Therefore white neither profits nor loses by making a capture on that square.

Swapoff values can be used to determine whether or not it is safe to move a piece to a particular (vacant) square. Simply make the first element of the array 0 (and remember that the piece which moves to this square no longer attacks it).

The purpose of using swap-off is to look out (without looking ahead!) for possibilities of gaining material by successively capturing on a particular square. The method used does not take into account pins, but I feel that it is still an extremely useful tool. A chess program which detected all elementary exchanging and capturing situations would already be a stronger player than 50% of the world's chess playing population.

Move generation should be carefully thought out. When 64-bit processors are readily available, move generation and position evaluation will become easier to code and quicker to execute (thank God chess is not played on a 9 x 9 board). Speed in move generation is almost as crucial as speed in position evaluation so any time that you invest in optimising these routines will be usefully spent.

The Three Phases of Chess

Chess is conventionally divided into the opening, the middle-game and the endgame. In the opening both sides seek to develop their pieces on sensible squares. A vast amount has been written about the openings but programmers should be wary of storing dozens or hundreds of opening moves as they will be useless against an opponent who chooses to follow a different variation. It is far better to teach your program some basic principles of opening play, such as encouraging it to develop its pieces and to castle.

The middle-game is usually witness to most of the cut and thrust fighting that goes on on the chess board. A good tactical analyzer is very useful here, so try to grow your trees in an intelligent way so that most CPU time is spent in looking at the critical variations. Use obvious heuristics to cut down the growth of your tree; for example, do not spend much time looking at moves that allow the opponent an immediate gain of material.

The endgame has always been and will always be the hardest part of chess to program. It is the part of the game that best sorts out the masters from the lesser mortals. Read the final two pages of Reuben Fine's "Basic Chess Endings" (Bell and Hyman) and try to implement some of his rules. In particular, your program should know about advancing passed pawns — an extra queen never did anybody any harm.

Since you are obviously interested in computer chess you should join the International Computer Chess Association. This organization publishes a news bulletin a few times each year and it will help you to keep up with latest trends. To join ICCA send $5 (US) to: *Professor B. Mittman*, Vogelback Computer Center, 2129 Sheridan Road, Evanston, Il, USA 60201.

Good luck in your programming, and if you write the first program that wins a match against me you will collect the $5,000 Levy/OMNI prize.

P.S. I regret that pressure of work prevents me from dealing with correspondence on this subject.

PCW *David Levy is setting up a company to develop intelligent software for sophisticated microprocessor applications. Any readers interested in writing assembler programs on a contract basis please write to David Levy, Box 123, Personal Computer World, 62a Westbourne Grove, London W2.*

Anyone actively interested in speech recognition, music composition or robotics is also invited to apply. PCW

PCC 2000 Business Computer

Low-Cost

Features

- 8085 CPU
- 64K Bytes RAM
- 4 Channel DMA
- Vectored interrupt handling
- 1.2 megabytes of disk storage
- Built-in CRT
- Detached, typewriter-like keyboard
- Powerful Disk BASIC operating system

● CP/M ● BASIC

● FORTRAN ● COBOL

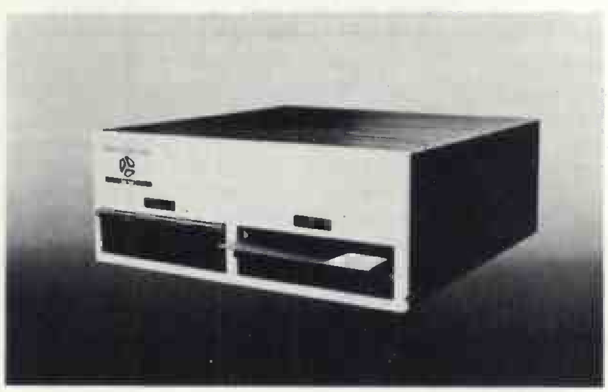
SERIES I

BUSINESS/SCIENTIFIC COMPUTER

Low-Cost

NEW

£2,400



HARDWARE

- CPU: Z80 microprocessor, 4 MHz
- RAM: 64 kilobytes
- ROM: 1 kilobyte
- DISC: 0.5M - 4M bytes
- DRIVES: Shugart, two 8 inch, single or double sided
- I/O: 2 RS232, 1 Parallel
- DMA: Z80 [optional]
- FPP: AMD 9511 floating point processor [optional]

SOFTWARE

- CP/M Disc Operating System
- CBASIC [extended BASIC]
- FORTRAN IV [compiler]
- PASCAL
- COBOL
- Macro Assembler [Z80]
- Business and word processing packages available

Sole agents

compelec altair agents

14-15 BERNERS STREET, LONDON W1P 3DE Telephone 01-636 1392 (4 lines)

The Buyer Bites Back

An occasional look at the law

Brian W. Haines

At a recent personal computing conference there were passing comments by several speakers about the hazards of purchasing for the amateur. Overdue deliveries, faulty components and poor service manuals were amongst the most prevalent troubles. One man complained that he had been supplied with a circuit diagram for a self-build kit that was so poor as to be unreadable, and in the event turned out to be wrong anyway! Other people had similar trouble. He had gone back to the supplier only to be told that the board was really only in the development stage; not a very happy explanation for him as his expertise did not extend to finishing the development, nor did he want to; he was looking to save himself a few pennies. With the shop some hundred miles away he wasted a lot of time and money getting the thing put right — he also had the worry and inconvenience of a board that did not work.

The problem is that the few shops dealing in computers are under heavy pressure to get items to the public often before they have been properly tested; the demand is for something publicised in an overseas journal; and everyone is caught up in an industry that is developing at a tremendous speed. Materials are liable to become obsolete even before they have left the shelves with buyers crying out for the latest designs, yet willing to take what is available.

In such an atmosphere the door is wide open to the cowboys, the 'in and out' traders who will advise on things they have not got, or 'equivalents' or near copies. They were in Central heating, double-glazing, Hi-fi and they will be in Computers. The keen amateur, prepared to forgive the pioneer traders who are often keen amateurs themselves, is liable to loose a lot of money unless he keeps his legal rights in mind now the market is expanding.

Now what are those legal rights and do they protect?.

In the main, consumer protection is based upon the Sale of Goods Act 1893. There are three main rules, as it were, that apply to anything you buy from a shop. If the goods don't conform to the rules you have the legal right to compensation and damages. This means in effect you can get your money back if what you buy is no good, or if it does not comply with its' description or it is not fit for the purpose for which it is sold. And you can claim for any inconvenience or damage you may have suffered. Thus the man with the poor assembly kit could have returned it and asked for his money back and the costs he incurred in buying and sending it back.

As a rule not many people bother with the relatively trifling cost of inconvenience. However if the firm refuses to do anything the only alternative is a Court action when such sums should be included. The other alternative is to ask for a replacement which does work, or a rebate on the price; these are a matter of negotiation with the shop and not a legal remedy or right.

But what about the matter of overdue delivery? Well, all you have to do is to stipulate a reasonable date after which the goods are unacceptable, if that date passes with no goods in sight then you are entitled to your money back if you have paid any, or no liability for payment if you have not. It is worth remembering with all goods sent by mail order that they are yours the moment they are appropriated to you, or in general terms the moment they are packed and sent.

The rights you have as a consumer may not be taken away by any notice of disclaimer but if you get goods at a trader's discount you could lose that protection as you have *become a trader*. It is often as well with expensive equipment to resist the temptation to 'get a bit off' and deal only with established outlets who will give after sales service.

Exercising legal rights takes time if recourse has to be made to the Courts. The best method is prevention, with the firm knowledge that the Courts will back you if you have taken the trouble to look after your own interests. The law is based mainly on common-sense behaviour between the buyer and seller, things often go wrong because either one party or the other hasn't taken a reasonable precaution. The advice given to avoid expensive mistakes seems so obvious many people overlook it:

If you are dealing by mail-order double check your order is correctly addressed and your own address is enclosed. Check as far as you can that the trader has an established business and is more than an address and telephone number. Of course people have to start sometime, if the firm is a new one telephone them and get a name of someone who can vouch for them.

Never part with any money against unspecified future dealing dates, ask if the dealer has the article in stock, there is no need to subsidise a firm while it buys abroad or anywhere else.

Be sure you know what you want before buying and if there is an acceptable substitute, do not be persuaded you need something else if the item is out of stock or is said to be obsolete.

If you are in a store, check off each item as it is given to you and examine all documentation to see it is properly printed and complete with pages running consecutively; a sealed plastic container means nothing! If you sign for anything without checking, it is difficult to contend later that you did not examine it first.

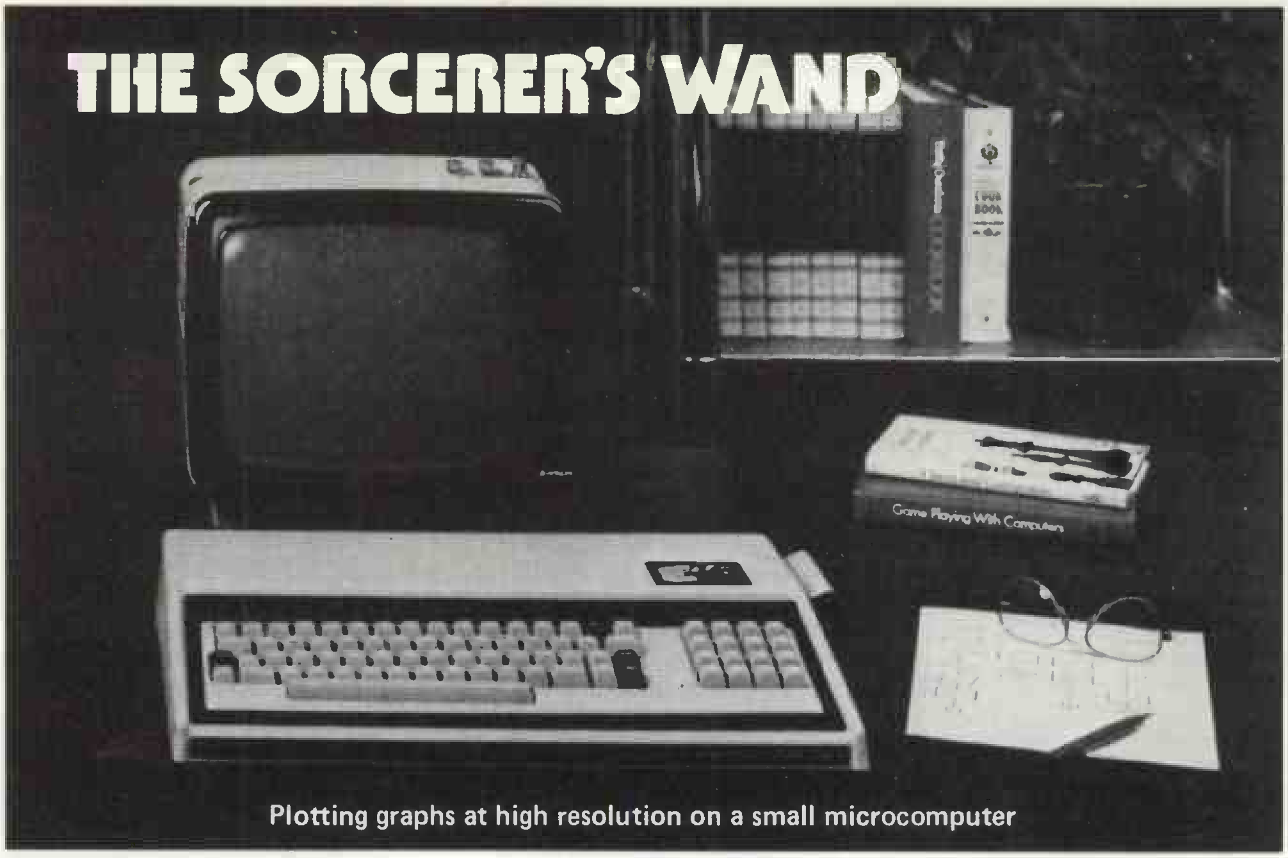
The final piece of advice has been given time and again, and although not legal is quite the most important. Never spend more than you can afford to lose, the law moves very slowly and is not proof against people who disappear.

When things go wrong the first step is to contact the trader, find out the name of the manager, tell him in person if possible what the problem is. Do this at once, never wait around getting advice from friends or complaining to everyone you meet. Most traders want satisfied customers, they know they are primarily liable. You can try the manufacturer if the trader proves obdurate and even overseas ones will sometimes be very co-operative.

A last resort is a writ in the County Court, the procedure is neither difficult or expensive, it just takes a long time. Far better to keep out of the hands of the lawyers and the Court by starting your last letter 'I hope we can resolve this without going to Court', then making it plain you know your legal rights under the Sale of Goods Act.

PCW *This is sound advice. The overwhelming majority of people in the business want to stay in business and are sometimes more sinned against than sinning. Patience and cooperation across the counter!* PCW

THE SORCERER'S WAND



Plotting graphs at high resolution on a small microcomputer

Dr. R. J. Beynon
Department of Biochemistry, The University of Liverpool

In addition to business and home applications, a major use for the microcomputer is in the scientific or educational environment. Many systems fail in such a context because of their limited ability to present data in a graphical form. A stream of numbers appearing on a VDU screen or on a line printer are restricted in the immediacy of information that they provide, requiring close scrutiny or plotting on paper. In all scientific journals data is presented in graphical form wherever possible, the information thus conveyed is considerably superior to a table of numbers. A trend in a sequence of data is often far more easily observed by studying a pictorial representation; this argument applies equally well to scientific or educational fields.

The graphics facility of a microcomputer may be defined by a) the resolution, and b) the degree of software or firmware support. Thus, the PET has a screen size of 25 lines of 40 characters, with no inbuilt facility for addressing points on a X = 0 to 40, Y = 0 to 25 'map' of the screen. The Research Machines 38OZ is superior to the PET — using 24 lines of 40 characters it provides software routines allowing X, Y addressing at a resolution of 48 x 80. This improvement at a resolution is achieved by the "quarter square" method — described in more detail later in this article. The Sorcerer provides 30 lines of 64 characters with no X, Y addressing support, and so on. The facilities on some machines are shown in Table 1.

It is a relatively simple matter to write BASIC routines that allow each character space on the screen to be addressed in an X, Y mode — by the judicious use of POKE commands. This provides a simple graph plotting facility that has a resolving power equal to the number of character spaces on the screen (e.g. for the PET [40 x 25] the resolution is 2.5% by 4%). Any attempt to introduce axes, or scale labelling, or to leave room on the screen for commands or data results in a further reduction in effective resolution. A graph drawn with points consisting of 2.5% of the height and 4% of the width of the graph produces an unsatisfactory 'stepped' curve, especially if large characters such as a whole white square are used.

There are some tricks that may be used to improve the smoothness of the plot in a subjective fashion. The simplest to implement is the replacement of a full white square by a smaller character such as a ".".

Although the effect is one of improved smoothness the resolution is unchanged. The second approach is to write routines that sense when a series of characters are printed adjacent to one another on the same line. In this case the program would eliminate all but the central character, as it is reasonable to assume that this character represents the curve best.

These routines will reduce the number of characters plotted on the

Table 1

Graph plotting facilities on some small systems

Microcomputer	Screen size (lines x char/line)	Resolution without software enhancement (X, Y)	Firmware support	User defined graphics
PET	25 x 40	40,25	NO	NO
APPLE	24 x 48	240,192	YES	YES
TRS 80	16 x 64	128,48	YES	NO
RM 38OZ	24 x 40	80,48	YES	NO
SORCERER	30 x 64	64,30	NO	YES
CHALLENGER 2P	32 x 64	64,32	NO	NO

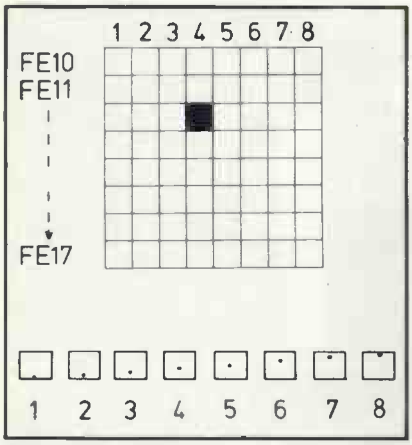


Fig. 1. *The characters matrix of the Sorcerer. Column 4 is used in this program. The memory locations for the graphic character corresponding to key '3' are indicated on the left. The decimal equivalents of a 2 character key code are POKE'ed into these locations to define character 6 in the lower part of the diagram.*

screen, so care must be taken with lines of shallow gradient — a line parallel to the x-axis would otherwise be represented by a single point! A further note of caution — the decimal point that is used in some commercially available routines is not located at the centre of the character space, which has the effect of introducing a permanent bias into the plotted line. It might be preferable to use a "+" or a "*" character, as these are more central in the dot matrix composing the character.

Despite the subjective improvement that such modifications provide, there remains a need for a genuine improvement in resolution from the limitations of a (number of lines) by (number of character/line) system. The high resolution potential of many microcomputers is never realised because of character-space plotting restrictions. The effective maximum resolution that represents the "impossible dream" on many systems is defined as (number of dots across the screen) by (number of dots down the screen) — for PET this would be (40 x 5) by (25 x 7). Such an effective resolution facility is only proved on the Apple system allowing points to be plotted on a matrix of 192 x 240.

The Apple is currently the only micro-based system that provides true high resolution graphics facilities — the diagrams of three dimensional plots in their literature attest to the superiority of their graphics offerings.

The "Quarter Square" method

However, there are other ways to 'tune up' your VDU display to provide higher resolution. The first of these methods is well known and uses the "quarter square" approach. Most microcomputers that offer a graphics facility allow the user to draw a "quarter square" type of character (four in all). By including appropriate routines to select which of these four quarter squares is best fitted to the current plotting requirements a 2-fold enhancement of resolution may be achieved in both the Y and X axes. Such a routine in the PET for example would increase the number of points to 80 x 50 providing a resolution of 1.25% by 2%. The doubling of resolving power is a step in the right direction, but is still far from the theoretical maximum. The application of a quarter square method to the Sorcerer VDU would give an effective resolution of 0.8% by 1.3%; better, but still not good.

An approach to high resolution graphs on the Sorcerer

The rest of this article describes methods that I am developing to allow full use to be made of the resolving power of the Sorcerer's screen.

The method is applicable to any microcomputer that allows the user to define his own graphic characters. With the Sorcerer it is possible to define, either through the monitor or in BASIC, 128 graphic characters. Each character is defined as a pattern of dots within an 8 x 8 matrix (the character space of the Sorcerer).

Briefly, the method is as follows:

1. Define 8 graphics characters as displaying one dot out of 64 possible. Each character differs from the 7 others in the position of the dot in a vertical line, the vertical line being in this case column 4 out of the 8 (Fig. 1).
2. The Y values of a function are calculated (or input) for a specified position along the X axis.
3. The value of Y — INT(Y) defines which of the 8 graphics characters is to be plotted at the position specified by INT(Y).
4. The selected graphic character is plotted using POKE statements.
5. The whole process from 2. is repeated for a new position along the X axis.

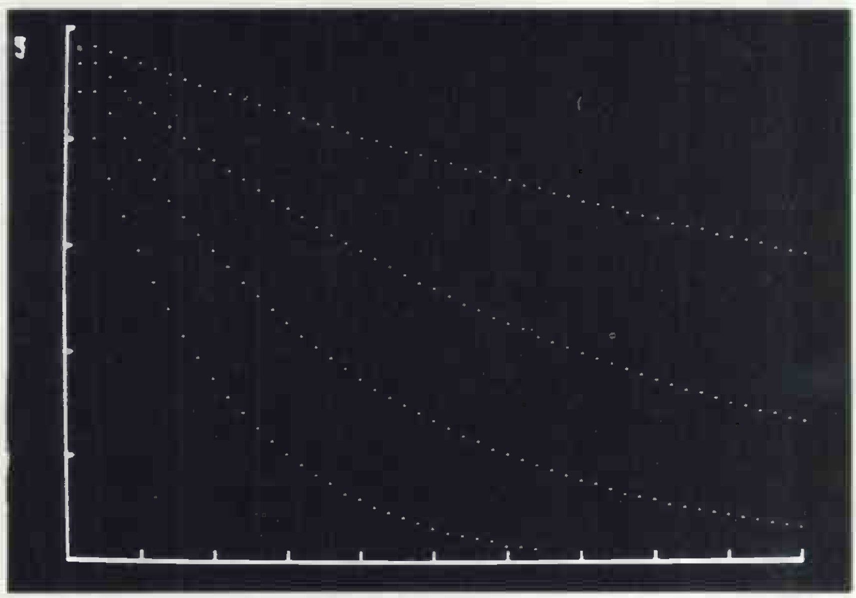


Fig. 2a. *The product of a curve plotting routine for the Sorcerer as listed in Fig. 3. Note that although the tail of the lower curve overwrote the X-axis a routine was incorporated to re-establish axes after plotting.*

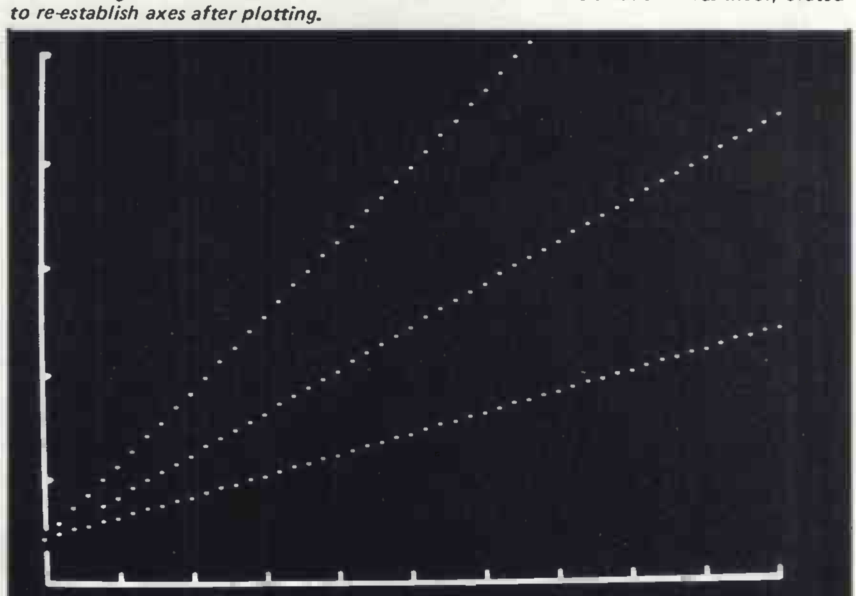


Fig. 2b. *The straight line capabilities of this approach produced by plotting a different equation (not listed) in line 70.*

To illustrate this method Figure 2 shows a plot of a series of concave curves representing the radio-isotope decay. The program that produced this series of curves is listed in Figure 3. A line by line breakdown is given below:

line 7 — Sorcerer's way of clearing the screen.

lines 10 - 50 — Draw the axes and indentations. Note that the axes are not drawn to occupy the whole screen. 10 columns to the left of the Y axis, are retained for scale labelling and prompts. Similarly, two lines are left below the X axis. Blank lines are deliberately left to the top of the graph to prevent accidental data loss due to scrolling up.

line 51 — After switching on, Sorcerer's user — definable graphics are not blank, but filled with random gibberish. This line clears out that rubbish.

line 52 — Defines the 8 graphics characters that are used by specifying which dot to light up in each.

line 57 — Nothing to do with the plotting routines, just inputs a constant for equation in line 70.

line 70 — The equation that defines the curves described here.

line 80 — Selects which graphic character to plot.

line 100 — Plots the selected graphic character at INT(X), INT(Y).

lines 130 - 145 — Restores X axis.

line 150 — Permits a new curve to be plotted.

This simple program has several notable features. Firstly it is important to note that the lines are not continuous — as only 1 in 8 dots is plotted. The software needed to achieve plotting of every line would be much more extensive and would slow the program down considerably. As it is presented here this routine takes about two seconds to draw a line across the screen. The tradeoff of speed for continuous lines is probably not worthwhile. Secondly, more than one curve may be plotted on the screen (Figure 2). However, if the second curve to be plotted uses the same character spaces as the first then the first dots will be overwritten. This is apparent in the top left hand corner of Figure 2a. Finally, the axes may be overwritten by dots on the curve, so lines are included to restore these at the end of the run.

Whilst it is appreciated that this approach is not applicable to every microcomputer system, the recent appearance of models such as the Sorcerer that allow user-defined graphics may indicate a trend in the provision of a new feature. One use of this feature has been illustrated above. It is realised that the routines are far from complete; axis labelling, tests for off-scale values, etc., have all been omitted. The program listed here occupies approximately 0.8K and may form a useful subroutine in more extensive programs such as interactive curve fitting or simulation programs. A LEM program that shows height as a *continuous* function of descent time? Each of these could themselves occupy a whole article.

```
LIST
1 REM THIS PROG DRAWS A HIGH RESOLUTION LINE(200*400)
2 REM PLOTS THE FIRST ORDER DECAY FROM 0 TO 50 FOR
3 REM A GIVEN RATE CONSTANT
5 REM RECORDED AT 0VU AND 300 BAUD
7 PRINT CHR$(12)
10 FORA=1TO50:POKE(-2166+A),151:NEXT
20 FORB=1TO25:POKE(-2166-(B*64)),162:NEXT
30 FORC=1TO50STEP5:POKE(-2167+C),178:NEXT
40 FORD=1TO25STEP5:POKE(-2166-(64*(D-1))),180:NEXT
50 POKE-2116,191:POKE-3766,188:POKE-2166,190
51 FORL1=1TO64:POKE-512+(L1-1),00:NEXT
52 FORL2=7TO56STEP7:POKE(-512+L2),16:NEXTL2
57 INPUT"K=";K
60 FORX=1TO51
70 Y=25*EXP(-K*(X-1))
80 Z=INT((Y-INT(Y))/.125)
105 CH=192+Z
110 POKE -2166-(INT(Y)*64)+INT(X-1),CH
120 NEXT X
130 FORE=1TO50:POKE(-2166+E),151:NEXT
140 FORF=1TO50STEP5:POKE(-2167+F),178:NEXT
145 POKE-2166,190:POKE-2116,191:POKE-3766,188
150 GOTO57
READY
CSAVE PLEX5
READY
```

Fig. 3. *The curve plotting routine for the Sorcerer. A line by line breakdown of the program is given in the text.*

SMART 1

How to make a combined buffer and 8K RAM board for the NASCOM without really trying

Mike Dennis

Introduction

NASCOM are now offering their expansion kits. You need to buy their buffer board before buying the memory or I/O expansion boards. This buffer board converts the 43 NASCOM expansion socket into the 78 was NASBUS. NASBUS uses an identical connector to the 77—68 kit that Newbear market, is readily available from either and consists of a 80 way socket on a 0.1" pitch — the two end pins not used. The actual mother board that carries extra expansion sockets for other boards is mounted vertically and the buffer board plugs in at one end. The mother board can be nothing more fancy than an 8" wide strip of Veroboard. Figure 1 should give you a rough idea of one way that it could all hang together.

I feel a separate buffer board effectively wastes potentially valuable square inches in the system. I also question the need for any EPROM sockets as you already have one spare on the NASCOM board; and I hear it on the grapevine that a cheap board that will take up to 8 EPROMs on one board is just around the corner. It's almost as cheap and simple to interface to, and decode, eight EPROM's as it is one and, in any case, do you really want all that dedicated software at this stage?

As a result, SMART 1 was born, is NASBUS compatible and consists of a combined 8" x 8" buffer board that provides all of the functions plus the added bonus of up to 32K of dynamic RAM (if your wallet can stand the strain!). At the time that this article is being written (Oct. 78), kits and a PCB are well under way

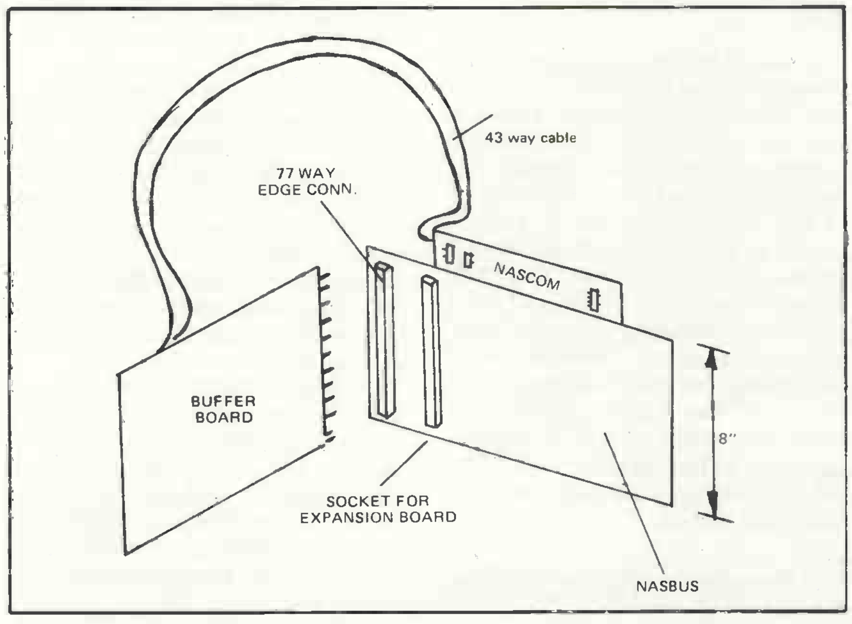


Fig. 1

and should be ready by the time that you read this article — more details of that later. First, we'll take a look at the basic concepts behind the final design.

Buffers

In its naked state, the NASCOM 1 board is completely unbuffered. In fact, there are so many ic's hung across some of the signal lines from the CPU etc., that there's not much drive capability left to play with. It is possible to interface to the NASCOM but some care needs to be taken in the designs as buffers are normally dead easy to implement in an original design. If you already have a design with on-board memory and/or I/O then you need to be able to disable these devices whenever you don't want to use them. The basic gating signal required can be a combination of $\overline{MREQ}$ and $\overline{IORQ}$ plus some address decoding so let us look at the memory expansion first.

Memory Expansion

The NASCOM provides Link 5 to facilitate this and when it is in the $\overline{MEMEXT}$ position, then a '0' on pin 40 on the NASCOM Expansion socket will enable the decoder (N36a)* to the on-board ROM and RAM. The memory map for the NASCOM is shown in Figure 2.

0000 – 03FF	NASBUG
0400 – 07FF	Space for extra EPROM
0800 – 0BFF	VDU RAM
0C00 – 0FFF	User RAM

Fig. 2

It would appear, then, that as long as we only take $\overline{MEMEXT}$ low whenever address 0xxx (where x represents an indeterminate hex number) occurs then this should satisfy our requirements and the simple circuit of Fig. 3 will achieve this. It is not as simple as that, however, due to the apparently little known fact that during refresh, the contents of the I register (used during interrupt handling) appears on the top eight lines of the address bus. If it happens to be Ox, then our simple decoder will think that N36a should be enabled and with certain combinations of 'x' will confuse the VDU and result in a scrambled display.

* to discriminate between SMART 1 and NASCOM, all NASCOM ic's are prefixed by 'N'.

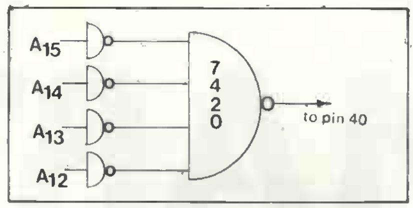


Fig. 3

The trick is to gate the signal to pin 40 with $\overline{RFSH}$. In SMART 1, this decoding is done by IC13, 3/10 (IC3 pin 10) and 14/4. Links are provided on the inputs to IC13 to enable the necessary rejigging of the address decoding if you decide to put in 16K dynamics at a later date.

I/O expansion

Although NASCO have provided a link on pin 39 for gating with an external signal ($\overline{IOEXT}$), if you want to expand the I/O you have to remove the PIO from the NASCOM board to prevent conflicts on the bus. This even applies whether or not you use NASCO's own expansion board. You will end up with one less PIO position than you thought you had. The reason for this is that there is a connection on the NASCOM board from N45a/1 to N46a/2 and is either a '0' for ports 0 to 3 or a '1' for ports 4 to 7 (the PIO). This is the same connection that will be used for $\overline{IOEXT}$. The basic idea should have been similar to $\overline{MEMEXT}$ in that a '0' enabled the on-board ic's and a '1' disabled them. Unfortunately, a '1' on pin 39 will therefore also enable the on-board PIO and so this line cannot be used exclusively to decode the on-board ports whilst still retaining the facility of the PIO. I did contemplate transferring the PIO onto SMART 1 but in the end decided against it due to lack of space.

What else do we need to consider? For a start, we need to buffer all the outgoing control signals such as $\overline{RD}$,$\overline{WR}$ etc., and the address lines A_0 —A_{15}. These are all unidirectional and could be buffered with ordinary low-powered Schottky TTL. However, we might want to disconnect the NASCOM signals from the bus at a later date and, as it is a relatively simple matter, we can put these signals through tri-state buffers, IC24, 33 and 34. The data buffers need to be bidirectional and so two tri-state buffers (IC 11 and 12) are connected back-to-back.

Normally, the NASCOM will talk to the world and so IC12 is enabled unless some other device wishes to talk to the NASCOM. The device must signal its wish to talk to the NASCOM by pulling $\overline{DBDR}$ low which disables IC12 and enables IC11. $\overline{DBDR}$ is made an open collector line and so any other boards plugged into NASBUS can use this line.

All five buffers are disabled when $\overline{BUSAK}$ goes low indicating that the NASCOM CPU has relinquished control of the bus. All the outputs from the buffers are fed onto the NASBUS and certain control lines are used by the on-board memory. As a result, where necessary, additional buffer gates are used to minimise the loading on the bus by the memory and IC5/12 and 10 in the $\overline{WR}$ line are an example of this. The actual buffer chips used in SMART 1 are 81LS95 or 97's and have worked perfectly satisfactorily with all the NASCOM boards that I have tried. In the unlikely event of there not being sufficient drive available on your particular NASCOM then you will have to insert some National 8836's between NASCOM and SMART 1. I have not made any provision for this, apart from some links in the lines to IC12, as you will need eight packages to do the lot, which will take up a lot of room. I found that I didn't need them. That just about completes the buffering.

Next we'll have a look at the memory available in SMART 1 and see how that's done.

Dynamics are cheap

As I mentioned in a previous article, dynamic memory doesn't knock a hole either in your pocket (at least, not for 4k RAM's) or your power supply. There are various devices around but the most logical type is the 4027 which is in the format of 4k x 1. This is virtually compatible with 4116's which are 16k x 1 and so, providing the original design caters for the slight address changes necessary, the upgrade can be made quite painlessly apart from the cost! How do dynamic memories of this type work?

To address 4k of memory will require 12 pins for the address and four more are needed for the power supply feeds (+12, +5, −5 and Ov). This doesn't leave us with many pins left to get data in and out, etc., if we use the 16 pin package that industry prefers. Fortunately, these memory chips have been designed into an array of 64 rows by 64 columns. If we first of all feed in a 6 bit row address and latch it inside the chip then we can use the same pins to feed in the 6 bit column address and leave the chip to combine the two addresses into the required 12 bit address. Selection for the row address or column address is performed by IC1 and 2 but we still need to generate two negative going clocks to latch in the two halves of the address. These two clocks are called $\overline{RAS}$ and $\overline{CAS}$ and need careful implementation.

If we send just $\overline{RAS}$ and the row

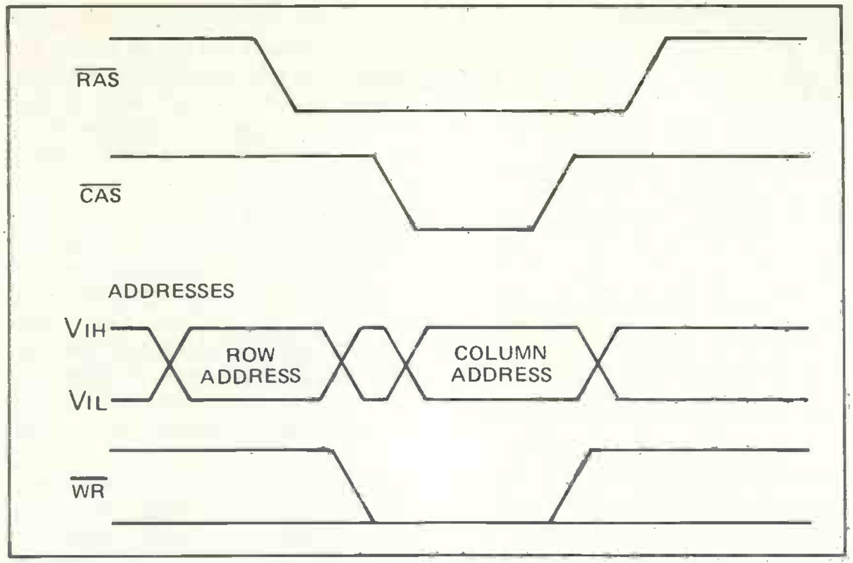


Fig. 4

address, then the chip will perform a refresh cycle and keep the data inside the chip valid. We still need to cycle through all the combinations of row address but thankfully the Z80 automatically does this for us. Sending just CAS will tri-state the on-chip data output latch and so present a high O/P impedance enabling us to connect several blocks of 4k bytes of memory together. If we send RAS then CAS, we will access the specified bit and, depending on the status of the WR pin, perform a read or a write and enable the output latch. The basic waveforms are given in Figure 4. This completes a very cursory look at the internal workings of the RAM and I suggest that you get hold of some data sheets for any further bed-time reading that you may wish to do!

I have deliberately ignored one pin (CS) until now. This is because in the 4116 chip, this pin is used for the seventh address line (remember that a 14 line address is now required). If we allow a combination of RAS and CAS to occur for a valid address, then we can dispense with CS completely and permanently tie it low. However, we must remember to allow for expansion to 16k and link E does just that. The other links from F to L are there to accommodate the changes necessary to the address lines incoming to IC1 and 2 for the 16k upgrade — see 7b.

We need to always allow a CAS through as soon as possible after a valid memory read or write in order to disable the data output latch. A suitable signal would be MREQ but it is rather too long for our requirements. Instead, we clock it through a latch about one clock cycle later on and use MREQ again to clear the latch down so that CAS will disappear at the correct time — see Figure 5. This also gives us the appropriate delay requirement after RAS for whenever we really do want to access the chip.

However, we must inhibit CAS during RFSH and so there is some further gating necessary. The generation of CAS is provided in SMART 1 by IC5/2, 4a, 9/6, 8/1 and 9/8. This signal is in fact used to switch over the address multiplexer (normally switched to the row address) and one of the spare inputs to this multiplexer is used as the actual CAS strobe to allow for any timing delays as the multiplexers changeover. CAS is further buffered by a normal TTL gate —IC14/6.

RAS is needed either during refresh or whenever a valid address has been recognised. Since we are designing for two blocks of memory, there are two RAS signals required. The actual gating is performed by IC6, 3/4 and 1. IC3 is a standard TTL device in order to provide adequate drive to the inputs on the chips.

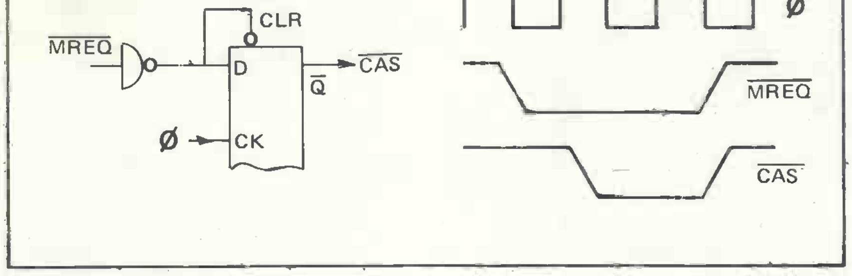


Fig. 5

Since we've gone to the trouble of buffering the bus, we really ought to do the job properly and buffer the output of the RAM. This is not done to prevent bus conflicts, since CAS will tri-state the RAM's anyway, but is done to provide sufficient drive for future expansion. We're going to do this job properly, right? We need to enable the buffer whenever either valid address is present together with RD and MREQ. This is done with IC6/13, 3/13 and 9/4. We also must remember to take DBDR low and that is done with IC14/2. There are just three minor points left to explain.

The first is that the line labelled RAM DISABLE can be used to access a ROM instead of RAM when they both live at the same address. The line is open collector and is pulled down low to disable RAM. If wish to use this facility on SMART 1, then simply connect link M. The second point is that the address bus is not guaranteed to be valid past the rising edge of MREQ on an Op Code Fetch. The trick is to latch the four important address lines (A_{12} to A_{15}) while MREQ is high. This will then prevent any spurious decoding of valid addresses occurring at the wrong time which would cause a 'glitch' on the RAS lines and possibly destroy one row of data. The latch is IC10. One added bonus of doing this is that the 7475(IC10) has complementary outputs which is just what we need on other boards for decoding other addresses. You *could* use some of the spare lines on NASBUS to carry the latched address lines.

The final minor point is to modify the Reset line due to the use of dynamic RAM. Since these chips need continual refreshing which is done by the CPU, any ham-fisted use of the normal reset line will suspend all operation within the CPU and hence there will be no refresh. We use the reset key to trigger a monostable having a short time constant. The triggering must be synchronised to M1 to prevent spurious operation of MREQ, just after reset, upsetting our RAM. This reset is achieved using IC's 5/4, 4b, 15 and 9/12. Power-on reset must also be automatically provided for since there are no M1 cycles available on switch on. The reset line to NASCOM pin 12 on SK1 must be disconnected from the NASCOM board and taken to NASBUS pin 10 instead. This enables you to use the Reset button on the keyboard. Alternatively, you can use a separate reset switch connected to NASBUS pin 10 and use the keyboard Reset as well if you aren't worried about corrupting your data in dynamic RAM — see 7c.

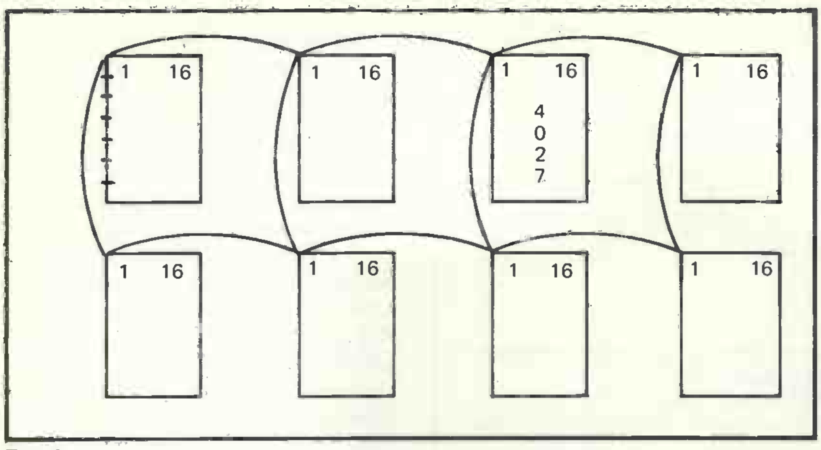


Fig. 6

Constructional details

The only critical layout is for the dynamic RAM. The power lines should be gridded — Figure 6 shows what I mean by that. The reason is that most of the energy during operation of the dynamic RAM is supplied by the decoupling capacitors in the form of a very short burst of some 60mA per chip. The actual drain on the power supply is minimal and I measured a current drain of 200mA from the +5 supply and only 10mA from the +12v rail. The latter was only an average reading but still indicates that your existing power supply should hardly notice the addition of this board.

A 0.1uF capacitor should be connected between +12v and ground every other device; and a 0.1uF between —5v and ground alternately with the previous capacitors. A 10uF tantalum should be connected between +12v and ground near the array and perhaps a 3.3uF could be put between —5v and ground. The +5v rail should be decoupled with a few 0.1uF capacitors but is not nearly so critical as that required for the other two rails.

Figures 7a-d shows complete circuit diagram and Figure 8 ties up the loose ends! A printed circuit board and/or kit is being made but at the time this article was being written, the final price had not yet been assessed. However, a conservative estimate puts the total cost for buffer board, ic's and 8k of RAM at under £90.

PCW *For more details write only, with an sae, to the author at Blackberries, Sherriffs Lench, Evesham, Worcs.* PCW.

SMART 1 I.C No's.

IC	1 — 74LS157	6 — 74LS02	11 — 81LS95
	2 — "	7 — 81LS95	12 — 81LS95
	3 — 7402	8 — 74LS02	13 — 74LS155
	4 — 74LS74	9 — 7414	14 — 7405
	5 — 74LS04	10 — 74LS75	15 — 74121

16–23 = 4027
24 = 81LS97 or 81LS95
25–32 = 4027
33–34 = 81LS97 or 81LS95

ALL POWER SUPPLIES ARE PIN 7 = 0v, PIN 14 = +5v

APART FROM:—	0v	+5v	+12	—5
81LS95/97	10	20		
74LS75	12	5		
74LS155/157	8	16		
4027	16	9	8	1

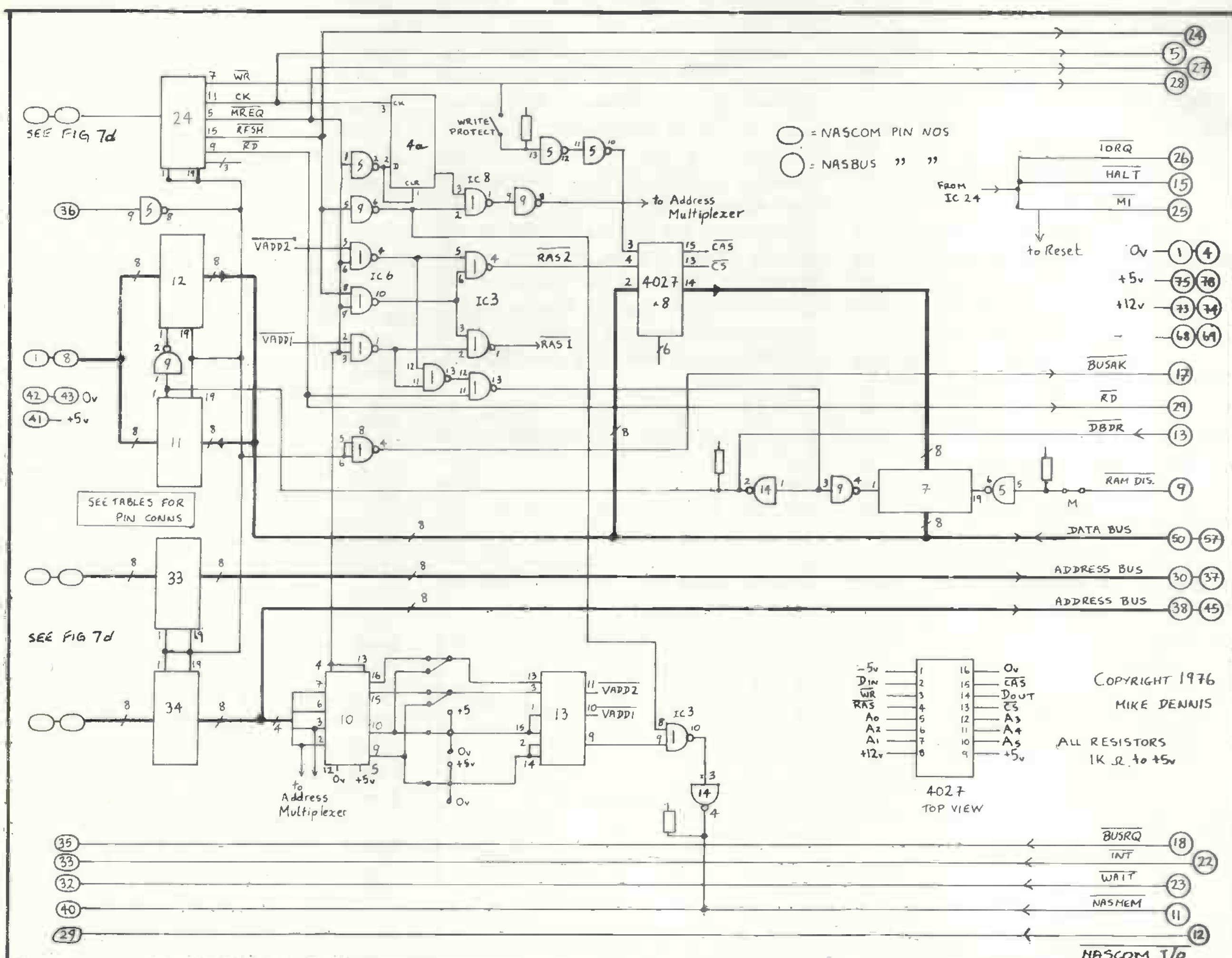


Fig. 7a

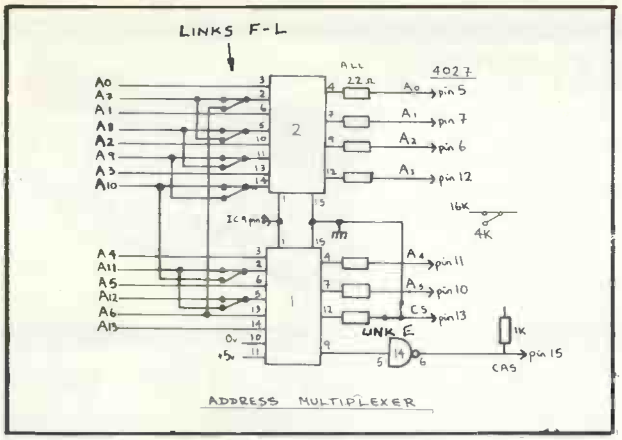


Fig. 7b

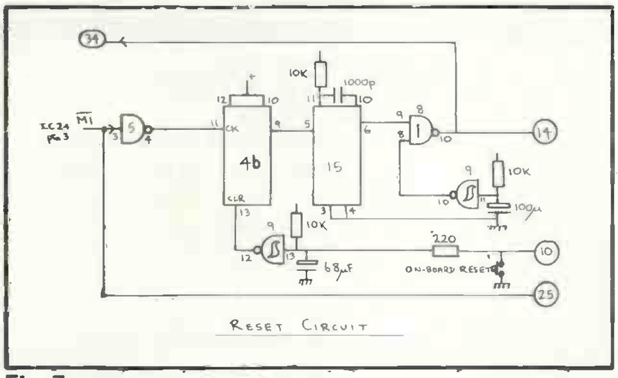


Fig. 7c

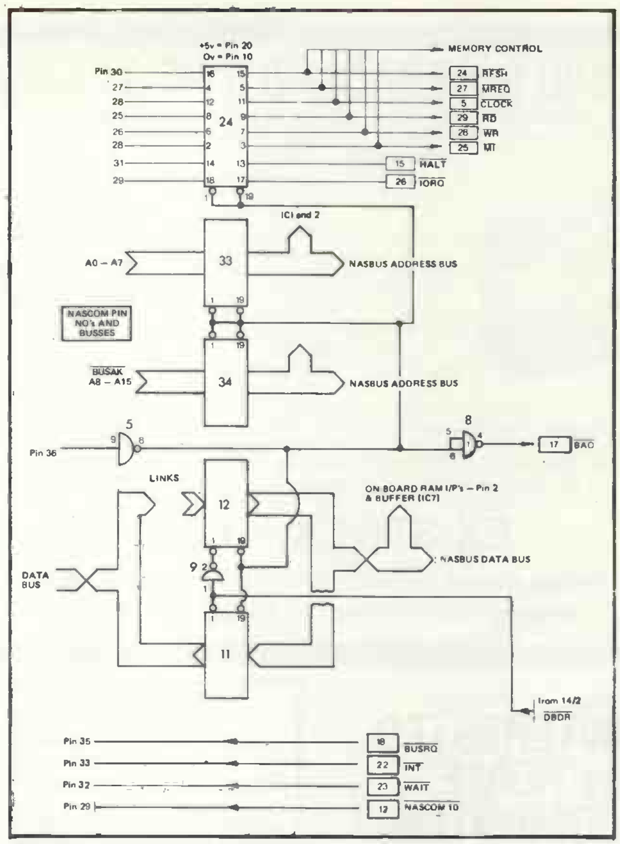


Fig. 7d

NASBUS POWER & AUXILIARIES

DESIGNATION	PIN NO.				
Ov	1 – 4	Reserved A16 – A19?	46 – 49	– 12v	70 – 71
Spare	6 – 8	Reserved D8 – D16?	58 – 65	Keyway	72
Reserved for NM1	21	Unused	66 – 67	+ 12v	73 – 74
		– 5v	68 – 69	+ 5v	75 – 78

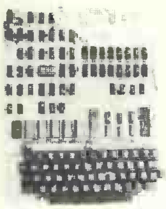
Fig. 8

Situations Vacant

PCW is read by a vast number of the most intelligent and upwardly mobile people in Britain. Advertise your job and staff vacancies here.

OHIO SUPERBOARD II NEW — FIRST IN UK — ON DEMO

For electronic buffs. Fully assembled and tested. Requires +5V at 3 Amps and a video monitor or TV with RF converter to be up and running.



STANDARD FEATURES

- Uses the ultra powerful 6502 microprocessor
- 8K Microsoft BASIC-in-ROM
  Full feature BASIC runs faster than currently available personal computers and all 8080-based business computers.
- 4K static RAM on board expandable to 8K
- Full 53-keyboard with upper/lower case and user programmability
- Kansas City standard audio cassette interface for high reliability
- Full machine code monitor and I/O utilities in ROM

Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters; without overscan up to 30 x 30 characters.

EX-STOCK — £263

Please phone to check availability or send £10 to reserve one and pay the balance on delivery

KEY BOARD

756 GEORGE RISK

Brand new professional ASCII keyboards (USA)

Full technical details included. **RRP £60.00**

Only **£49.90** + 8% VAT.

Ready built, tested and guaranteed.

TRS 80 SOFTWARE *NEW* £39.00

100 MIXED PROGRAMMES on cassette

Educational Games, Graphics, Finance, etc. on 5 cassettes plus Full Documentation.

COMP PRO Mixer

Professional audio mixer that you can build yourself and save over £100.

6 into 2 with full equalization and echo, cve and pan controls.

All you need for your own recording studio is a stereo tape or cassette recorder.

This superb mixer kit has slider faders, level meters and additional auxilliary inputs.

Only **£99.90** plus 8% VAT for complete kit Plus FREE power supply valued at £25.00

Ideal for

DISCOS STAGE MIXING HOME STUDIOS
AND MANY OTHER APPLICATIONS

INTERESTED IN HOME COMPUTING?

FREE B BUG valued at £23.00 plus 10 x C12 cassettes valued at £4.00 plus Standard Modulator valued at £2.25 **WITH EVERY NASCOM**

Start now and don't get left behind THE NASCOM 1 is here Ex-stock with full technical services

Plus the opportunity to join the fastest moving club of personal computer users enabling you to get the most our of your computer. You can OBTAIN and EXCHANGE programs and other software — many now available.

SAVE £60

- The Powerful Z80 Microprocessor
- Professional Keyboard
- 1 Kbyte Monitor in EPROM
- 2 Kbyte RAM (expandable)
- Audio Cassette interface
- Plugs into your domestic TV
- Easy construction from straightforward instructions
  - no drilling or special tools
  - Just neat soldering required.

Only ~~£197.50~~ + 8% VAT (includes p & p + insurance)

NEW LOW PRICE £165

Manuals seperately	2.95
Z80 programming Manual	6.90
Z80 Technical Manual	2.95
PIO Technical Manual	2.95
(All prices add 8% VAT)	
Power supply suitable for NASCOM	19.90

NASCOM AD ONS — Nascom improved monitor B Bug (2K) featuring — *Four times tape speed *Direct text entry without ASCII *Extended keyboard facility *Additional useful subroutines **£23.00**

Nascom Vero Case **£22.50**

Nascom Joy Stick Kit **£14.90**

Nascom Music Box Kit **£9.90** *(write your own tunes and play them on your Nascom. Complete with full documentation).*

GRAPHICS ADD ON BOARD £9.90
Complete kit to upgrade your NASCOM for graphics capability includes full documentation and demonstration program.

Teleplay presents the PROGRAMAGAME of all time!

TELEPLAY have produced a product of cartridge games, for you and all your family, that will not go out of date. Just ADD each new cartridge as it becomes available, and your colour video centre will be up to date the Teleplay way. You will save pounds in the long run. Works OK on Black and White TV.

EX STOCK IMMEDIATE DELIVERY

REALISTIC SOUND EFFECTS

The Teleplay consists of a Professional Finished Inject Moulded Box Joystick Hand Controls one cartridge which consists of 10 games Football, Tennis, Solo Squash (1 & 2 players), Hockey, Basketball (1 & 2 players), Shoot etc. Mains Adaptor TELEPLAY GIVES YOU TWICE THE FUN BUILDING AND PLAYING THE MOST PROFESSIONAL TV GAME KIT EVER OFFERED.

We offer all this to you for ONLY **£28·90** + £2.31 VAT ~~Fully Assembled £35.90~~

SAVE £10

Main Console has 3 Facility Handicaping Switches

Every 3 months a NEW game will become available to you and those already under production include Submarines, Space War Wipe Out and many more. The price of these will vary from £12 to £19 depending upon the complexity of the game

Additional cartridges NOW available include

Stunt Cycle	4 games	Cartridges and Hand Controls	£14.90
Tank Battle	Cartridge and Hand Controls		£18.90
Road Race	2 games	Cartridge Controls already with basic kit	£12.90

NEW— WIPE OUT cartridge — 6 games £13.90

Cartridges come complete with printed case and easy to follow instructions

All parts fully guaranteed

MODULATORS UHF Channel 36

Standard 6 meg band width **£2.25**
High Quality 8 meg band width £4.90

SHORT C12 CASSETTES 10 for £4.00

FOR COMPUTER PROGRAMMES

Orders over £5 post and packing free otherwise add 20p. Please make cheques and postal orders payable to COMP, or phone your order quoting BARCLAYCARD or ACCESS number.

OPEN — 10am to 7pm — Monday to Saturday
CONTINUOUS DEMONSTRATIONS

SEND FOR OUR SPRING '79 CATALOGUE - 0·60p

Owner's Report

IN DEFENSE OF PET

Tyrone Crudis

PCW Readers should refer to Vol. 1, No. 2 for the first PET Review by our Consultant John Coll. We were waiting for just such a brilliantly written article before presenting a devotee's account of the PET. This acolyte prefers to remain anonymous: his pen name is from the Latin "Tyro nec Crudis"; a free translation gives, " a newcomer but not without experience". PCW

I selected PET over all available competitors by making a list of desired features, present and future, and then drawing up a matrix of vendors vs. specs to determine who offered how much of what I wanted and at what price. The matrix was constructed in October 1977, so don't ask me for a copy: it's hopelessly outdated by dropouts and new additions. Decisions made, finances shuffled, and wife mortgaged, I took delivery in late April, '78, so that by mid-June, when John Coll's rather curt review of PET was published, I already had many happy flying hours under my belt.

I'm sorry Mr. Coll couldn't get a PET for long enough to please him, and I'm sorry that, as a reader later pointed out, it seemed to show in the tone he took. Can anyone tell me what computer he was comparing it with that can be had for £695 including keyboard, VDU, cassette, monitor, 6K operating system and 8K BASIC in ROM, power-on jump start, lower case, 63 graphic symbols, and VAT? You have to purchase APPLESAUCE or CANDY LEVEL II "extra cost options" to get the Basic that is standard with PET. And ask your friendly computer dealer if Brand X can retrieve named files. - That is, when you print LOAD "PONTOON", will the computer search the cassette, rejecting all other programs, until it reaches and loads the one you have selected? Ask him whether you can intermix characters and graphics on the screen. Ask what *extra* cards you have to buy before you can communicate with the outside world! Ask him if you get more than one array or more than 26 variables!

The author goes for a ramble with his PET

Well, I, who have no connection with the manufacturer or related industries, feel that an injustice has been done to PET (did you guess?) and I propose to ramble for 3000 words or so to try to correct that situation. Some of the time I'll be agreeing with Mr. Coll and emphasizing some of the positive points he makes. Elsewhere I'll be disagreeing with him, taking issue without, I hope, giving umbrage. And now and then I'll be bringing up a few facts that have developed in the time since he wrote.

It might be well to start with that last category. Documentation is vastly improved from the early days. Here is a list of what I have on hand, all but one available from Commodore, UK.

1.) "An Introduction To Your New PET": 38pp typeset with many illustrations; a good head-start for the *totally* uninitiated new owner and a handy reference for PET's BASIC vocabulary.

2.) "PET Communication With The Outside World": 53pp, matching format to the above: describes the IEEE-488 interface, parallel user port, etc., how tape files are recorded, how to patch a few bugs in the tape handling section of ROM, and other I/O information. (PCW *But see PCW Vol. 1, No. 11 article "Unravelling the PET Bus")*

3.) "Commodore PET Users' Handbook": 134pp mimeographed, evidently generated by Commodore UK in a hurry to fill the gap. Far from perfect, it nevertheless is full of useful information. Some samples are: A discussion of error messages which the computer generates. There are over 30, many identifying the program line number in which the programmer has erred. Space hints, speed hints, and hints on converting programs written in other dialects of BASIC. Discussion of the USR function, which allows the programmer to create a machine-language subroutine callable from BASIC.

4.) "MCS 6500 Microcomputer Family Hardware Manual": 165pp spiral-bound fully detailed descriptions of the MCS 6508 CPU and its support chips.

5.) "MCS 6500 Microcomputer Programming Manual": 240pp, format matching the above, Xerographic reproduction of photographically-reduced typescript. Complete explanation of machine-language operations.

6.) "Condensed Description of the Hewlett-Packard Interface Bus": this is an H-P publication, also available from Commodore, which in 18 printed pages with diagrams gives concise details of what is now almost universally referred to as the IEEE—488 Bus.

7.) Four issues of the (UK) "Commodore PET Users' club Newsletter": each issue has been full of useful data, programs, ads for other vendors' programs and add-on devices, and listings of Commodore's program library (nearly 100 items at present, including disassembler, machine language monitor, machine code handler, and hex editor and loader.) A nice renumbering program, not listed here, is available elsewhere and a desirable addition to anyone's formulary. The current issue of the CPUCN is 41pp, well-reproduced from typewritten material.

8.) Six issues of the (USA, non-vendor-related) "PET User Notes": everyone should have these! An airmail subscription is $12 from PET User Group, Box 371, Montgomeryville, PA, 18936, USA. They're full of inside information on the manufacturer's plans, games, users' hints, utility programs, ads, and a library now numbering over 90 contributed programs available at $2.00 for the first program plus $1.00 each for each additional program up to four per tape, or free in exchange for donated original programs of quality. The current issue is 16pp of small print, well-printed, and much denser than CPUCN. Features in this issue alone include an article on optimizing tape head alignment, a program which indexes the tape for storing multiple programs by timing in fast forward, another on changing BASIC statements while a program is running, a schematic for a game console with two joysticks, two pushbuttons, and a loudspeaker, and many others. I am taking the liberty of reproducing elsewhere here a fascinating and really testing *four-line* game, also included in this one typical issue.

I did not find any of Commodore's advertising material misleading, with one exception: they refer to multi-dimensional arrays as 'matrices', and I had therefore expected to find matrix commands in BASIC.

The author's PET is pretty

What are some of the things which John didn't mention or didn't emphasize sufficiently in my prejudiced opinion? Let me talk about the video display. I don't know what bandwidth the monitor provides, but it's far above what you can get from any unmodified home TV, and the effect is electrifying if you've never seen anything but an RF-modulator-driven home idiot box. "Crisp" isn't a strong-enough word. In REVERSE mode, with white screen and dark characters and graphics, it is even more effective, as the normal spot bloom works to make lines even thinner and sharper. You can select NORMAL or REVERSE mode character-by-character. Let me give you a feel for the display by explaining that it contains 40 X 24 = 1000 "pixels" or picture elements, each with its dedicated memory byte, any one of which can be read and changed with PEEK or POKE. Each picture element consists of an 8 X 8 contiguous dot matrix. There are no forced spaces or columns as in some micros, so that the 63 special graphics available in SHIFT mode can be connected together to form continuous lines and complex pictures. (see Figure 1).

Compare this with what you can do with Brand X and six squares per pixel! Those who lust after color, incidentally, should be reminded of some facts of life: color TV resolution is inherently worse than B & W TV resolution. Professional color monitors are great, but very expensive. On top of the poorer resolution there are further restrictions on the number of color changes per inch, due to bandwidth problems as the effective bandwidth per color is $\frac{1}{3}$ of the total for a three-color screen, so pure color thin vertical lines are not possible.

Have you heard that Commodore has already delivered over 5000 units in the UK alone? That they are now to be built here, initially at the rate of 500 per month? This kind of market response in a few months in the face of Commodore's exceedingly low-key advertising bodes well for a tremendous body of dedicated users which will make moot the questions of tape-exchange compatibility with other manufacturers' machines raised by J.C.

CUTS is no more a universal standard for cassette tape recording than several others he does not mention. I think Commodore have done well to *go their own way* with a tape recording technique that uses a full redundancy check for highest reliability. (Yes, each program is written twice on the tape, and readout is not passed unless the two versions are identical.) Of the many sources with which I have exchanged tapes so far, not one "foreign" tape has failed to read on my machine the first time. To be fair, I have heard of others who have had a problem; this invariably is solvable by a very slight adjustment of the tape head alignment.

The author jiffies around the Clock

John (I'm beginning to feel like an old friend) criticizes the pseudo-random number generator because it is not truly random. That's like requiring the Thermos Bottle Company to call it a pseudo-vacuum flask! Every micro BASIC RANDOM function I'm aware of is equally "pseudo", and this fact is well known. PET's RND(X) has the useful special features that the choice of a plus or minus argument determines whether you will always get the same number for a given argument (useful for testing) or a series of differing numbers (a different series for each argument). For a given positive argument or seed the same series will always be generated, starting with the same number.

I overcome this where it is a problem by using PET's clock to run off a number of terms in the series corresponding to the time in jiffies since startup, before entering the series in the program. PET's clock? Nobody told you that the firmware includes a real-time clock? It counts jiffies (units of 1/60 second) since startup, non-resettable, always accessible as the dedicated variable TI. It counts clock time, six digits of hours, minutes, and seconds, resettable to real time and accessible as TI$. There are also 1 µsec and 250 µsec clocks, and two which step at about 4-sec and 18-min periods respectively. This whole subject is treated exhaustively in the article, "Measuring time on the PET and Other Microcomputers" by Larry Tesler in the magazine "People's Computers", Vol. 6 No. 5, p48, no doubt available from advertisers in PCW.

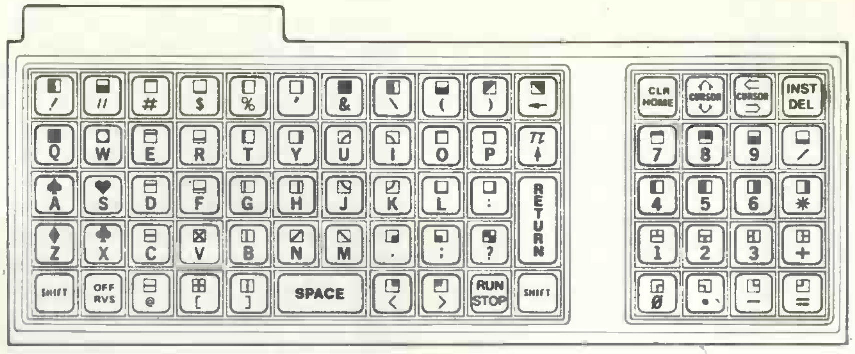
The very same issue contains a listing for a program called DRAW 8K, which gives you repeat keys in eight directions! Select a graphic, hold down the proper key, and draw a line of symbols on any horizontal, vertical, or diagonal! A great time-saver. If you get as good as I am, you, too can draw Fig. 1. The program also saves your drawing either as a data file or as a BASIC program on tape and includes a mode which uses the four-squares-to-a-pixel graphics (16 of them, count 'em, including REVERSE field of the first eight, see Fig. 2) to draw with a resolution of 76 across by 48 down. Perhaps if there's enough interest the editor will seek permission to reprint this article. If he does, I will volunteer to fix all the devious spots which gave me a week of work before I had the program up and running. This four-square-per-pixel mode makes PET highly suitable for display of the famous game of LIFE. Those of you who don't know what LIFE is have only the editor to complain to.

For the inner circle, let me just say that Frank Covitz in the USA has developed for PET a machine-language LIFE program, 64 X 64 elements, large enough to display all but the monstrous growth patterns at a rate of better than two generations per second! How chagrined I was when my 24 X 40 Life, written in BASIC and taking up to two minutes per generation, became obsolete. The new program is available from PET User Notes in the USA aforesaid. I don't mind giving them the free advertising since they're a non-profit society. Why don't you drooling PET lovers out there buy £4.50 worth of international postal thingummies and send them $7.00 ($5.00 plus $2.00 for airmail) asking for LIFE 64 X 64, 24—SEC. QUBIC, ELIZA, and STARTREK IV? 24—SEC. QUBIC is three-dimensional Noughts and Crosses, 4 X 4 X 4, in high-speed machine language.

ELIZA is a micro version of the famous full-scale program, an amusing attempt to portray the computer as a psychologist. STARTREK IV uses every byte in the box and extensive compression techniques to give you all the standard features of the famous game as seen on many much larger computers plus the advantage of PET graphics for a continually-refreshed display. STARTREK IV is another program the editor might be interested in getting reprint approval for. Apparently it's numbered IV because it is three generations ahead of other commercially-offered games with things called Klingons and Enterprises in them. I assure you that when you load it and write PRINT FRE(E), the number you'll get will be less than 400 bytes, and you'd better leave them available in case the program decides to open a new array in midgame. (The dedicated function FRE (X) gives you the number of bytes of available unused memory at any time, another unsung PET extra.)

The author cannot resist crowing

A few more goodies to crow about: on power-up start, a checkerboard pattern is written and read back to check the entire memory, and an all —OK message is displayed. Other diagnostic routines are also built into the ROM and are useable by any authorized PET dealer who simply plugs into the user port and runs a complete diagnosis a la VW without opening the case. The famous MOS 6522 Versatile Interface Adaptor is the power-behind-the-throne of the IEEE—488 bus provisions. It would take another article the size of this to detail its many functions. Let me whet your appetite by stating that Commodore list over 128 IEEE—488 compatible devices from various manufacturers, and that the bus is fundamentally a 16-line cable which goes in parallel to up to 15 such instruments. One at a time may be the controller, one a talker, and up to 14 may be simultaneous listeners.



The keyboard

A network of PETs! Eight lines transmit serial bytes and eight lines are used for timing and control. Seven-bit ASCII code is used, with the eighth bit available for parity check or other uses. It accommodates asynchronous communication over a wide range of data rates. Eventually, in my 'umble hopinion, IEEE—488 will be *the* standard for printers, display devices, and the vast family of controllable scientific instruments and data logging devices. I understand that it is also under consideration by the IEC as an international standard.

The 6522 VIA can also be used to generate complex timing patterns, service external interrupts in a burglar or fire alarm system, generate fixed-duration solenoid-driver pulses, interrupt the main program to display time of day at desired intervals, generate pulse-width-modulated signals for radio control of robots or remote devices, synthesize waveforms for sound effects or game controls, and many other mind-boggling applications. Get the chip spec from Commodore if you don't believe me!

The user port contains an eight-bit parallel input-output register in which each bit can be individually programmed as an input or an output, plus TV video *and* TV horizontal and vertical sync for external displays. Why don't they tell you all this? They're afraid you'll overload, crash, or abort under the impact of it all! Ekshluh, I think they wrongly visualize their market as the top income level of the group which buys TV games, and they hope they'll sell you first and get you to learn BASIC second. Perhaps we shouldn't disillusion them!

The PET printer is promised for any day now, and may have been seen by the time this is in print. I am told that the original version has been scrapped and that the new version will use the C. Itoh mechanism and will be available in roller and tractor form feed options. The specs are expected to include upper and lower case, double-width characters, *complete graphics*, a programmable character, and special formatting such as left, right, or decimal point alignment and others, apparently under the control of an on-board chip. Memories of FORTRAN!

The author does a Schweppes

You-know-who objected to the keyboard, saying that the keys were too small to use comfortably at speed. For your information, in case you haven't been able to scrounge some hands-on time, they are on ½" spacing. My typewriter keys are on ¾" spacing, but the significant difference to touch-typists is that typewriter keys are offset row to row, where PET keys are all in straight columns. You will probably not be able to touch-type on your PET. By the same token, you will probably not have two-key rollover on your typewriter or be able to insert characters or delete them, with the line opening up or closing down in length accordingly. I find the speed which I achieved after a few days' practice to be highly satisfactory, and consider that the criticism is a quibble.

On the other hand, it is a major defect that the keys are marked with an anodized aluminium stick-on label instead of the industry-standard two-shot moulding wherein the character goes through the full depth of the key and cannot be worn out. In an attempt to keep prices down, Commodore has settled for a key label which wears away in a few months to be almost illegible on the most-used keys. True, you have learned the keyboard by heart by that time, but what about your nubile young friend of the opposite sex whom you want to impress? What about the cosmetic effect? Speak, PET, and tell us what you plan to do about it!

The author fulfills a promise

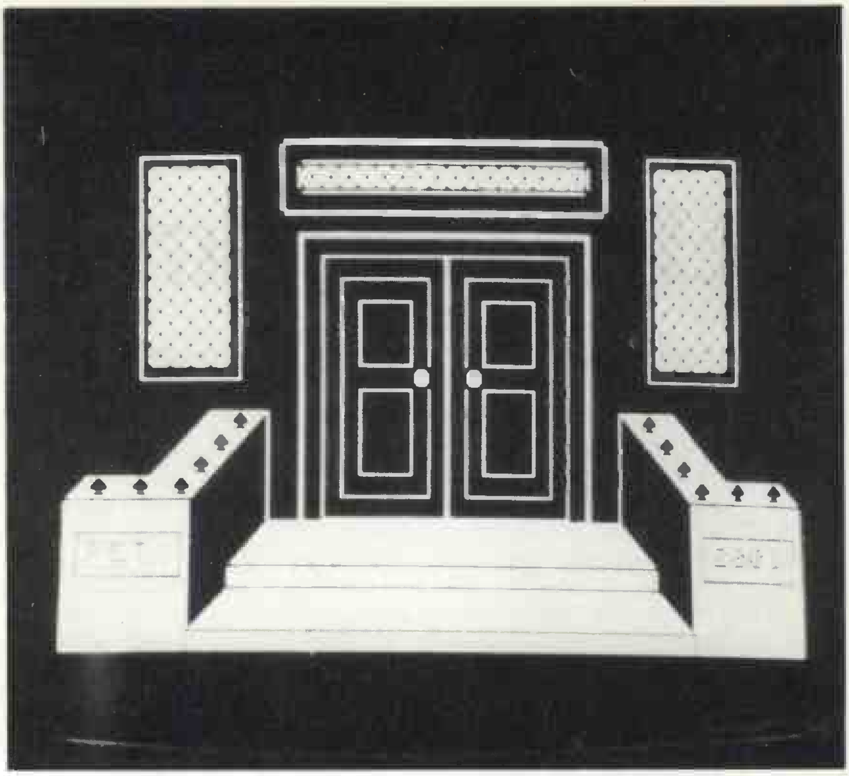
Oh, yes! Here's that four-line game I promised.

It's by Andy Fraley of Reading, PA, USA. The object is to move yourself ("*") through the maze to the "$" within the allotted time, using the key "5" to go down, "4" to go diagonally left and down, "6" to go diagonally right and down, and "9" to go to the right.

SHORTMAZE

```
10 PRINT "clr": FORI=32808TO33765:
   POKEI,INT((3*RNO(10))+1)*32:
   NEXTI
20 TI$="000000":A=32768:POKE80,0:
   POKE84,39:POKE86,41:POKE89,1:
   POKE85,40
30 GETS:A$=STR$(PEEK(S+80)):A=
   A+VAL(A$):IFPEEK(A)=64ORTI$
   ="000100"THEN10
40 POKE33767,36:POKEA,42:PRINT
   "h"SPC(12)TI$:GOTO30
```

I'll leave it to your imagination to add instructions and win-lose read-outs.



ON THE LINE

David Hebditch

5. Communications Interfaces

Last month we established that we need some kind of *modem* (modulation/demodulator) to get data bits from one end of a dialled telephone line to another. Start-stop timing enables us to do this on a character-by-character basis. Even though it is most convenient for us to think of data being transmitted on a character-by-character, it must be remembered that the modem only deals with individual bits.

This means that the characters we want to transmit need to be converted from the 8-bit parallel form in which we conventionally store and manipulate them in our computer to bit-serial form for transmission (and, of course, the other way round for reception). This serialization and de-serialization is done on our interface board, something that will be considered in more detail in a later article.

But once we have a stream of bits to transmit how do we get these across to the modem? Our line controller board will be connected to the modem using a standard interface arrangement (see Figure 1). Such communications standards are established by the International Consultative Committee for Telegraph and Telephone (the CCITT) which is part of the International Telecommunications Union (a UN body).

The standards which relate to the use of the existing telephone network are called the V-Series. The V24 Interface Recommendation comprises a list of 36 possible interchange circuits for data transmission and reception. *This is why you will see* many items of equipment described as being 'V24 compatible' or 'RS-232 compatible' the latter being the *equivalent* US standard.

This is somewhat inaccurate in the sense that there is no such thing as a 'V24 interface'. The actual interfaces are subsets of the available leads specified by V24. For low-speed asynchronous (start-stop) transmission of the sort that we are interested in, the relevant interface is V21 and the interchange circuits used are shown in Figure 2. Signalling between the interface board and the modem is as follows:

+6 volts (represents "0" or "ON")
—6 volts (represents "1" or "OF")

The voltages are nominal (in the range 5 to 25 volts) and are with reference to the Signal Common Return circuit on P in 7.

The first two leads, TRANSMITTED DATA and RECEIVED DATA are used to transfer data bits between the interface board and the modem. The next four leads (105, 106, 107 and 109) are used to control the transmission of data.

The procedure for transmitting is as follows:

1. Set REQUEST TO SEND (105) ON.
   This causes the modem to start transmitting the carrier frequency.
2. Wait for READY FOR SENDING (106) to come ON. The delay is caused by the time it takes for the carrier to become stabilized and should not normally exceed milliseconds.
3. Start transmitting data on the TRANSMIT DATA (103) circuit. Note that in asynchronous transmission the timing needs to be provided by the interface board.

The procedure for receiving data is as follows:

1. Wait for the DATA CARRIER DETECTOR (109) circuit to change to the ON state.
2. Start sampling the incoming signal on the RECEIVED DATA (104) circuit. The lead will normally be at —6 volts (binary 1) and the first change to +6 volts (binary 0) will indicate a character start bit.

Because the circuit is FULL DUPLEX (capable of simultaneous bothway transmission) the above send and receive functions need to be able to operate in parallel. However, the detailed operation of the interface is usually a function of the interface board and need not concern the user

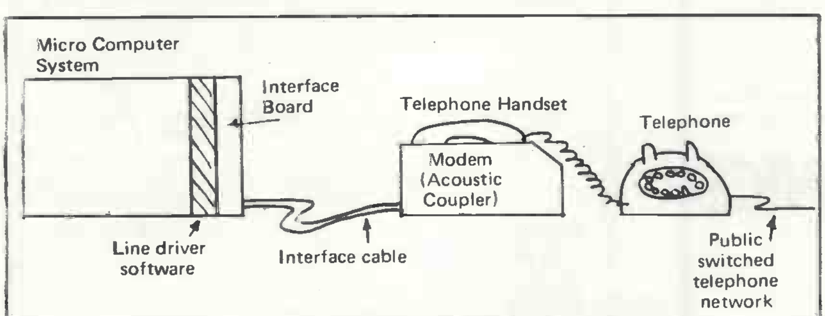


Figure 1: Interface Arrangement for Data Transmission

CCITT V24 CIRCUIT NUMBER	CONNECTOR PIN NUMBER	FROM MODEM	TO MODEM	CIRCUIT NAME	
103	2		X	TRANSMITTED DATA	DATA
104	3	X		RECEIVED DATA	DATA
105	4		X	REQUEST TO SEND	CONTROL
106	5	X		READY FOR SENDING	CONTROL
109	8	X		DATA CARRIER DETECTOR	CONTROL
125	22	X		CALLING INDICATOR	CONTROL
108/1	20		X	CONNECT DATA SET (MODEM) TO LINE	CONTROL
107	6	X		DATA SET (MODEM) READY	CONTROL
102	7			SIGNAL COMMON RETURN	GROUND

Figure 2: Interchange Circuit Designations for the V21 Interface

too much. The above explanation is provided to help in subsequent 'de-bugging' of the interface.

The remaining three circuits shown in Figure 2 (125, 108/1 and 107) are only used for modems which are 'hard-wired' to the telephone line. It is not possible to employ them with the acoustic-coupler type of modem we shall be playing with in later articles. Their use enables the computer to automatically connect to an incoming call on the public telephone network. Operation is as follows:

1. Wait for CALLING INDICATOR (125) to come ON. This is the equivalent to hearing a telephone ring out.
2. If the computer is able to accept the call it should set the CONNECT MODEM TO LINE (108/1) circuit ON. This is equivalent to lifting the handset from the receiver.
3. Wait for the MODEM READY (107) lead to come ON, indicating that it is OK to start transmission or reception.

Hopefully, the Post Office will before long make it possible for low-cost modems supplied by themselves or by the industry to be jack-plug connected to exchange lines for automatic call-reception. There are no major technical reasons why this cannot be done.

The 25-way D-type connector plug and sockets are used as standard for the V-series interface arrangements. (The actual cable used need have only enough leads for interface being used at least 6 or 9 in the case of V21). Unfortunately, what is not standard is which component should have the male and which the female connectors. Even in these days of Gay Liberation, it is pretty difficult to get a male connector into another male connector (and the same with the female ones). I bet we have this problem when we start plugging things together with the D-type connectors, that is!

Next month: Interface boards and the ubiquitous UART.

PCW OPEN PAGE

THE AMATEUR VIEW

Mike Lord

The North Kent Amateur Computer Club has now established itself with regular meetings being held on the second Tuesday of each month, usually at Charles Darwin School in Jail Lane, Biggin Hill, although occasionally alternative arrangements have to be made, and their meetings are typically attended by 30 members. Recent meetings have featured, among other speakers, Bill Strong of the SIRA Institute and a Z-80 evening proved very popular. NKACC secretary Barry Biddles has made the worthwhile suggestion that local groups who have discovered members within their ranks who can give an interesting talk or demonstration should 'trade' them with other groups. This would spread knowledge and go some way to solving the Club Secretary's perennial problem of thinking of new Club activities. Anyone wishing to take Barry up on his suggestion, or to find out more about the NKACC should ring him at Biggin Hill 71742.

The East London ACC is also holding regular meetings – on the third Tuesday of each month – in the Meeting Room of the Harrow Green Library in Cathall Road, Leytonstone from 7 to 10 p.m. Their membership is very varied, including hobbyists, professionals and academics, so there is always a useful exchange of ideas and experience. One of their planned future meetings will be a 'members only' evening at a local personal computer shop; this sounds like a good idea that could usefully be copied by other groups. So; those PCW readers who live within reach of Leytonstone, and who haven't already joined the ELACC, are urged to go along to the next meeting, or to contact Jim Turner at 63 Millais Road, London E11.

Another London group which has just been formed is the **South East London Microcomputer Club**, who held an inaugural event in February at the South East London College, SE4. Gil Gilhespie of 50 Lzane Road, Bexleyheath (01-303 4968) will be pleased to supply further details of forthcoming events.

Several of the larger groups are now producing their own newsletters, and in some cases to a very high standard indeed, as for example that produced by the Scottish Amateur Computer Club (ring Stewart Stevenson on Linlithgow 2657 for details of SACC activities in Edinburgh and Glasgow). But surely the best I've seen so far was sent by the Netherlands Hobby Computer Club. Unfortunately – from my point of view – it is not produced in english, but 42 A5 size pages containing articles on subjects as diverse as LISP, APL, model railway control, and a 6800 based floppy disc controller show a healthy interest in personal computing in Holland and Belgium (there are ten local groups in these two countries), and those interested are invited to write to "HCC ENQUETE, p/a POSTBUS 149, 2550 AC VOORSCHOTEN, NEDERLAND.

Three enthusiasts have called recently to say that they would like to help start up local groups in their areas. They were Mr. R. Buttery of 55 Northumberland Road., **Kettering, Northants** (tel. 0536 3154), Mr. M. Brough of 11 Beech Drive, **Kidsgrove, Stoke on Trent** (07816 4387), and for those living at the end (or is it the start?) of the District line, Bob Forster (01-892 1873) would like to start a **South West London Group** based on Richmond.

Which is a good point to say that if you wish to start a local group, or to publicise an existing organisation, then drop a line to Mike Lord at 7 Dordells, Basildon, Essex or 'phone him on 0628 411125.

POP SHOP

The PCW Open Page *Service for amateurs. Buy, sell, exchange –*

One-off advertisements only accepted. Not more than 50 words.

"**Newbear 4K RAM/Prototyping Double Eurocard, unused, £5. Vero S100 Prototyping Card** (2 available), **£10 each.**

Motorola D2 system, built and tested, manuals and special connector, unused – best cash offer or trade for V.D.U., S100 cards or anything of interest.

J.D. Millne,
38 Front Street West, Bedlington,
Northumberland, NE22 5UB.

MK14 – Built and working with power supply. Also tape interface (assembled but not yet tested). £45 Tel. Steve **01-359 2578.**

TRS-80 LOWERCASE

Get the Lowercase that is hiding in your TRS-80. Full instructions and cassette software, and all parts.

Takes only 70 bytes of memory
only £10 (level II only)

CHRIS CAIN
Eng. Wing, RAF West Drayton, Middlesex

For Sale:

77-68, CPU Card, Built working in case with PSU, front panel leds + switches, backplane + 2 edge connectors (with room for many more). Also copy of SCELBI 6800 Software Gourmet Guide. Buyer can also have Hex Keyboard and Proto PCB. Tel. Bob Buttery 0536 3154 Price £60 o.n.o.

NASCOM 1 Z80 MICROCOMPUTER

With B-Bug 2K monitor. 8A power supply for maximum expansion, Scientific calculator and keyboard all in purpose built case. Ideal for expansion to 8K BASIC. Genuine reason for sale.

£295 o.n.o.
With or without 12" Video monitor /UHF TV. £43

Nottingham (0602) 877037 Steve.

For Sale:

MK 14 Standard Micro
Assembled £40.00

Telephone **John Taylor 01-866 0646 (Harrow)**

For Sale:

Principles of Interactive Computer Graphics, authors Newman/ Sproull, publisher **McGraw-Hill,** Hardback, Mint Condition.

Half-price £6

Peter Hayden
33 Broadacres, Hatfield, Herts, AL10 9LE.

SECOND-HAND PET FOR SALE

Selling to upgrade. Complete with sound effects and light-pen. Price £590 o.n.o. Includes over £200 worth of programs.

Phone 01-997 8319.

Anyone in the Chorley/Preston area of Lancashire interested in personal computers please ring Chris on Chorley 78376 with a view to exchanging ideas, etc.

NASCOM 1 FOR SALE

Put together by profesionals and working. Included is 1K RAM, 1K ROM, keyboard, heavy duty power supply, made-up NASCOM, 10 data tapes, Aztec TV modulator, bleep bit, all the manuals.

£250.00

Warrington 65897 J. ASTLEY

I am very much interested in forming a Computer Club (local) in Nottingham. Anyone interested. I would be pleased if they would contact me at the address below or telephone after 6.30pm. Tue-Fri. Any time SAT. & SUN.

KEITH S. SWAINSON (20),
9 BRAYTON CRESCENT, HIGHBURY VALE ESTATE,
BULWELL, NOTTINGHAM (0602) 751742.

FOR SALE

High Speed 8 Channel Paper Tape Punch **£40**
Phone: 01-680 6915

TRITON Computer, Tiny Basic, graphics, 3K, cassette, uses standard TV. **£400.00.**
Mr. R. Singh. 21 Balcaskie Road, Eltham, SE9

EMU MK3870 Unused Single Board Microcomputer. One Chip F8, MK2870 Simulator, this Board contains a Clock, Programmable Counter/Timers, 64 bytes RAM, Sockets for 2 x 2708 ROM, 4 Fully Latched I/O Ports. Complete with Manuals. Any reasonable offer, or swop for RAM etc.
M. Ibison, 108 Cedar Road, Portswood, Southampton.

FOR SALE: ELF II, Built and Shop Tested, plus Power Supply and Manuals, nearly new **£90.**
Dave Climie, 91 Moss Park Ave., Glasgow G52 1LQ

FOR SALE: NM based MICROS, factory built; 1K Monitor, 2K RAM; with CED Number Cruncher & Literature; Editor & Assembler **£285**
Enquiries (0662) 2109, Omagh, Co. Tyrone, N. Ireland

Beat the MK14 queue. I have a Science of Cambridge MK14, with extra memory, up and running but will sell at cost as I am moving on to bigger things **£47.50. B.V. Mantell, 'Ardoram' Broadway, Charing, Ashford, Kent.**

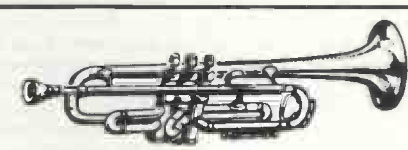
For sale excellent power supply by Coutant ideal for Micro-Computers. 5 volt 7 amp regulated DC, 24 volt 5 amp unregulated DC, and 12 volt. AC. 5 Volt supply fully protected. Circuit diagram and specification included. **£30** plus carriage.
Contact: **J. van Someren**
01-283 8566 during office hours.

FOR SALE: 6800 D2 MICROPROCESSOR SYSTEM

Extras include: 512 Bytes of on-board RAM, D/A Convertor, Power Supply Unit, Motorola's Applications Manual, Some Software. Kit built and in full working order: **£150.00 o.n.o.**
David Roberts, Pen-y-maes, 23 Underley Avenue, Kendal, Cumbria, LA9 5EU.

FOR SALE

Model KB756 George Risk ASCII Encoded Keyboard – Full Data **£48.**
System-68 CPU, VDU + B – All Socketed + Wired. All three for **£12** [no chips]
2111 – A4 RAMS bought for Triton but unused **£2** each + 5v 6a, + 12v 1a, –5v 1a PSU housed **£45**
B. Mistry, 75 St. Margaret's Road, Bradford, W. Yorks.



TRUMPET, VOLUNTARY

Responding to TIDBITS information or an advertisement?
Say you saw it in PCW
Blow your own trumpet.
Mention your magazine.

MICROS and SMALL BUSINESSES

D. R. Worsley
School of Rural Economics and Related Studies, Wye College, University of London

PCW. *It is not part of our outlook to shrink from reality. The author will make many readers pause, and think, and think again. However, we believe that this and similar articles will in the end increase the use of small business systems. Forewarned is forearmed, for both the prospective Customer and the Seller. More and more software is now being written for the small businessman.* PCW.

Will this future arrive?

No one can fail to be struck with the progress that has been made in miniaturisation, reductions in cost, and innovation which has made the micro computer available to anyone who can afford the price of a car — new or secondhand. Billed as suitable for personal, home or business use, they are portrayed as "an answer" and "the future". Peter Walker is quoted as saying that micros will mean the end of drudgery, and there have been many articles and programs in all the media concerning their development.

Of all the questions raised, three stand out, of particular interest to people connected with the micro industry:

(i) are micros yet ready to help small businesses?
(ii) are small businesses anything like ready to use them?
(iii) what will be the generation gap in, say ten years' time, when most children will leave school with some computer background?

I went to the *Personal Computer World* Exhibition in the hope of answering at least the first of these questions.

A busy exhibition, noted especially for its enthusiasm, both from the visitors and the exhibitors. Quite an amount of equipment was sold, including larger packages to businessmen. Perhaps this fact alone answers the question — unfortunately not. For the people who were at the exhibition were part of a special group of people and not representative of businessmen in general.

Sociologists in various fields have known for some time that when a new product is introduced to a market, the people in that market who buy do so according to a set pattern of behaviour. This pattern identifies behaviour ranging from those who are keen on all new things and will try and buy anything new (innovators), through to those who will never buy until they are forced to by some legislation (laggards).

This behaviour was in evidence at the exhibition where the visiting businessmen were clearly "sold" before they arrived, and were ready to buy whichever equipment met their needs best. They so obviously met the definition of "innovators in the market" in that they were prepared to go much more than halfway towards the new technology; to cut through or learn the jargon; to discover for themselves the benefits of operating a micro-computer; and even to act as guinea pigs.

The Rair Black Box

Running out of steam?

However, the number of these innovators is really quite small (about 2½ per cent of the overall market), and does not represent the remainder of the market. The initial rush of the businessmen and enthusiastic hobbyists who are willing — even anxious — to buy, will keep the industry busy for just so long. After that, sales will become progressively harder to achieve, and the micro-industry will have to change its sales tactics.

The speed with which the new product will spread through to the remainder of the market has also been studied, and this speed depends upon "visibility" of the product both to the eye and to the "mind's eye".

Large items such as new combine harvesters, the latest range of car, are easily visible to the market and these are soon bought by the average buyers. A micro-computer however, is much less visible, for it will be hidden in the general office, and business associates, customers and suppliers may be quite unaware of its existence.

The Jargon Hunters

In some ways there are similarities between the micro market and Hi-Fi. This went through the same stage of extreme enthusiasm, preoccupation with response rates and specifications, ugly equipment with open chassis and wires everywhere, and an excess of jargon. As the market awoke to the benefits of Hi-Fi, the multinational companies stepped in, and catered for the average person in the market; notably making the product acceptable to the wife in the house by paying careful attention to the design of the equipment. They then exposed the product by *sophisticated marketing* so that it became not only visible to the average buyer, but appealed to him as a piece of furniture to grace any living-room. The biggest step came, however by *removing the excess of jargon* so that the average householder who has no knowledge of sound recording, electronics and the like, could understand enough about the controls, and be able to operate the equipment, without fear of damaging his records or of embarrassment before his friends.

Presented to him in a way that encapsulated the high technology within a simple array of controls, the equipment became visible to the "mind's eye" of the buyer. The result is now that few homes have either the bastard marriage of equipment so prevalent at one stage, or are without a Music-Centre altogether.

But here all the similarity between the two markets ends. Unlike record players, which scarcely anyone does not know how to use, or what benefits can be expected from their operation, few small businessmen know how to use a computer or what benefits can be gained. Neither can the few software houses producing application programs (from which the benefits of computer operation will be gained) be compared with the multi-million pound industry devoted to the production and sale of records.

At exhibitions, not only do the exhibitors display the latest technology but also their knowledge and understanding of the market through their representation. An analysis of the products and their representation at the PCW Exhibition will show whether the micro-computer industry is yet ready to sell to the small businessman.

Games

It was interesting that the main benefit shown by the exhibitors was the fun of pitting one's wits against the machine in various games, presumably because this is easily demonstrated. The systems analysis design, and the programming of a game, can be done by individuals in isolation and spare moments. Worthwhile domestic applications — for example, the effective control of hot water and central heating systems — require research, trials and special equipment, and being therefore more costly to produce, no such systems were demonstrated.

The Equinox 300

In sharp contrast to the games, no one offered the *benefits* likely to accrue from the operation of a microcomputer in a business. Some business systems were demonstrated, but they were either incomplete, contained bugs, or were demonstrated by someone with no knowledge. None could show a thorough understanding of the different needs of different kinds of businesses, and noone made an attempt to even discover whether the system being demonstrated was suitable for the visitor. Fundamental omissions, such as a lack of an audit trail, reduced the credibility of the demonstrations, whilst frequent frivolous and unhelpful comments regarding maintenance — a *vital* ingredient for any business — merely increased the overwhelming impression that micros are for fun or hobbyists.

There is *no* criticism of the exhibitors or the exhibition in these remarks either expressed or implied, for they could not seriously have expected visitors to the exhibition to have just "come in off the street" with no prior knowledge of computing. When the "converted" talk amongst themselves, the faults of systems can be ignored or looked upon as interesting challenges to be put right. Similarly the major benefits can be taken as read because having a knowledge and understanding of them is what being "converted" means.

However, it is not certain that even the enthusiastic businessmen who turned up at this exhibition understood all the benefits and problems of various applications, and even though the exhibitors had gone to the trouble of listing them to jog their memories; (pay-roll, sales ledger, purchase ledger, stock control) this will not have helped very much. A simple demonstration of a "stock control" system may show the enlightened what can be done, the speed of response etc., and leave him with a greater knowledge of the equipment — provided he is willing to work at its understanding. But to the sceptical businessman the "stock control" systems proved only to be "balances only" record systems, and this demonstrated a lack of understanding of both stock records and stock control.

Clouds on the horizon

One wonders how long it will be before the multinational office equipment companies move into micro-computers. When they do they will undoubtedly bring with them decades of experience of selling business systems to first time users, and will have all the resources necessary to produce glossy training books, instructors, salesmen, and catalogues complete with application programs to suit the different businesses they are likely to meet. Their main aim will be at those businessmen who are not now converted to computers but who can recognize the benefits of computerization once they are *clearly spelt out and presented in an easily digestible form.* These manufacturers will neither wait for the businessman to come to them nor will they take the benefits of computerization as read. Their strategy will be to actively bridge the gap by taking the benefits to the businessmen in their own language.

There are other factors to do with the products themselves which can hinder the development of the market. If micros are to be used in a technical role:—management services; production control etc.; then the users must both be capable and have time to put their problems on to the machine. More important however, companies who have invested in this way usually want the fact hidden, not released for public consumption. Thus these technical applications will spread only slowly through the market.

Alternatively, systems designed to end administrative drudgery are aimed at the very centre of a business' paper flow where they are seldom able to be introduced piecemeal, such as other pieces of office equipment — dictaphones for example, in order that their performance can be assessed before making a complete change. Instead the user is usually obliged to make the change, often irreversible, at one time, and he has to be pretty sure that its impact will be beneficial. That horrifying stories exist about abortive implementation efforts only heightens the expectation that a great deal of personal effort and perhaps sacrifice will be needed by the user. If the benefits at the end of all this are at all doubtful, only the most determined will go ahead.

Summary

1. There is no doubting the pull of micro-computers for the converted; the intellectual stimulus; the intoxication of being in the forefront of knowledge; the belief that they can reduce drudgery and therefore add to the quality of life for people currently enslaved in dreary tasks.

2. Useful for personal, home and business use, they are mainly demonstrated in the fun role for two basic reasons; games are easy to understand, to design and program and businesses are not; and games require only small cheap equipment.

3. The industry is at the present made up of many small enterprises in competition with each other and preoccupied with the establishment of industry standards.

4. Sales have been made to individuals and businessmen with sufficient knowledge to undertake their own programming tasks and for whom such tasks form a hobby, whether or not they are paid.

5. If the industry is to rely upon the cash of the hobbyist only then it will not go far, but must instead try to attract money and sales from business enterprises. The potential for sales to small businesses is enormous whether measured in money or numbers of units sold. This is clearly the area in which fortunes are going to be made. However, this is the very reason why the multinational companies will enter the scene and eventually there will only be room for one or two small businesses to remain with some special advantage that allows them to carve out a small niche of their own in the face of the fierce competition to come.

6. The people currently in the micro-computer industry should anticipate these changes in their market and should even now be making plans to sell to the main body of businessmen who are not the innovators.

7. The most successful companies will be those with experience of the special problems of selling business systems to small businessmen and first time users.

8. Their success will stand upon their ability to make the benefits of operating computers visible both to the eye and to the mind's eye i.e., not only must micros be seen to be working well, they must also be easy to operate by the average user with no knowledge of computing.

However, the future is not necessarily going to be easy even when the marketing homework has been done and the products are available with all the required back-up. At the start of this article I raised a second question concerning the businessman's ability to use computers. There is evidence to doubt this — but that is another story.

PCW *We invite comments from the small business computer industry.*

PCW

TIDY IT UP

TRS-80 Level II Renumbering Routine

I. Addinsell

Introduction

This program is written in Z80 assembler and is run using the 'SYSTEM' command. It occupies just under ½ K bytes and should be located at the top of Ram to give as much space as possible for the 'BASIC' programs. The area to be occupied by the program should be protected by 'MEMORY SIZE' set to 20039 in the case of a 4 K machine or 32269 in the case of a 16 K machine.

It will renumber both the lines themselves *and* any references to them except in 'REM' statements. This includes occurrences in multiple statement lines and such intricacies as: ON X GOTO 50, 60, 70.

In the form as published the lines will be numbered from 10 in steps of 10. This can be changed to suit very easily.

Actually getting this program into your TRS-80 can be done in a number of ways.

1. If you have the 'EDITOR/ASSEMBLER' then it is a simple matter of typing in the source and producing a tape of the object code.
2. If you have 'TBUG' then the process becomes slightly more tedious in that you have to type in the object code and store that on tape. Those with 'TBUG' will no doubt be familiar with the technique.
3. I developed this on a '4 K' machine which cannot run the 'EDITOR ASSEMBLER', neither do I have 'TBUG' so I write machine code programs using a monitor written in 'BASIC' utilising 'PEEK' 'POKE' and 'USR'.

If you have a 16 K machine then obviously you will want to relocate the program at the top of 'RAM'. To do this references to absolute addresses, i.e. data areas, stack and absolute jumps will have to be adjusted.

The program is executed by typing 'SYSTEM' and 120040 on a 4 K machine or 132270 on a 16 K machine, assuming you have relocated the program.

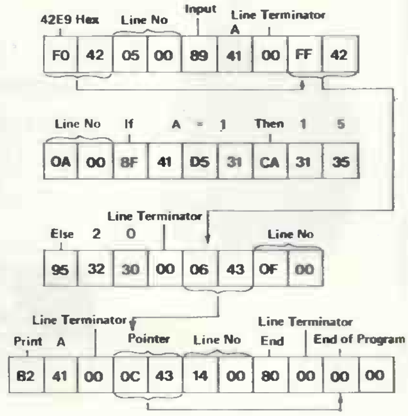
How it Works

First one has to understand how 'LEVEL II' software stores the text that makes up the 'BASIC' program. This sort of information is not at present available from 'TANDY' so I have had to work it out myself by trial and error. The format is . . . . .

The program text always starts at 42E9 Hex. The first two bytes are the address of the start of the next line, the next two bytes are the line number in binary format. From there onwards is the text until 0 Hex is reached, this is the line terminator. The next two bytes are the address of the next line and so on until the end of the program which is signified by the two bytes which would point to the next line in fact being zero. The only other detail which needs explaining is that the 'RESERVED WORDS' are stored as single bytes in the range 80 Hex to FF Hex. The software converts them back to strings when displaying them on the screen.

A sample program would be stored as follows:

```
 5  INPUT A
10  IF A=1 THEN 15 ELSE 20
15  PRINT A
20  END
```



So the renumbering program has to search each line for the compressed codes for GOTO, GOSUB etc, and get the string representing the line number which follows. There may of course not be a number in the case of 'THEN' or 'ELSE'. e.g. '10 IF X=1 THEN PRINT "X=1" '. This numeric string must then be converted into a 16 bit binary number and stored in a data area along with a pointer to its location and, for reasons to be explained later, the room for it to be returned in including any spaces. The program goes through the whole program storing this data. Each valid number uses five bytes of data area.

The second pass then takes the first line number and searches through the data area for any references to it. If it finds one the new line number is converted to a string and inserted back in the program and so on down the data area. When the end of the data is reached the line itself is updated and it moves on to the next line and repeats the exercise until the last line is reached.

Error Conditions

There are a couple of error conditions which can occur. These are, I should imagine, more likely to occur on a 4 K machine.

1. If the program is written with every space deleted to save space, particularly likely with 4 K systems, then if the new number to be inserted is greater than the old one then it cannot be inserted and a message to that effect is printed on the screen along with the number which could not be inserted. I have not, for reasons of size, calculated which line the number should belong on so if there are several such error messages you will have to work out where they belong. The locations affected will contain stars to draw your attention to them. In the unlikely case where more than ten errors occur then the program will stop to allow you time to note them down and will continue when any key is pressed. To avoid this error simply insert a leading and/or trailing space round numbers which might become longer after numbering.

2. This error is not flagged by the program, again for space reasons, and is unlikely to occur unless the program occupies nearly all the memory available: What could happen is that the data area, which starts at the bottom of the machine code routine and runs down memory, could overwrite the end of the 'BASIC' program. If you think this might happen store your program on cassette temporarily before renumbering and if it does happen then reload the program, delete a couple of lines, renumber it, and type the lines back in.

3. Finally, jumps to undefined lines will be left with stars in them. No message is output.

If you want to change the numbering 'LNINC' gives the line spacing and 'LNSTRT' gives the number of the first line. These can be changed freely. The only caution is there is no check that the line number has gone over 65535 and round to zero again. This will, needless to say, give the interpreter hiccoughs.

Any comments concerning the program itself or on the 'TRS-80' would be read with interest. I make no claims about the style of the program. It is the first program I have written in 'Z80' assembler, also if more was known about 'LEVEL II' software the program could no doubt be shortened by using sub-routines in the ROM.

I am currently disassembling the ROM and gradually finding out how for example the printer is addressed by the commands 'LLIST' and 'LPRINT' so that 'TANDY'S' printer need not be used.

```
ADDR OBJECT   LABEL   OPCD OPERAND         COMMENT

                ;       TRS-80 RENUMBERING ROUTINE
                ;       **************************
                        LIST X
                        ORG  4E48H         ;PROGRAM START ADDRESS
>0091           GOSUB   EQU  91H           ;)
>008D           GOTO    EQU  8DH           ;) COMPRESSED CODES USED BY TRS80 SOFTWARE
>00CA           THEN    EQU  0CAH          ;)
>0095           ELSE    EQU  95H           ;)
>0020           SPACE   EQU  ' '
>002C           COMMA   EQU  ','
>002A           STAR    EQU  '*'
>00FF           COMMA1  EQU  0FFH          ; TEMPORARY CODE USED FOR COMMA IN "ON  GOTO"
>1A19           BASIC   EQU  1A19H         ;  RE-ENTRY ADDRESS TO BASIC
>4020           CURSOR  EQU  4020H         ;  CURRENT CURSOR POSITION ( TRS80'S )
>3C00           SCRNAD  EQU  3C00H         ;  START OF SCREEN
>42E9           START   EQU  42E9H         ;  START OF BASIC PROGRAM TEXT
>09C2           GETLN   EQU  09C2H         ;GETS PTR TO NEXT LINE IN DE & LINE NO. IN BC
>0049           KBD     EQU  0049H         ;RETURN WITH VALID KEY
>057C           ERASEOF EQU  057CH         ;ERASE FROM HL TO END OF FRAME
                ;
                ;
>000A           LNSTRT  EQU  0AH           ;LINE NO. TO START FROM
>000A           LNINC   EQU  0AH           ;LINE SPACING
                ;
                ;
                ;
                ;
                ;
                ; ******************** INITIALIZE ********************
                ;
4E48            BEGIN:
4E48 FD21474E           LD   IY,BEGIN-1    ;SET DATA AREA FROM START OF PROGRAM DOWNWARDS
4E4C 310050             LD   SP,5000H      ;SET UP NEW STACK
4E4F 210A00             LD   HL,LNSTRT     ;**** INITIALIZE LINE NO. START TO 10
4E52 22DA4F             LD   (LNCOUNT),HL
4E55 CD824F             CALL CLS
                ; ******************** 1ST PASS ******************
4E58 21E942             LD   HL,START      ; START AT BEGINNING OF FILE
4E5B            NEXTLN:
4E5B CDBC4F             CALL GETLINE
4E5E CAF84E             JP   Z,LAST       ;CHECK FOR END OF PROGRAM
4E61 D5                 PUSH DE            ; SAVE POINTER TO NEXT LINE
4E62 EB                 EX   DE,HL         ;)
4E63 ED52               SBC  HL,DE         ;) FIND NO. OF BYTES IN LINE
4E65 E5                 PUSH HL            ; SET COUNTER FOR LENGTH OF LINE
4E66 C1                 POP  BC
4E67 C5                 PUSH BC            ; SAVE NO. BYTES IN LINE
4E68 3E8D               LD   A,GOTO        ;) SEARCH FOR A "GOTO"
4E6A D5                 PUSH DE
4E6B            AGAIN:
4E6B E1                 POP  HL            ;) GET START ADD OF TEXT IN LINE
4E6C C1                 POP  BC            ; )
4E6D C5                 PUSH BC            ; )START AT BEGINNING OF LINE
4E6E E5                 PUSH HL
                CARYON:                    ;CONTINUE THROUGH LINE FROM WHERE YOU LEFT OFF
4E6F EDB1               CPIR               ;SEARCH LINE
4E71 281B               JR   Z,FOUND-$     ;FOUND A MATCH
4E73 FE8D               CP   GOTO          ;HAVE I JUST DONE GOTO ?
4E75 2813               JR   Z,ADD4-$
4E77 FE91               CP   GOSUB         ;HAVE I JUST DONE GOSUB ?
4E79 280F               JR   Z,ADD4-$
4E7B FE95               CP   ELSE          ;HAVE I JUST DONE ELSE ?
4E7D 2809               JR   Z,ADD35-$
4E7F FECA               CP   THEN          ;HAVE I JUST DONE THEN ?
4E81 2805               JR   Z,ADD35-$
4E83 E1                 POP  HL            ;)
4E84 E1                 POP  HL            ;) GET POINTER TO NEXT LINE
4E85 E1                 POP  HL            ;)
4E86 18D3               JR   NEXTLN-$
                ADD35:                     ;( GOTO+4=GOSUB    )
4E88 C631               ADD  A,31H         ;( GOSUB+4=ELSE    )
                ADD4:                      ;( ELSE+35H=THEN   )
4E8A C604               ADD  A,4           ;( THEN+35H=COMMA1 )
4E8C 18DD               JR   AGAIN-$
4E8E            FOUND:
```

```
4E8E F5                     PUSH AF          ; MAKE A NOTE OF GOTO/GOSUB/ETC
4E8F C5                     PUSH BC          ; SAVE LOC OF GOTO/GOSUB IN LINE
4E90 0E00                   LD   C,0         ;CLEAR FOR COUNTING.B IS 0 -LINE LENGTH<=255
4E92 E5                     PUSH HL          ;SAVE ADDR OF POSN IN LINE TO CONTINUE FROM
4E93 FD7500                 LD   (IY+0),L    ;)
4E96 FD74FF                 LD   (IY-1),H    ;)SAVE ADDR OF GOTO/GOSUB'S NUMERIC STRING
4E99 DD21DD4F               LD   IX,TEMP     ;START ADDR OF TEMP STORAGE FOR NUMERIC STRING
4E9D FEFF                   CP   COMMA1
4E9F 2004                   JR   NZ,MORE-$   ;CHECK IF TESTING FOR COMMA IN "ON GOTO" ETC.
4EA1 2B                     DEC  HL          ;YES CONVERT IT BACK TO A COMMA
4EA2 362C                   LD   (HL),COMMA
4EA4 23                     INC  HL
4EA5               MORE:
4EA5 7E                     LD   A,(HL)      ; GET CHAR
4EA6 FE20                   CP   SPACE       ; IS IT A SPACE   ?
4EA8 2848                   JR   Z,SPCCNT-$  ; GO ADD 1 TO SPACE COUNT
4EAA FE2C                   CP   COMMA
4EAC 2002                   JR   NZ,NUMCHK-$ ;IF A COMMA THEN CONVERT IT TO A SPECIAL CODE
4EAE 36FF                   LD   (HL),COMMA1 ;FOR TESTING ON SUBSEQUENT PASS
4EB0               NUMCHK:
4EB0 FE30                   CP   30H         ;)
4EB2 FABA4E                 JP   M,ENDNUM    ;)  IS IT NUMERIC ?
4EB5 FE3A                   CP   3AH         ;)
4EB7 FAEA4E                 JP   M,NUM       ;)
4EBA               ENDNUM:
4EBA 78                     LD   A,B
4EBB B7                     OR   A           ;GET NUMBER OF DIGITS IN NO.
4EBC 2827                   JR   Z,NOTNUM-$  ; IF NUMERAL CONTAINS NO DIGITS
4EBE 81                     ADD  A,C         ;ADD NO. OF SPACES
4EBF FD77FE                 LD   (IY-2),A    ;NOTE ROOM AVAILABLE FOR NUMBER TO BE RETURNED
4EC2 210000                 LD   HL,0
4EC5 DD21DD4F               LD   IX,TEMP
                   LOOP:                     ;  )
4EC9 29                     ADD  HL,HL       ;  )
4ECA E5                     PUSH HL          ;  )
4ECB 29                     ADD  HL,HL       ; )
4ECC 29                     ADD  HL,HL       ; )
4ECD D1                     POP  DE          ; )
4ECE 19                     ADD  HL,DE       ; )CONVERT NUMERIC STRING TO 16 BIT BINARY NO.
4ECF DD7E00                 LD   A,(IX+0)    ; )
4ED2 D630                   SUB  30H         ; )
4ED4 5F                     LD   E,A         ; )
4ED5 1600                   LD   D,0         ; )
4ED7 19                     ADD  HL,DE       ; )
4ED8 DD23                   INC  IX          ; )
4EDA 10ED                   DJNZ LOOP-$      ; )
4EDC FD75FD                 LD   (IY-3),L    ;)
4EDF FD74FC                 LD   (IY-4),H    ;) STORE CONVERTED NO.
4EE2 CDC24F                 CALL DATAMV      ;MOVE DOWN STORAGE AREA 5 BYTES
4EE5               NOTNUM:
4EE5 E1                     POP  HL          ; GET ADD TO CONTINUE FROM IN LINE
4EE6 C1                     POP  BC          ; GET DISPLACEMENT IN LINE
4EE7 F1                     POP  AF          ; GET GOTO/GOSUB ETC TO CONTINUE TESTING
4EE8 1885                   JR   CARYON-$
4EEA               NUM:
4EEA 362A                   LD   (HL),STAR   ;PUT IN AS ERROR MARKER
4EEC DD7700                 LD   (IX+0),A    ;CREATE NUMERIC STRING FOR PROCESSING
4EEF 04                     INC  B           ;KEEP COUNT OF LENGTH OF STRING
4EF0 DD23                   INC  IX
4EF2               SPCE:
4EF2 23                     INC  HL
4EF3 1880                   JR   MORE-$
4EF5               SPCCNT:
4EF5 0C                     INC  C           ;KEEP COUNT OF SPACES
4EF6 18FA                   JR   SPCE-$
                   ; ******************2ND PASS ***********************
4EF8               LAST:
4EF8 FDE5                   PUSH IY          ;PUT END OF DATA AREA IN IX
4EFA DDE1                   POP  IX
4EFC 21E942                 LD   HL,START
4EFF               PASS2:
4EFF CDBC4F                 CALL GETLINE     ;GET ADD OF NEXT LINE IN DE REG
4F02 FD21474E               LD   IY,BEGIN-1  ;PUT IY AT START OF DATA AREA
4F06 CA191A                 JP   Z,BASIC     ;FINISHED
4F09 D5                     PUSH DE          ;SAVE ADDR OF NEXT LINE
4F0A E5                     PUSH HL          ;SAVE POSITION IN LINE
4F0B               NEXBLK:
4F0B FDE5                   PUSH IY          ;FIND OUT IF AT END OF DATA AREA
4F0D E1                     POP  HL
4F0E DDE5                   PUSH IX
4F10 D1                     POP  DE
4F11 B7                     OR   A
4F12 ED52                   SBC  HL,DE
4F14 ED5BDA4F               LD   DE,(LNCOUNT) ;GET NEW LINE NO.
4F18 2863                   JR   Z,DATEND-$
4F1A C5                     PUSH BC          ;SAVE LINE NO.
4F1B FD6EFD                 LD   L,(IY-3)    ;GET STORED LINE NO.
4F1E FD66FC                 LD   H,(IY-4)
4F21 ED42                   SBC  HL,BC       ;FIND OUT IF IT IS THE CURRENT LINE
4F23 2052                   JR   NZ,NEXDAT-$
4F25 19                     ADD  HL,DE       ;GET NEW LINE NO. TO BE CONVERTED ( HL=0 )
4F26 DDE5                   PUSH IX          ;SAVE END OF DATA AREA
4F28 DD21D04F               LD   IX,P10TAB
4F2C FDE5                   PUSH IY          ;SAVE POSITION IN DATA AREA
4F2E FD21DD4F               LD   IY,TEMP
4F32               LOOP0:
4F32 3E2F                   LD   A,2FH       ;SET UP 'A' REG FOR AN ASCII CHAR
4F34 DD4E00                 LD   C,(IX+0)    ;)GET POWER OF TEN FROM TABLE
4F37 DD4601                 LD   B,(IX+1)    ;)
4F3A               LOOP1:
4F3A C601                   ADD  A,1         ;INC NO. & CLEAR CARRY
4F3C ED42                   SBC  HL,BC
4F3E F23A4F                 JP   P,LOOP1     ;DO AGAIN IF RESULT NOT NEGATIVE
4F41 09                     ADD  HL,BC       ; RE-ADJUST BACK TO POSITIVE
4F42 FD7700                 LD   (IY+0),A    ;PUT ASCII DIGIT IN TEMP
4F45 FD23                   INC  IY          ;MOVE UP TEMP
4F47 DD23                   INC  IX          ; )MOVE UP POWER OF TEN TABLE
4F49 DD23                   INC  IX          ; )
4F4B 0D                     DEC  C           ;WAS THAT THE LAST POWER OF TEN (WAS IT 1)
4F4C 20E4                   JR   NZ,LOOP0-$
4F4E 010500                 LD   BC,5        ;)
4F51 21DD4F                 LD   HL,TEMP     ;)
4F54 3E30                   LD   A,30H       ;) IGNORE LEADING ZEROS
                   FNDDIG:                   ;)
4F56 BE                     CP   (HL)        ;)
4F57 FA5E4F                 JP   M,DIGT      ;)
4F5A 0B                     DEC  BC          ;) GET COUNT OF VALID DIGITS IN NUMBER
4F5B 23                     INC  HL          ;)
4F5C 18F8                   JR   FNDDIG-$    ;)
4F5E               DIGT:
4F5E FDE1                   POP  IY
4F60 FD7EFE                 LD   A,(IY-2)    ;SIZE OF NUM TO BE INSERTED IN C (BC<=5)
4F63 B9                     CP   C
4F64 FA8D4F                 JP   M,ERROR     ;IF NOT ENOUGH SPACE
4F67 2804                   JR   Z,JUSTRM-$
4F69 03                     INC  BC
4F6A 2B                     DEC  HL          ;IF ROOM TO SPARE PUT IN A LEADING SPACE
4F6B 3620                   LD   (HL),SPACE
4F6D               JUSTRM:
4F6D FD5E00                 LD   E,(IY+0)    ;GET POSITION IN LINE TO PUT NO.
4F70 FD56FF                 LD   D,(IY-1)
4F73               ERRRET:
4F73 EDB0                   LDIR             ;MOVE NEW NUMBER
4F75 DDE1                   POP  IX
4F77               NEXDAT:
4F77 C1                     POP  BC          GET LINE NO. BACK FOR NEXT TEST
4F78 CDC24F                 CALL DATAMV      ;MOVE DOWN DATA AREA
4F7B 188E                   JR   NEXBLK-$
                   DATEND:                   ;REACHED END OF DATA AREA
4F7D E1                     POP  HL          ;)WRITE NEW LINE NO. OUT TO LINE
4F7E 2B                     DEC  HL          ;)
4F7F 72                     LD   (HL),D      ;)
4F80 2B                     DEC  HL          ;)
4F81 73                     LD   (HL),E      ;)
4F82 210A00                 LD   HL,LNINC    ;****)INCREMENT LINE POINTER TO NEXT LINE
4F85 09                     ADD  HL,BC       ;      )
4F86 22DA4F                 LD   (LNCOUNT),HL ;     )
4F89 E1                     POP  HL          ;GET POINTER TO NEXT LINE
4F8A C3FF4E                 JP   PASS2
4F8D               ERROR:
4F8D E5                     PUSH HL
4F8E C5                     PUSH BC
4F8F ED5B2040               LD   DE,(CURSOR) ;FIND POSITION TO PRINT AT
```

```
4F93 214000          LD   HL,40H         ;SET TAB FOR NEXT ERROR MESSAGE
4F96 19              ADD  HL,DE
4F97 3E3F            LD   A,3FH          ;STOP BEFORE BOTTOM OF SCREEN
4F99 BC              CP   H
4F9A 2006            JR   NZ,ONSCRN-$
4F9C CD4900          CALL KBD            ;IF SCREEN FULL WAIT FOR KEY
4F9F CD824F          CALL CLS
4FA2         ONSCRN:
4FA2 E5              PUSH HL             ;SET DE TO POSITION ON SCREEN TO PRINT
4FA3 0E08            LD   C,8            ;SIZE OF ERROR MESSAGE (B REG IS 0)
4FA5 222040          LD   (CURSOR),HL    ;SET CURSOR TO NEXT PRINT POSN.
4FA8 D1              POP  DE             ;GET POSITION TO PRINT AT.
4FA9 21C84F          LD   HL,ERRMES
4FAC EDB0            LDIR                ;OUTPUT ERROR MESSAGE
4FAE C1              POP  BC
4FAF E1              POP  HL
4FB0 18C1            JR   ERRRET-$
              ; ************* SUBROUTINES ********************
              ;
             CLS:                        ; ** CLEAR SCREEN **
4FB2 21003C          LD   HL,SCRNAD      ;SET HL TO TOP OF SCREEN
4FB5 222040          LD   (CURSOR),HL    ;RESET CURSOR
4FB8 CD7C05          CALL ERASEOF        ;ERASE SCREEN
4FBB C9              RET
             GETLINE:                    ; ** GET POINTER TO NEXT LINE **
4FBC CDC209          CALL GETLN          ;          & GET LINE NO.
4FBF 7A              LD   A,D
4FC0 B3              OR   E              ;SET FLAGS TO TEST IF AT END OF PROGRAM
4FC1 C9              RET
             DATAMV:                     ; ** MOVE DOWN DATA AREA BY 5 BYTES **
4FC2 11FBFF          LD   DE,-5
4FC5 FD19            ADD  IY,DE
4FC7 C9              RET
              ;
              ;
              ;
              ;
              ;
              ; ********** DATA & SCRATCH PAD ***********
              ;
              ;
4FC8 4F4D4954 ERRMES: DEFM 'OMITTED:'
             P10TAB:                     ;POWER OF TEN LOOKUP TABLE
4FD0 1027            DEFW 10000
4FD2 E803            DEFW 1000
4FD4 6400            DEFW 100
4FD6 0A00            DEFW 10
4FD8 0100            DEFW 1
4FDA         LNCOUNT DEFS 2              ;CURRENT LINE COUNTER
4FDC                 DEFS 1              ;ROOM FOR LEADING SPACE IF 5 VALID DIGITS
4FDD         TEMP    DEFS 5              ;SCRATCH AREA FOR ASCII/BIN & VICE VERSA CONV
4FE2                 END

TOTAL ASSEMBLER ERRORS =    0
```

PCW We apologise for the late appearance of the April issue and the delay in this issue.
A printing dispute held up distribution **PCW**

PARKINSON'S REVAS

(LISTING TO FOLLOW IN FUTURE ISSUES)

D. W. Parkinson

As the name implies, a reverse assembler, or disassembler, performs the opposite function of an assembler, — it translates object code back to a source listing in mnemonic form. However, an important point must be remembered from the outset. An original source listing (assuming the programmer has some common sense) is full of comments and explanatory labels so that the flow of the program can be followed easily by anyone looking at it. These aids are all lost during the assembly process as they contribute nothing to the final code output. Thus, translating in the reverse direction. REVAS cannot insert any of these items which have no representation in the object code. It is, however, possible to arrange for the program to generate a form of label and to provide a small amount of additional information to aid the deciphering of the listing.

What use is a reverse assembler?

The reverse assembler in its basic form is a simple and small program (1½ Kbytes), and for those with no assembler (like me when I wrote REVAS) it provides an admirable way of checking the coding of a program, that the codes are correct and that relative jumps end up where they were intended. The other and perhaps more obvious use is in the disassembling of other people's software. Commercial software usually comes with no source listing and has either been written for a specific machine (which you don't happen to have!) or, for commercial reasons, has been written to be as general as possible. By reverse assembling the object code you can discover how the program works, and, with luck and/or several hours work, modify it to suit yourself. (Often you will find that by the time you have worked your way through six or more levels of subroutine calls you'll have forgotten where you started, and what you thought was happening there).

It can also be used to relocate programs by reverse assembling the object code to produce a source version, and then re-assembling the source with a new origin, although to do this successfully can require a certain amount of work. The reason for this can be seen by considering the instruction LD HL,NNNN. NNNN is a sixteen-bit operand which could be program relocatable (a table or buffer address), or absolute (a constant in an arithmetic routine). The reverse assembler has no information on which to base a decision and so must, in general, stick to a single convention. By assuming all sixteen-bit operands are program relocatable it is unnecessary to identify the operation (JP, CALL, LD direct and indirect) the operand is associated with, and leaves the constants to be identified and amended in the source listing before re-assembly.

Using REVAS

An example of the use of REVAS can be seen in its own listing. I was dissapointed that the Banahan assembler (previous PCW's) did not use the standard mnemonics (which I like and find easy to use) and having already decided that I hadn't the energy to write an assembler without an assembler to assemble it (!) I ended up buying a commercial operating system/assembler. The assembler used the Intel approach for hex-

HIGH NIBBLE

LOW NIBBLE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	DJNZ	JR NZ	JR NC	LD B,B	LD D,B	LD H,B	LD (HL),B	ADD B	SUB B	AND B	OR B	RET NZ	RET NC	RET PO	RET P
1	LD BC,nn	LD DE,nn	LD HL,nn	LD SP,nn	LD B,C	LD D,C	LD H,C	LD (HL), C	ADD C	SUB C	AND C	OR C	POP BC	POP DE	POP HL	POP AF
2	LD (BC). A	LD (DE). A	LD (nn),HL	LD (nn),A	LD B,D	LD D,D	LD H,D	LD (HL), D	ADD D	SUB D	AND D	OR D	JP NZ	JP NC	JP PO	JP POS
3	INC BC	INC DE	INC HL	INC SP	LD B,E	LD D,E	LD H,E	LD (HL),E	ADD E	SUB E	AND E	OR E	JP	OUT	EX HL,(SP)	DI
4	INC B	INC D	INC H	INC (HL)	LD B,H	LD D,H	LD H,H	LD (HL),H	ADD H	SUB H	AND H	OR H	CALL NZ	CALL NC	CALL PO	CALL P
5	DEC B	DEC D	DEC H	DEC (HL)	LD B,L	LD D,L	LD H,L	LD (HL),L	ADD L	SUB L	AND L	OR L	PUSH BC	PUSH DE	PUSH HL	PUSH AF
6	LD B,n	LD D,n	LD H,n	LD (HL),n	LD B,(HL)	LD D,(HL)	LD H,(HL)	HALT	ADD (HL)	SUB (HL)	AND (HL)	DR (HL)	ADD A,n	SUB A,n	AND A,n	OR A,n
7	RLCA	RLA	DAA	SCF	LD B,A	LD D,A	LD H,A	LD (HL), A	ADD A	SUB A	AND A	DR A	RST O	RST 16	RST 32	RST 48
8	EX AF,AF'	JR	JR Z	JR C	LD C,B	LD E,B	LD L,B	LD A,B	ADC B	SBC B	XOR B	CP B	RET Z	RET C	RET PE	RET M
9	ADD HL,BC	ADD HL,DE	ADD HG HL	ADD HL,SP	LD C,C	LD E,C	LD L,C	LD A,C	ADC C	SBC C	XOR C	CP C	RET	EXX	JP (HL)	LD SP,HL
A	LD A,(BC)	LD A,(DE)	LD HL,(nn)	LD A,(nn)	LD C,D	LD E,D	LD L,D	LD A,D	ADC D	SBC D	XOR D	CP D	JP Z	JP C	JP PE	JP M
B	DEC BC	DEC DE	DEC HL	DEC SP	LD C,E	LD E,E	LD L,E	LD A,E	ADC E	SBC E	XOR E	CP E	ROT/SET BIT/RES	IN	EX DE,HL	EI
C	INC C	INC E	INC L	INC A	LD C,H	LD E,H	LD L,H	LD A,H	ADC H	SBC H	XOR H	CP H	CALL Z	CALL C	CALL PE	CALL M
D	DEC C	DEC E	DEC L	DEC A	LD C,L	LD E,L	LD L,L	LD A,L	ADC L	SBC L	XOR L	CP L	CALL	IX	Extended	IY
E	LD C,n	LD E,n	LD L,n	LD A,n	LD C,(HL)	LD E,(HL)	LD L,(HL)	LD A,(HL)	ADC (HL)	SBC (HL)	XOR (HL)	CP (HL)	ADC A,n	SBC A,n	XOR A,n	CP A,n
F	RRCA	RRA	CPL	CCF	LD C,A	LD E,A	LD L,A	LD A,A	ADC A	SBC A	XDR A	CP A	RST 8	RST 24	RST 40	RST 56

Figure 1.

adecimal constants. (All constants must begin with a numeric, and if hexadecimal end with an H). I far prefer the Motorola convention of a leading $ for hex. To my mind $D, $A or $FO is far more readable than ODH, OAH or OFOH, the leading $ clearly identifying hex rather than leaving the eye reading backwards and forwards while the mind decides whether it is decimal, hex, hex but you forgot the leading 0, or a label. It took about half an hour's study of the 'REVASed' listing before I was able to make the small patch necessary to suit my preferences.

Thus in the appended listing all hexadecimal numbers begin with a $, and as a result the program counter representation has been changed to * rather than the $ that it was.

The Way it Works

The listing is well commented and should be easy to follow, but to see the reason behind the approach taken it is necessary to draw up three tables in the form of Figure 1. In the first enter the operations appropriate to the first (or only) bytes of all the opcodes. (e.g. box C3 should contain JP, box 24 INC H, etc). The second should hold all the operations that could follow the extended operator ED, and the third those that could follow CB. There is no need to produce tables for the index operators DD and FD as the following opcodes are identical to those for register pair HL, and so DD and FD only set a flag to indicate use of the appropriate index register in place of HL.

Having got a byte to decode, the program is faced with deciding which of 256 possibilities it is. Decoding directly by a large look up table is possible, but very wasteful of memory. However looking at the first table of opcodes two things immediately stand out:

Codes in the range $01XXXXXX_2$ are all LDs (with the exception of HALT).

Codes in the range $10XXXXXX_2$ are all 8-bit arithmetic/logical.

In each case the registers and/or operations are easily identified from the two subgroups, bits 5,4,3 and bits 2,1,0.

Having checked for these two possibilities, we are left with 128 alternatives. Further inspection of the remaining opcodes reveals that they subdivide nicely (on the whole) into groups of four, and decoding is done on this basis, the relevant bits being used as an index into a 32 entry transfer address table. Appropriate routines then decode the opcode fully, reading further bytes of code as necessary.

How to use it

REVAS was written as a self contained subroutine which could be used easily by anyone, being completely free of any operating system peculiarities. It is romable and requires two additional programs to run, both of which can be extremely simple. The first program is the control program that tells REVAS what to run on and where it is, the second is a subroutine to deal with the output from REVAS. As REVAS works through the object code, it fills an output buffer with the equivalent mnemonics and calls the output routine every time a line is complete. (On calling HL points to the start of the buffer). The output is of the form:

XXXX	XX XX XX XX	XXX	XX,XXX	;XXXX<CR><NULL>
Hex addr.	Opcodes (1-4 bytes)	Mnemonic	Operand if any.	Any ASCII equivalent of opcode.

The mnemonic is the standard Zilog mnemonic with the exception of OUTI which appears as OTI. Also, to suit my assembler arithmetic, operations involving the accumulator (ADD A, — etc) have the A, omitted, and all hex constants are identified by a leading $. With relative jumps the target address is computed and appears as the operand. To aid recognition of messages, tables etc., any ASCII equivalent(s) of the opcode(s) appear in the comment field of the line. An example of the use of this can be seen in the sample output which is in fact part of REVAS.

The simplest control program can be loaded by hand and is:

```
01 XX XX   LD     BC,$XXXX   XXXX = Address for listing
11 YY YY   LD     DE,$YYYY   YYYY = Start address
21 ZZ ZZ   LD     HL,$ZZZZ   ZZZZ = Last address
CD 00 F8   CALL   REVAS
C3 ?? ??   JP     MONITOR
```

The monitor's standard line output routine can be used to print the output.

A simple routine I use in my monitor is included as listing 3. This executes a command "What's". It is invoked by typing WXXXX, where XXXX is a hexadecimal address, and starts REVAS running from that address. After fifteen lines of output, which takes around ⅛ second at my VDU speed, it pauses until a key on the keyboard is pressed. The spacebar produces the next fifteen lines, any other key returns to the monitor. Nascom users wanting to do this will need to write a small routine to reduce the length of the output line so that it can fit conveniently onto their screen (48 characters long). The buffer contains room for a label, but as this is not used in this instance, the space for reserved for it can be ignored on output. (List the first 17 characters, skip 7, then list the remaining ones).

The Pointers

Included in REVAS's variables are several pointers (LABELP,LABELA, LINEA) which enable the control program to add labels to the listing.

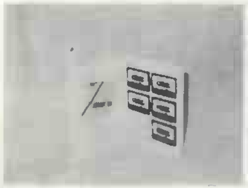
To do this a two pass approach is needed, one to locate all memory references and enter them into a symbol table, and the second to do the listing, adding labels as necessary. A control program which does this is REVASC which offers the following options:

Labelled listing	
Cross reference table	(Very useful if you want to modify a program)
Output to tape	
Source form only	(No addresses or opcodes output)
Squashed output	(Non-significant blanks removed)
Paginated listing	
Append line feed & nulls	(Tape output only)

The program is about 1 Kbyte long and, as you would expect, uses several system dependent routines.

LIBRARY 100

for the TRS-80



A complete library of 100 programmes for the TRS-80 Level II. The 5 cassettes are attractively packaged in a plastic binder which also holds the full documentation. All programmes were specifically written for the Library and, as a bonus, include a Tiny version of the Pilot language, to give your TRS-80 the added dimension of a new language. Less than 50p a programme !!

Finance: Present Value of a Future Sum—Simple Interest for Days— Future Value of a Present Sum—Amortization Schedule—Interest Rate: Compound Interest—Interest Rate: Installment Loan—Days Between Dates—Term of an Installment Loan—Present Value of a Series of Payments—Real Estate Capital Investment—Nominal and Effective Interest Rates—Internal Rate of Return—Future Value of Regular Deposits—Regular Deposits for Future Value—Depreciation Amount: Rate; Salvage Value; Schedule—Bond Present Value—Bond Yield to Maturity—Sale—Cost—Margin—Day of the Week—Moving Ad.

Education: Multiplication & Division—Add—Subtract—Fraction & Decimal—States & Capitals—States & Order of Entry—States & Abbreviation—Inventors & Inventions—World Capitals & Countries—Urban Areas & Population—Authors & Books—Presidents & Order—States & Largest City—Basenum.

Graphics: Left Right—Random Ad—Graphic—Blocks—Fireside— Snow—Step Ad—Step Ad 2—Launch—Ratrace—War Game—Weird—Herring—Blinker—Snoopy.

Home: Message Board—Expense Account—Nutrition—Mileage— Remember—Phone Codes—Night Check Off—Drunkometer—Perpetual Calendar—Babysitter—Calculator—Bartender—Christmas List—Vacation Check Off—Conversion.

Games: Speedy—Odd One—R. Roulette—Star Blazer—Search— Spyship—Tiger Shark—Jumble 2—Sting Ray—Stars—Sketch—Flipper—Scissors—Horse—Doomsday—Craps—Jumble 1—Mem. Quiz Letters—Mem. Quiz Numbers—Wheel of Fortune—Decision—Unjumble—Fifteen—Towers—Life—Star Trek—Race Track—Count—Roachrace—Gypsy.

£46.95 (inc)

A. J. HARDING

28 Collington Ave., Bexhill, E. Sussex. (0424) 220391.

TOR TOR

ITT 2020

The Apple of Your Eye!

Micro Computer

Priced from £827 to £1278 (exc. VAT)

Unbeatable features:

- 8 – 48K RAM
- High resolution colour graphics.
- ITT floppy disc drives
- ITT Serial printers

plus

Full business systems software support and customised packages — available NOW.

Contact the exclusive distributors:

Tor Business Systems Ltd., 83 Timberbank, Vigo Village, Meopham, KENT. Tel: 0732 822956 *and* 01-828 7692

Limited number of dealerships available.

SMALL SYSTEMS ENGINEERING LTD.
(Incorporating R. Bailey Associates)
62 New Cavendish Street, London W1M 7LD.
Telephone: 01-637 0777, Telex: 8813085 Abacus

A PRINCIPAL SUPPLIER OF PET MEMORY BOARDS AND INTERFACES IN EUROPE

PET MEMORY BOARDS
Sole official U.K. Agent for PME memory boards. We provide approved technical back-up, up-grade and service facilities for these boards in the U.K. Internally mounting memory boards available in 2 configurations; – 24K – £328 – 32K – £432.

IEEE-488/RS232C SERIAL INTERFACE
- Full IEEE address decoding, R232C or 20MA Loop output,
- Switch selectable Baud Rate, Crystal controlled Baud Rate timing,
- Boxed units complete with connectors, full operating instructions and sample programs supplied,
- Serial Interface B, input & output **£186** Serial Interface A, output only **£106**,
- Lower Case Printing.

CUSTOM INTERFACE DESIGN
Interfaces designed for special applications. Interfaces supplied so far include Analog Input/Output, XY plotter, stepper motor control.

PET INTELLIGENT TERMINAL SOFTWARE PACKAGE
A software package which, in conjunction with an Interface B enables the PET to operate as an intelligent terminal. The software implements full IBM or DEC protocols **£100.**

TV/VIDEO MONITOR INTERFACES
- Video Monitor output only £29.50
- Video and UHF output (plugs into aerial sockets of domestic TV) £35.

IEEE-488 TELEX PUNCH AND INTERFACE
- 50 chars/sec. Telex punch. Price to be announced.

TERMS: All prices Ex. VAT. All orders C.W.O. Cheques should be made payable to:
SMALL SYSTEMS ENGINEERING LTD.
Orders should include £2.50 p&p per unit. All goods supplied under 90 days warranty.

OHIO SCIENTIFIC
Small Systems Engineering is an official distributor for OHIO SCIENTIFIC products including:

CHALLENGER I (available March 79)

– Superboard II 4K computer on a board	£ 298.00
– CIP 4K Superboard in case and power supply	£ 415.00
– CIP MF 16K CIP with mini-floppy and OS-65D V3.0	£1,235.00

CHALLENGER II

– C2-4P "professional portable" computer	£ 620.00
– C2-4P MF 20K C2-4P with mini-floppy and OS-65D V3	£1,595.00
– C2-8P 4K mainframe class personal computer	£ 825.00
– C2-8P DF 32K dual 8" floppy personal/business system	£2,670.00

The **SUPERBOARD II** includes 8K BASIC in ROM, up to 8K of RAM on board, full keyboard, cassette interface and video display interface.

The **CHALLENGER C2-4P M4** includes 20K of RAM, 1 minifloppy drive, full keyboard and TV monitor, software supplied.

Includes:
- Business disk,
- Education disk,
- Personal disk,
- Games disk,
- OS-65D V3.0 program development disk

BASIC and assembler run under the OS-65D V3.0 disk operating system.
All systems supplied with full documentation.

Available now:

COMPUCOLOUR II MODEL 3
- 13" 8-Colour CRT, 8080 Microprocessor
- 16K extended disk BASIC in ROM
- 71-key detached keyboard
- 8K RAM memory for user programs
- 64 characters per line by 32 lines per page
- Special graphics package with 128 x 128 point plotting
- Built-in mini-floppy disk drive
- 50 pin bus
- RS-232 I/O port for serial printers, etc. £1,390.00

THE RESEARCH MACHINES 380Z COMPUTER SYSTEM

THE RESEARCH MACHINES 380Z
A UNIQUE TOOL FOR RESEARCH AND EDUCATION

Microcomputers are extremely good value. The outright purchase price of a 380Z installation with dual mini floppy disk drives, digital I/O and a real-time clock, is about the same as the annual maintenance cost of a typical laboratory minicomputer. It is worth thinking about!

The **RESEARCH MACHINES 380Z** is an excellent microcomputer for on-line data logging and control. In university departments in general, it is also a very attractive alternative to a central mainframe. Having your own 380Z means an end to fighting the central operating system, immediate feedback of program bugs, no more queuing and a virtually unlimited computing budget. You can program in interactive BASIC or, using our unique Text Editor, run very large programs with a 380Z FORTRAN Compiler. If you already have a minicomputer, you can use your 380Z with a floppy disk system for data capture.

What about Schools and Colleges? You can purchase a 380Z for your Computer Science or Computer Studies department at about the same cost as a terminal. A 380Z has a performance equal to many minicomputers and is ideal for teaching BASIC and Cesil. For A Level machine language instruction, the 380Z has the best software front panel of any computer. This enables a teacher to single-step through programs and observe the effects on registers and memory, using a single keystroke.

WHAT OTHER FEATURES SET THE 380Z APART?
The 380Z with its professional keyboard is a robust, hardwearing piece of equipment that will endure continual handling for years. It has an integral VDU interface – you only have to plug a black and white television into the system in order to provide a display unit – you do not need to buy a separate terminal. The integral VDU interface gives you upper and *lower* case characters and low resolution graphics. Text and graphics can be mixed *anywhere* on the screen. The 380Z has an integral cassette interface, software and hardware, which uses *named* cassette files for both program and data storage. This means that it is easy to store more than one program per cassette.

Owners of a 380Z microcomputer can upgrade their system to include floppy (standard or mini) disk storage and take full advantage of a unique occurence in the history of computing – the CP/M™* industry standard disk operating system. The 380Z uses an 8080 family microprocessor – the Z80 – and this has enabled us to use CP/M. This means that the 380Z user has access to a growing body of CP/M based software, supplied from many independent sources.

380Z mini floppy disk systems are available with the drives mounted in the computer case itself, presenting a compact and tidy installation. The FDS-2 standard floppy disk system uses double-sided disk drives, providing 1 Megabyte of on-line storage.

**Trademark, Digital Research.*

Versions of BASIC are available with the 380Z which automatically provide controlled cassette data files, allow programs to be loaded from paper tape, mark sense card readers or from a mainframe. A disk BASIC is also available with serial and random access to disk files. Most BASICs are available in erasable ROM which will allow for periodic updating.

If you already have a teletype, the 380Z can use this for hard copy or for paper tape input. Alternatively, you can purchase a low cost 380Z compatible printer for under £300, or choose from a range of higher performance printers.

380Z/32K complete with SINGLE MINI FLOPPY DISK SYSTEM MDS-1

£1787.00

380Z/16K System with Keyboard

£965.00

RESEARCH MACHINES Computer Systems are distributed by RESEARCH MACHINES LTD., P.O. Box 75, Chapel Street, Oxford. Telephone: OXFORD (0865) 49792. Please send for the 380Z Information Leaflet. Prices do not include Carriage or VAT @ 8%.

PCW SEMINAR

TICKETS £12 PLUS FREE EXHIBITION – EVERYONE WELCOME

an introduction to
Personal and Business Computing

PCW in association with **SYBEX** presents **Rodnay Zaks** speaking on the various aspects of small computers: understanding them and using them.

Rodnay **Zaks** is the author of best sellers "Introduction to Personal and Business Computing", "Microprocessors: from chips to systems".

As a speaker, he is in great demand in the United States and the Continent.

Now **PCW**, in keeping with its high standards, has secured his services as a speaker.

BOOK YOUR TICKETS BY PHONE 01 229 5589

Date: Saturday, May 12th

Venue: The West Centre Hotel, Lillie Road, London SW6

Time: 9.30 a.m. to 5 p.m. (Registration at 9 a.m.)

Personal Computer World aims to give a definitive introduction to, as well as a refresher in, personal and business computing with this Seminar. Few people are better equipped to do this than **Rodnay Zaks**. The opportunity to take advantage of his first appearance in Britain should not be missed!

Price of tickets is £12

A note to CASBA members: Membership of CASBA entitles you to a 12½% reduction in the price of tickets.

How to obtain your tickets: Simply fill in the coupon below and send it together with a cheque or postal order payable to Intra Press, or enclose payment with a letter.

To: **PCW Seminar**, 62A Westbourne Grove, London W2.

Please send me .... tickets at £12 each*. I enclose a cheque/P.O. for £......... made payable to Intra Press

Name ______________________________

Company/Organisation ______________________________

Address ______________________________

______________________________ Tel No.: ______________________________

**Less*
12½% reduction for CASBA members. (Send £10.50 – Save £1.50)

Mugs Away

A Computer Darts Trainer

Sheridan Williams

Computers can be used for training in many fields. In teaching they can be used in programmed learning, and fill many roles as teaching aids. In aviation they can be used for flight simulation and pilot training. I wondered whether it was possible to develop a program to help with a sport such as darts. I felt that the computer's main role in this field would be to provide an opponent of known, but adjustable, standard. By constantly playing at a fixed standard it would be possible to gauge whether one's standard was improving, simply by counting the games that you won in a series.

Five hundred pounds down the throat

I myself am a keen darts player, but find that to keep in practice, I need to visit the 'local' every night. This can turn out to be rather expensive; an estimate of the cost is, 5 pints a night, 6 nights a week, total £500 per year.

Seriously though, there are many times when you either have no opponent, or the opponent you have is of a completely different standard to you. If you have a portable computer then there is no reason why you should not take it to the 'local' with you. The computer will provide an opponent of any level of skill, and by altering the computer's standard it can keep pace with you as you improve.

In the following program I have attempted to provide a very sophisticated and realistic simulation of a genuine darts game. For this reason the program is rather complicated, and does *not* select a random number between 1 and 20 and make that its score; instead it simulates a real game, deciding on the best place to aim; and throws a dart at that point. Throws are made on a statistical basis, and impact with the board is calculated in cartesian coordinates which are later converted to polar form for ease of interpretation.

As an example of its play, if it aimed for treble 'top' it could easily fall into a 1, 5, treble 1, or treble 5; or if the standard is really low then it could miss the board altogether. Try the game on standard 40 and watch 'bed and breakfast' come up occasionally. The computer will provide a genuine opponent at any standard, and with any start 101, 301, 1001, etc. It plays both 'bust' and 'no bust' rules and allows double starts if required.

Flexible Standards

I will give a rough indication of the standards of the computer's play; you will note that the range of standards is very wide and that when you find standard 60 too easy and 61 too difficult you can use a standard of 60.5.

Standard of play	Type of player
1	Rubbish
40	Average
60	Local team
70	County team
85	Champion
100	Virtually perfect

It is possible to watch your standard improve over the weeks. I have had many hours improving my play, and have managed to improve by 6 points in 2 weeks.

The program has been written and developed on a Research Machines 380Z, but I have kept to standard BASIC to enable the program to run on any system. As it stands the program is fairly long but can be shortened using the hints given at the end of this article. I have had to use trig functions I'm afraid, but only the standard ones. The output is designed for a small sized screen of 40 character width, and hopefully should need no modification in that department. To summarise: the program was written to be as portable as possible. a point which should always be taken into account. I have taken no account of speed as the response time is rapid enough, instead I have tried to keep the size of the program to a minimum.

The algorithm

I will outline the algorithm used for the program, for those who might just be interested. The board is represented using polar coordinates. The origin is the centre (bulls-eye) and the zero degree line is vertically upwards through the centre of the 20 section. The radii are all measured in millimetres, and need not be exact as long as the ratios are in proportion. The measurements taken from my board at home are: distance to end of inner bull 7mm, to end of outer bull 16mm, treble ring from 95-105mm, double ring from 157-167mm. If these figures need altering then you will find them on lines 1140, 1180, 1220, 1240, 1280, 1300; however, these figures are not critical and should be left as they are. As you can see the board is a standard board with both treble and double rings. It is because most of the game will be centred around the 20 that I have chosen the above coordinate system, namely, zero degrees is vertically upwards. It should be noted that whatever the aim point is, the board is rotated so as to make that aim point vertical; this is only valid if the errors in both directions are the same. It is assumed that every throw has an error that can be resolved in two directions, horizontal and vertical; and although strictly speaking not true, the two errors are very nearly the same. If you don't believe me then try a small series of tests on yourself. If there are any differences they will be minute.

The program needs first to decide upon an aim point, this is done in the sub-routine 1580 and returns with the number to be aimed at and whether a double, treble or single is required. The program then makes a throw by calculating separate vertical and horizontal error components in cartesian coordinates, using a Normal distribution.

A brief note should be made here about lines 670 and 680. The BASIC RND function generates a random number between 0 and 1 but it is a rectangular distribution; we want a 'Normal' distribution. This can be achieved by using the function R = M + S*TAN(2.498*RND(1)–1.249) where M and S are the required mean and standard deviation of the normal distribution. This function is only an approximation to the normal distribution and has a maximum error of 10% which is good enough for our purposes; the distribution is also limited to ±3sd. From this point of impact a score is obtained, by converting the cartesian coordinates into polar form. This gives a distance from the board's centre and also an angle, which enable the program to calculate the score.

The target

I have not mentioned how the program calculates *what* to aim for. A decision is made whether a double is required, this is fairly straightforward; either a double start is required or the score is even and below 40. The aim point is usually for treble 20 except when this would "bust" the player, or leave him with a low odd score. The program tries to home in on double 16, 8, 4, 2, 1 as would a professional player.

I am afraid that the program cannot check on the validity of your score, other than it being over 180.

You may select any starting option from the following list:

1) No bust, or otherwise. 2) Any starting score. 3) Double start or otherwise.

The program was not written with speed in mind as there is no appreciable delay between throws anyway. It has been written as concisely as is prudent, without making it unintelligible. If it is still too long to fit into your machine, then delete all or some of the following:—

Delete all REMark statements.

Replace the word PRINT by its equivalent symbol. (eg a "?" on the PET and 380Z).

Delete lines 160,200,210,270,280, 280,320,330,380-400,430-450,490-540.

If at all possible combine statements 790,800 and 860,870,880 and 830, 840

Finally, if you have any problems getting the program to run properly please contact me and not PCW. Oh, and do not site your computer too close to the dart board — a ricochet could be costly.

COMPETITION – INCREASE IN PRIZE MONEY!

Write a program in BASIC that will solve an "ALPHAMETIC". An alphametic is an alphabetic sum. For example CROSS+ROADS=DANGER or HORSE+OFTHE+YEAR=HARVEY. Each letter must be replaced by a single digit 0-9, and no digits are repeated. Entries will be judged by timing and the neatness of the output. Programs will be run on a Research Machines 380Z which has very fast function evaluation. (see John Coll's benchmark tests in the November issue). I will accept specialised programs that solve one particular alphametic provided I receive no general ones. Entries close two months after publication of this article; and the prizes will be £15 to the best entry, and £10 to the runner-up. Enclose a SAE, and state if you require the programs return, enclose a sample output if possible. Entries to me at 114 Beech Road, St. Albans, Herts. AL3 5AU.

DARTS TRAINER

```
10 REM SHERIDAN WILLIAMS DARTS TRAINER
20 REM ------------------------------
30 CLEAR 2000
40 DIM A(30)
50 FOR X=1 TO 30
60 READ A(X)
70 NEXT X
80 DATA 11,14,9,12,5,20,1,18,4,13,6,10,15,2,17
90 DATA 3,19,7,16,8,11,14,9,12,5,20,1,18
100 DATA 4,13
110 Y(1)=0
120 Y(2)=12
130 Y(3)=100
140 Y(4)=162
150 Y(5)=132
160 PRINT
170 PRINT
180 PRINT"DO YOU PLAY THE 'NO BUST' RULE";
190 INPUT B$
200 IF B$="YES" THEN 230
210 IF B$="NO" THEN 230
220 GOTO 160
230 PRINT
240 PRINT
250 PRINT"STANDARD OF PLAY (BETWEEN 1 & 100)"
260 INPUT S1
270 IF S1<1 THEN 250
280 IF S1>100 THEN 250
290 D1=50-0.47*S1
300 PRINT"STARTING SCORE";
310 INPUT T1
320 IF T1<2 THEN 300
330 IF T1<>INT(T1) THEN 300
340 T2=T1
350 PRINT
360 PRINT"DO YOU WANT A DOUBLE START";
370 INPUT D$
380 IF D$="YES" THEN 410
390 IF D$="NO" THEN 410
400 GOTO 360
410 PRINT"ARE YOU STARTING";
420 INPUT Q$
430 IF Q$="YES" THEN 460
440 IF Q$="NO" THEN 460
450 GOTO 410
460 GOSUB 2050
470 IF Q$="NO" THEN 640
480 INPUT Y
490 IF Y>180 THEN 530
500 IF Y<0 THEN 530
510 IF Y<>INT(Y) THEN 530
520 GOTO 550
530 PRINT"SILLY"
540 GOTO 480
550 IF T2=Y THEN 610
560 IF T2-Y>1 THEN 580
570 Y=0
580 T2=T2-Y
590 PRINT TAB(6);T2;
600 IF T2>0 THEN 640
610 PRINT
620 PRINT"YOU WIN..DAMN; STANDARD WAS";S1
630 GOTO 230
640 S=0
650 FOR X0=1 TO 3
660 GOSUB 1580
670 X9=D1*TAN(2.498*RND(1)-1.249)
680 Y9=Y(D9)+D1*TAN(2.498*RND(1)-1.249)
690 L1=SQR(X9*X9+Y9*Y9)
700 IF X9<>0 THEN 730
710 L2=0
720 GOTO 750
730 L3=57.21*ATN(Y9/X9)
740 L2=90*SGN(L3)-L3
750 GOSUB 1030
760 GOSUB 1420
770 S=S+T4
780 IF T1-S<>0 THEN810
790 IF D8=4 THEN 1000
800 IF D8=1 THEN 1000
810 IF T1-S<2 THEN 960
820 IF S1>40 THEN 930
830 IF B$="YES" THEN 930
840 IF T4/2=INT(T4/2) THEN 930
850 T9=(T1-S)/2
860 IF T9<>INT(T9) THEN 930
870 IF T9>20 THEN 930
880 IF X0=3 THEN 930
890 FOR X7=1 TO 3-X0
900 ?"     MISS";
910 NEXT X7
920 GOTO 940
930 NEXT X0
940 T1=T1-S
950 GOTO 980
960 IF B$="NO" THEN 980
970 T1=T1-S+T4
980 PRINT TAB(35);T1
990 GOTO 480
1000 PRINT
1010 PRINT"TEE-HEE...MY GAME AT STANDARD";S1
1020 GOTO 230
1030 REM TO RETURN SCORE AND WHETHER D, B, OB, T
```

```
1040 IF D9=1 THEN 1910
1050 FOR X=6 TO 25
1060 IF A(X)=V THEN 1080
1070 NEXT X
1080 D8=5
1090 FOR J1=9 TO 99 STEP 18
1100 IF ABS(L2)<J1 THEN 1130
1110 X=X+SGN(L2)
1120 NEXT J1
1130 T4=A(X)
1140 IF L1>7 THEN 1180
1150 D8=1
1160 T4=50
1170 RETURN
1180 IF L1>16 THEN 1220
1190 D8=2
1200 T4=25
1210 GOTO 1380
1220 IF L1>95 THEN 1240
1230 GOTO 1380
1240 IF L1>105 THEN 1280
1250 D8=3
1260 T4=3*T4
1270 GOTO 1380
1280 IF L1>157 THEN 1300
1290 GOTO 1380
1300 IF L1>167 THEN 1350
1310 D8=4
1320 T4=T4*2
1330 D$="NO"
1340 RETURN
1350 T4=0
1360 D8=6
1370 RETURN
1380 IF D$="NO" THEN 1410
1390 T4=0
1400 D8=6
1410 RETURN
1420 REM TO PRINT SCORE
1430 PRINT TAB(8*X0+6);
1440 ON D8 GOTO 1540,1520,1500,1480,1460,1560
1460 PRINT T4;
1470 RETURN
1480 PRINT"D";A(X);
1490 RETURN
1500 PRINT"T";A(X);
1510 RETURN
1520 PRINT"OUT B";
1530 RETURN
1540 PRINT"BULL";
1550 RETURN
1560 PRINT"MISS";
1570 RETURN
1580 REM TO SET V AND D9
1590 D9=3
1600 T4=T1-S
1610 IF T4>180 THEN 1840
1620 R6=T4/2
1630 IF T4<=40 THEN 1690
1640 IF T4=50 THEN 1860
1650 IF T4<=52 THEN 1700
1660 IF R6=INT(R6) THEN 1840
1670 V=19
1680 GOTO 1880
1690 IF R6=INT(R6) THEN 1810
1700 FOR X6=6 TO 1 STEP -1
1710 R6=INT(2↑X6+0.1)
1720 IF T4>R6 THEN 1750
1730 NEXT X6
1750 V=T4-R6
1760 D9=5
1770 IF D$="NO" THEN 1800
1780 V=1
1790 D9=4
1800 RETURN
1810 V=R6
1820 D9=4
1830 GOTO 1880
1840 V=20
1850 GOTO 1880
1860 D9=1
1870 RETURN
1880 IF D$="NO" THEN 1900
1890 D9=4
1900 RETURN
1910 REM BULL FINISH
1920 L9=-.362+.031*S1-.000724*S1↑2+.00000541*S1↑3
1930 IF RND(1)<L9 THEN 2010
1940 D8=5
1950 IF RND(1)<L9 THEN 1980
1960 T4=INT(20*RND(1)+1)
1970 GOTO 1410
1980 D8=2
1990 T4=25
2000 GOTO 1410
2010 D8=1
2020 T4=50
2030 D$="NO"
2040 GOTO 1410
2050 REM TO PRINT HEADINGS
2060 PRINT
2070 PRINT"YOUR  YOUR     M Y   S C O R E     M Y"
2080 PRINT"SCORE TOTAL THROW1  THROW2  THROW3 TOTAL"
2090 PRINT"----- ----- ------  ------  ------ -----"
2100 RETURN
2110 END
```

STATPACK

Colin Chatfield

This seventh article in the series produces a graphical representation of a set of figures as a bar chart. Either the figures of an individual column or all columns can be shown. The program is versatile in its ability to be displayed on terminals with varying character widths.

The usual opening statements from line 5 to 1160 are used with sub-routines at lines 9000 to 9690 to get the stored data from the disc. As a reminder to new readers this series is written in MSI Basic and was designed to be used with a MSI Disk and output to some form of hard copy although this is not necessary. The first question (line 20) requests the 'output port number'; using the SS50 line system the output port can be controlled by software in the statement PRINT (OR?) # (X), where X is the output part to be used. We use the variable Z9 for this purpose and in our case set it to port # 1, the control port for the VDU or printer or to port # 3 so that the questions come up on the VDU and the results are printed on hard copy.

This module has some additional sub-routines which are used to display the chart. Lines 9210 – 9260 produce the head and foot scales, the width of which is set by the question at line 4010. A zero is printed 5 spaces from the left edge followed by 4 dashes. Due to S being increased by 10, and a loop having been entered, the next number to the printed is 10. The scale then proceeds until the set width is reached. R has been found unnecessary in production and can be deleted in lines 9220, 9230 and 9250. When numbers with 3 digits, i.e. 100,110 etc., are printed it is necessary to back space oneplace otherwise the seals would spread too much. This is achieved at line 9240 with the statement PRINT or? # (Z9), CHRS (8);. The CHRS (8) is the ASCII number for 'BACKSPACE'. This is also used at line 1050 to place the full stop '.' immediately after a number rather than a space later.

The module is centered on lines 4000 to 4590. Line 4015 checks to see that a multiple of 10 for the scales has been entered and then a request is given asking for the column number; or if the total of all columns are to be displayed enter 999. Line 4030 sets the printout symbol for the lines of the chart and can be varied as required.

4140 and 4145 center the heading according to the width of the display. The marker heading is printed by line 4150. For a single column the FOR/NEXT loop in lines 4100-4177 finds the highest number in the column. At lines 4180 to 4250 the length of the line to be printed for the value of the number being considered is calculated as a proportion of the width of the display which is now equal to the highest number in the column. The item number is printed and if a very low value a decision is made to print one or two '*'s . Line 4240 prints out a stream of '*'s for higher numbers and in all cases the actual value is then printed at the end of the line. This procedure continues until the end of the column reached.

If all columns are being considered line 4500 to 4590 do the same job though adding up the total of the columns and putting the result in to D(I). If there are more than nine columns in the data an array could have been set up at line 4090 which should read IF B > 9 THEN Dim D (B). As it is now it will work quite well but an earlier amendment had not been altered in the listing. (An array is unnecessary for less than 10 items in MSI BASIC).

The foot markers are now printed, line 4270, and the scale is printed at line 4280. Five blank lines are output and the program then asks if more processing is required.

The next article is a spare module for those wishing to set up their own part of the package for special statistics.

```
0005 REM STAT7 - BAR CHART PROGRAM
0020 INPUT "     ENTER PORT # ",Z9
0080 LINE= 132: GOSUB9380
0100 ? TAB(22);"BAR CHART PROGRAM"
1000 REM ARRAY SIZE AND CHECKS
1010 GOSUB 9600
1050 ? "YOUR ARRAY IS ";A;"X ";B;:?CHR$(8);:?". ";A*B;"ITEMS."
1110 INPUT "     PRESS RETURN WHEN READY",A$
1120 GOSUB 4000
1125 GOSUB 9360
1130 IF A$="N"THEN?TAB(20);"STATPACK END": END
1135 IF A$<>"Y"THEN1125
1140 INPUT "     ENTER `Y' FOR BAR CHART",A$
1150 IF A$="Y"THEN1120
1160 CHAIN STAT1
4000 GOSUB 9380:?TAB(27);"BAR CHART":?
4010 INPUT "     ENTER SCALE IN MULTIPLES OF 10 ",A9
4012 IF A9>120THEN?"NOT GREATER THAN 120":GOTO4010
4015 IF INT(A9/10)<>A9/10 THEN4010
4020 INPUT "ENTER COLUMN BAR CHART REQUIRED FOR - 999 FOR ALL.",B2
4030 A$="*"
4040 IF B2=999THEN4080
4050 IF B2>B THEN4020
4060 IF B2<1 THENRETURN
4070 GOTO 4130
4080 REM SETS UP ARRAY OF TOTALS OF COLUMNS
4090 IF A8<>1THEN DIMD(B):A8=1
4100 FOR I=1TOB:A6=0:FORJ=1TOA:A6=A6+C(J,I):NEXTJ:D(I)=A6:NEXTI
4110 A5=0:FORI=1TOB:IFD(I)>A5THENA5=D(I)
4120 NEXT I
4130 GOSUB 9300:GOSUB9380
4140 ? #(Z9),TAB((A9/2)-4);"*** BAR CHART ***"
4145 ? #(Z9),TAB((A9/2)-1);"COLUMN # ";B2:?#(Z9)
4150 GOSUB 9160:GOSUB9220:IFB2=999THEN4500
4160 A5=0
4175 FOR I=1TOA:IFC(I,B2)>A5 THENA5=C(I,B2)
4177 NEXT I
4180 FOR I=1TOA
4190 A3=INT((C(I,B2)*A9)/A5):?#(Z9);I;
4200 IF A3<1 THEN 4220
4210 IF A3=2THEN4230
4215 IF A3>2 THEN4240
4220 ? #(Z9),TAB(5);A$;:GOTO4260
4230 ? #(Z9),TAB(5);A$;A$;:GOTO4260
4240 ? #(Z9),TAB(5);:FORK=1TOA3:?#(Z9),A$;:IFK=A9THEN4260
4250 NEXT K
4260 ? #(Z9)," ";C(I,B2):NEXTI
4270 GOSUB 9220:GOSUB9160
4280 ? #(Z9),"   SCALE = ";A5;":";A9:GOSUB9300:RETURN
```

```
4500 REM TOTALS OF COLUMNS
4510 FOR I=1TOB:A3=INT((D(I))*A9)/A5:?#(Z9);I;
4520 IF A3<1THEN4550
4530 IF A3=2THEN4560
4540 IF A3>2THEN4570
4550 ? #(Z9),TAB(5);A$;:GOTO4590
4560 ? #(Z9),TAB(5);A$;A$;:GOTO4590
4570 ? #(Z9),TAB(5);:FORK=1TOA3:?#(Z9),A$;:IFK=A9THEN4590
4580 NEXT K
4590 ? #(Z9)," ";D(I):NEXTI:GOTO4270
9000 REM SUB PROGRAMS
9160 FOR L=1TO(A9+5): ?#(Z9),"-";: NEXTL: ?#(Z9): RETURN
9210 REM SUBROUTINE TO PRINT SCALE HEADING
9220 R=5: S=0: ?#(Z9),"    0";
9230 ? #(Z9),TAB(R);"- - - -";
9240 S=S+10: IFS>=100 THEN?#(Z9),CHR$(8);
9250 ? #(Z9),S;:R=R+5
9260 IF S=A9THEN?#(Z9): RETURN
9270 GOTO 9230
9300 FOR I=1TO5:?#(Z9):NEXTI:RETURN
9360 INPUT "     ENTER `Y' FOR MORE, `N' FOR NONE ",A$: RETURN
9380 ? CHR$(25);:?CHR$(25);:?CHR$(22);:?CHR$(12);:RETURN
9400 B2=0:IFB=1THENB2=1:GOTO9430
9410 ? :INPUT"     COLUMN # STATISTICS REQUIRED FOR ",B2
9420 IF B2>B THEN?"TOO HIGH";:GOTO9410
9430 ? #(Z9):RETURN
9600 OPEN #10, STATFL1 FOR INPUT: FIELD#10,F=6
9610 OPEN #20, STATFL2 FOR INPUT: FIELD#20,A=6,B=6
9620 SET #10=1: SET#20=1: GET#20: DIMC(A,B)
9660 FOR I=1TOA:FORJ=1TOB:GET#10:C(I,J)=F:NEXTJ:NEXTI
9683 INPUT "     ENTER `Y' FOR VISUAL OF DATA",A$:IFA$<>"Y"THEN9690
9685 ? :FORI=1TOA:FORJ=1TOB:?C(I,J);:NEXTJ:?:NEXTI:?
9690 CLOSE #10:CLOSE#20:RETURN

THE GHOST
#
```

```
 CHAINSTAT7
     ENTER PORT # ? 1
                        BAR CHART PROGRAM
     ENTER `Y' FOR VISUAL OF DATA? Y

3 2 0 0 2 3 1 1 50 5 0 0 0 1 0 0 0 0 0
3 1 0 0 2 4 1 1 100 2 0 2 0 0 0 0 0 1 0
3 1 0 1 0 1 1 1 100 3 3 2 0 1 0 1 0 1 1
3 1 0 1 0 2 1 1 2 1 0 2 0 0 0 1 0 1 0
3 1 0 0 2 3 1 3 2 1 0 0 0 0 1 1 0 0 0
3 1 1 0 0 2 1 1 50 2 0 0 0 0 1 1 0 1 0

YOUR ARRAY IS 6 X 19. 114 ITEMS.
     PRESS RETURN WHEN READY?
                             BAR CHART

     ENTER SCALE IN MULTIPLES OF 10 ? 40
ENTER COLUMN BAR CHART REQUIRED FOR - 999 FOR ALL.? 6
```

```
                 *** BAR CHART ***
                    COLUMN # 6

---------------------------------------------
     0- - - -10 - - - -20 - - - -30 - - - -40
1    ****************************** 3
2    **************************************** 4
3    ********** 1
4    ******************** 2
5    ****************************** 3
6    ******************** 2
     0- - - -10 - - - -20 - - - -30 - - - -40
---------------------------------------------
    SCALE = 4 :40



      ENTER `Y' FOR MORE, `N' FOR NONE ? Y
      ENTER `Y' FOR BAR CHART? Y
                            BAR CHART

      ENTER SCALE IN MULTIPLES OF 10 ? 60
ENTER COLUMN BAR CHART REQUIRED FOR - 999 FOR ALL.? 6
```

```
                    *** BAR CHART ***
                       COLUMN # 6

-----------------------------------------------------------------
     0- - - -10 - - - -20 - - - -30 - - - -40 - - - -50 - - - -60
1    ********************************************* 3
2    ************************************************************ 4
3    *************** 1
4    ****************************** 2
5    ********************************************* 3
6    ****************************** 2
     0- - - -10 - - - -20 - - - -30 - - - -40 - - - -50 - - - -60
-----------------------------------------------------------------
    SCALE = 4 :60



      ENTER `Y' FOR MORE, `N' FOR NONE ? N
                     STATPACK END

THE GHOST
#
```

3D Noughts & Crosses

Steve Thomas

I read with interest Bill Davy's article on three dimensional noughts and crosses (December '78, PCW). About two years ago I acquired a program rather similar in design to Mr. Davy's, and ever since I have been devising methods of improving the way in which the program played. Extensive play-testing revealed four major flaws in the algorithm used by the original program.

1) The computer always made the same opening moves. This meant that once a winning sequence of moves had been found, it could be repeated. The best such strategy beat the computer in six moves. The solution is simply to introduce a random number to differentiate between moves of otherwise equal value.

2) The computer played with no overall strategy; it simply went all out to form lines of three and four. When it inevitably ran out of forcing moves, it was usually in a poor position, and quickly lost. It was observed that good human players rarely formed a line of three without having a larger plan in mind. Playing about with the 'priorities' — the values in Mr. Davy's Table 2 — to reduce the program's aggressiveness simply meant that the program played too defensively. No balance could be found. A solution was to vary the priorities as the game progressed, producing strategic play early on, then tactical play, and, should the game reach the closing stages, all out defence.

3) The program ignored concentrations of noughts and crosses on

the 18 planes. It is obviously undesirable to allow the opponent to concentrate his forces, and desirable for the computer to do it. The problem is solved by counting the number of noughts and crosses on each plane and adding another priority to the cells on the plane according to another table. A further refinement is to increase this priority if the cell is on a diagonal of the plane, as winning strategies depend on control of the diagonals. It is also convenient to treat the cells on the body diagonals (the eight corner and eight centre cells) as a nineteenth plane.

4) The program ignored the 'fork' move, i.e. forming two or more lines of three simultaneously, both in attack and defence. This can be cured by counting the number of lines of three formed by playing on each possible cell, and adding yet another priority if two or more lines of three are formed.

Time is the drawback

These improvements can be (and indeed were) implemented by extending the original program. The drawback was of course that the run time of the program was roughly doubled, and optimising the program for increased speed produced an unaesthetic mess. A rethink was clearly necessary, and a straight inversion of the representation of the lines produced a rather compact program which runs at about double the speed of the souped-up original. It requires rather more array space, but will run (just) on a Research Machines 380Z with 16K of memory, using the 9K BASIC, if the remarks are removed.

Array M holds the numbers of each of the lines passing through each cell. Those cells which only have four lines passing through them are padded with 'null' line numbers — 76 in this case. The ordering of the line numbers is such that the plane number can be obtained by dividing the line number by four (discarding the remainder), and the third row of the array holds those lines which are diagonals on the planes they represent, for those cells which are not on body diagonals. Array W holds the main priorities. Each row holds the priorities for: 1—empty, 2—0, 3—00. 4—000, 5—X, 6—XX, 7—XXX, 8—anything else, and the row used changes every four moves. Array P holds the priorities for the planes. To keep things simple, the array is kept to 8x8 and more 0's and X's are ignored. Array G holds the state of each cell, array N the number of 0's and X's on each line, and E the number of 0's and X's on each plane.

The section of code which works out the computer's best move has been written in a form which is reasonably transportable. The routine to display the position makes use of the graphics available within this BASIC and must obviously be rewritten for other BASICs.

Priorities

The priorities in this program are probably far from optimal. However, beating the program consistently is not particularly easy. Like Mr. Davy, I leave it as an exercise to find a better set of priorities. Readers attempting his second question, that of writing a program to play in four dimensions, should be aware that the board must be 5x5x5x5, because any less leads to triviality and any more leads to consistent draws. I suspect, however, that 5x5x5x5 would need some additional rules about the centre cell, as the first player to put his mark there would have a rather easy time of things. This leaves 6x6x6x6x6, but display is something of a problem here.

PCW *Steve Thomas works in the Software department of Research Machines Ltd.* PCW

3D Noughts and Crosses.

```
10 GRAPH 0 :REM RESTORE THE SCROLLER
20 RESTORE
30 CLEAR 100 :REM RESERVE STRING SPACE
40 ?CHR$(12) :REM CLEAR THE SCREEN
50 RANDOMIZE
60 ?TAB(13);"N O U G H T S"
70 ?TAB(13);"-------------"
80 ?
90 ?TAB(17);"A N D"
100 ?TAB(17);"-----"
110 ?
120 ?TAB(13);"C R O S S E S"
130 ?TAB(13);"-------------"
140 ?:?:?:?:?:?:?
150 DIM M(63,6),N(75),E(18),D(63),W(63)
160 DIM P(63),H(7),C(10),G(63),Q(15),B(13)
170 REM INITIALIZE THE DATA ARRAYS
180 FOR A = 0 TO 6
190  FOR B = 0 TO 63
200   READ M(B,A) :REM THE POINTERS TO THE LINES
210  NEXT B
220 NEXT A
230 FOR A = 0 TO 15
240  READ Q(A) :REM POINTERS INTO ARRAY W
250 NEXT A
260 FOR A = 0 TO 7
270  READ H(A) :REM FORK COUNTER
280 NEXT A
290 FOR A = 0 TO 63
300  READ W(A) :REM MAIN PRIORITIES
310 NEXT A
320 FOR A = 0 TO 10
330  READ C(A) :REM FORK PRIORITIES
340 NEXT A
350 FOR A = 0 TO 63
360  READ P(A) :REM PLANES PRIORITIES
370 NEXT A
380 FOR A = 0 TO 13
390  READ B(A) :REM DIAGONALS PRIORITIES
400 NEXT A
410 FOR A = 0 TO 75
420  N(A)=0 :REM LINES COUNTERS
430 NEXT A
440 FOR A = 0 TO 63
450  G(A)=0 :REM POSITION ARRAY
460 NEXT A
470 FOR A = 0 TO 18
480  E(A)=0 :REM PLANES COUNTERS
490 NEXT A
500 B$ = ".XO"
510 M = 1 :REM MOVE NUMBER
520 Q = 76 :REM NUMBER OF UNBLOCKED LINES
530 GRAPH 1 :REM SET THE SCROLLER
540 FOR F = 0 TO 63
550  GOSUB 1180 :REM DISPLAY EMPTY BOARD
560 NEXT F
570 C = 22 :REM DISPLAY TITLE
580 FOR D = 1 TO 17
590  PLOT C,45,ASC(MID$("LAST        MOVE",D,1))
600  C = C+2
610 NEXT D
620 INPUT "Shall I go first"; A$
630 Z = LEFT$(A$,1)="Y"
640 IF Z THEN 850
650 ?"Your move number";M+Z;"please";
660 INPUT F :REM GET THE MOVE
670 F=16*F-159*INT(F/10)-6*INT(F/100)-21
680 REM CONVERT FROM EXTERNAL TO INTERNAL COORDS
690 IF F >= 0 AND F<=63 THEN 720
700 ?"Illegal move"
710 GOTO 650
720 IF G(F) THEN?"Cell occupied":GOTO 650
730 G(F)=1 :REM UPDATE THE POSITION
740 GOSUB 1180 :REM ON THE SCREEN
750 REM NOW UPDATE THE LINES AND PLANES COUNTERS
760 FOR B = 0 TO 6+3*(M(F,6)=76) :REM FOR EACH LINE
770  N = N(M(F,B))
780.  E = E(M(F,B)/4)
790  IF N = 3 THEN?"OK, you win":GOTO 1300
800  IF N=INT(N/4)*4 AND N>3 THEN Q = Q-1
810  IF Q=0 THEN 1290 :REM TEST FOR DRAW
820  N(M(F,B)) = N+1 :REM UPDATE LINE
830  E(M(F,B)/4) = E-((E-INT(E/8)*8)<7) :REM AND PLANE
840 NEXT B
850 N=INT((M-1)/4)*8 :REM GET STRATEGY
860 P = -10000 :REM VALUE OF BEST MOVE SO FAR
870 FOR A = 0 TO 63 :REM FOR EACH CELL
880  IF G(A) THEN 1020 :REM EMPTY?
890  T = 0 :REM VALUE OF THIS CELL
900  E = 0 :REM FORK COUNTER
910  B = 6+3*(M(A,6)=76) :REM NUMBER OF LINES
920  G = (B/3-1)*7 :REM BODY DIAGONAL?
930  FOR C = 0 TO B
940   D = Q(N(M(A,C))) :REM POINTER
```

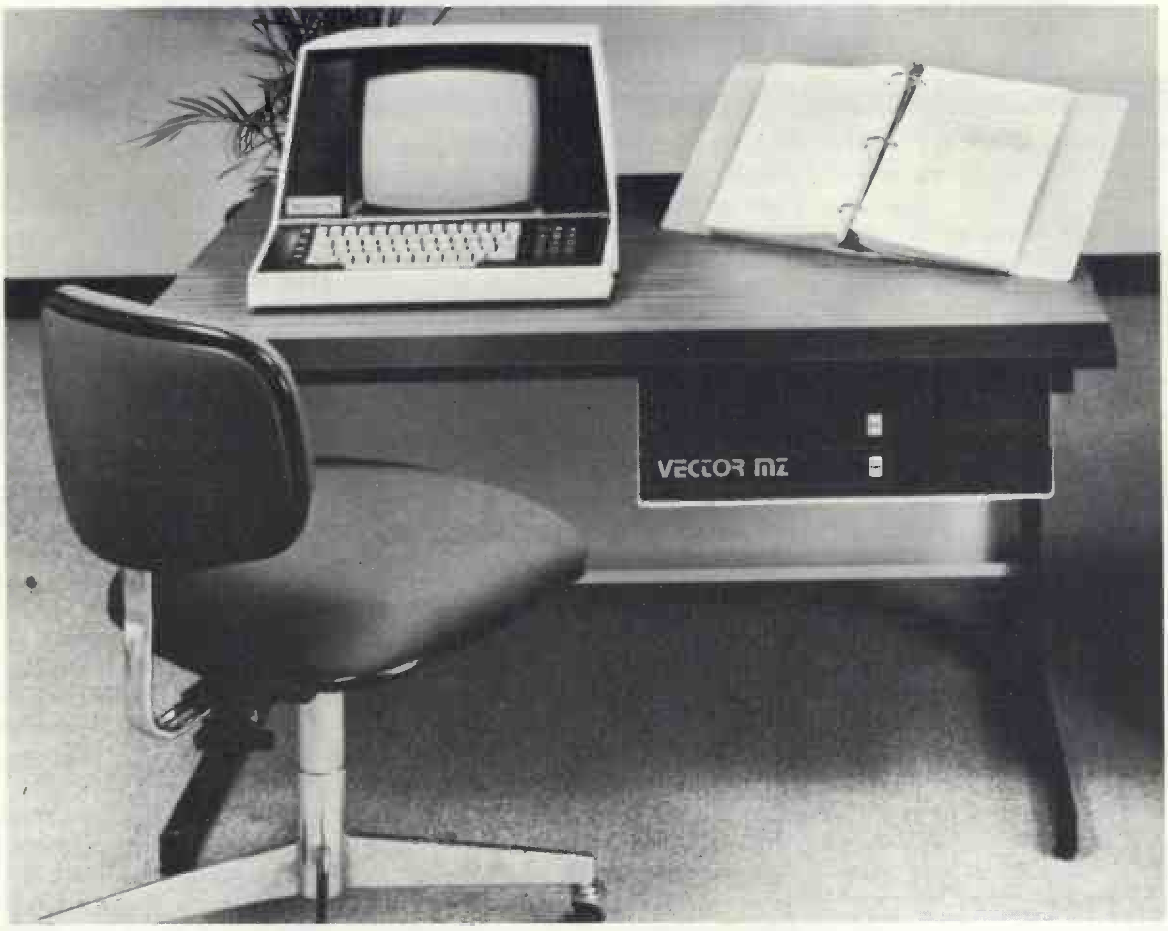
```
950   T = T+W(N+D)+P(E(M(A,C)/4))*B(G+C)
960   E = E+H(D) :REM CHECK FOR FORKS
970  NEXT C
980  IF E>10 THEN E=10 :REM MAXIMUM VALUE
990  T = T+C(E) :REM ADD FORK PRIORITY
1000  REM TEST IF THIS CELL IS ANY GOOD
1010  IF T+RND(1)>P THEN P=T+.5:F=A:T2=E
1020 NEXT A
1030 G(F) = 2 :REM UPDATE CELL
1040 GOSUB 1180 :REM ON THE SCREEN
1050 REM UPDATE LINES AND PLANES
1060 FOR B = 0 TO 6+3*(M(F,6)=76)
1070  N = N(M(F,B))
1080  E = E(M(F,B)/4)
1090  IF N=12 THEN ?"I win":GOTO 1300
1100  IF N<4 AND N<>0 THEN Q=Q-1
1110  IF Q=0 THEN 1290
1120  N(M(F,B)) = N+4 :REM LINES
1130  E(M(F,B)/4) = E-8*(E<56) :REM PLANES
1140 NEXT B
1150 M = M+1 :REM NEXT MOVE
1160 GOTO 650
1170 REM DISPLAY THE MOVE
1180 C = INT(F/16)
1190 E = F-16*C
1200 D = INT(E/4)
1210 E = E-4*D
1220 B = 36-3*C
1230 A = 20+10*D+2*E
1240 PLOT A,B,ASC(MID$(B$,G(F)+1,1))
1250 PLOT 34,45,D+49
1260 PLOT 38,45,E+49
1270 PLOT 42,45,C+49
1280 RETURN
1290 ?"A draw!"
1300 GRAPH 0
1310 INPUT "Do you want another game"; A$
1320 IF A$="Y" OR A$="YES" THEN 410
1330 IF A$="N" OR A$="NO" THEN 1360
1340 ?"Answer yes or no"
1350 GOTO 1300
1360 ?"OK OK end of program":END
1370 REM NOW THE DATA
1380 REM FIRST THE LINE POINTERS (448)
1390 DATA 0,0,0,0,1,37,38,1,2,41,42,2,3,3,3,3
1400 DATA 32,4,4,35,5,5,5,5,6,6,6,6,44,7,7,47
1410 DATA 32,8,8,35,9,9,9,9,10,10,10,10,44,11,11,47
1420 DATA 12,12,12,12,13,37,38,13,14,41,42,14,15,15,15,15
1430 DATA 16,33,34,28,16,20,24,28,16,20,24,28,16,45,46,28
1440 DATA 17,21,25,29,36,21,25,39,40,21,25,43,17,21,25,29
1450. DATA 18,22,26,30,36,22,26,39,40,22,26,43,18,22,26,30
1460 DATA 19,33,34,31,19,23,27,31,19,23,27,31,19,45,46,31
1470 DATA 32,20,24,35,36,1,1,39,40,2,2,43,44,20,24,47
1480 DATA 4,33,34,4,17,37,38,29,17,41,42,29,7,45,46,7
1490 DATA 8,33,34,8,18,37,38,30,18,41,42,30,11,45,46,11
1500 DATA 32,23,27,35,36,13,13,39,40,14,14,43,44,23,27,47
1510 DATA 48,65,66,52,57,48,52,61,58,52,48,62,52,69,70,48
1520 DATA 49,56,60,53,64,49,53,67,68,53,49,71,53,59,63,49
1530 DATA 50,60,56,54,68,50,54,71,64,54,50,67,54,63,59,50
1540 DATA 51,69,70,55,61,51,55,57,62,55,51,58,55,65,66,51
1550 DATA 56,76,76,60,76,76,76,76,76,76,76,76,59,76,76,63
1560 DATA 76,76,76,76,76,57,61,76,76,58,62,76,76,76,76,76
1570 DATA 76,76,76,76,76,61,57,76,76,62,58,76,76,76,76,76
1580 DATA 60,76,76,56,76,76,76,76,76,76,76,76,63,76,76,59
1590 DATA 64,76,76,67,76,76,76,76,76,76,76,76,68,76,76,71
1600 DATA 76,76,76,76,76,65,66,76,76,69,70,76,76,76,76,76
1610 DATA 76,76,76,76,76,69,70,76,76,65,66,76,76,76,76,76
1620 DATA 68,76,76,71,76,76,76,76,76,76,76,76,64,76,76,67
1630 DATA 72,76,76,73,76,76,76,76,76,76,76,76,74,76,76,75
1640 DATA 76,76,76,76,76,72,73,76,76,74,75,76,76,76,76,76
1650 DATA 76,76,76,76,76,75,74,76,76,73,72,76,76,76,76,76
1660 DATA 75,76,76,74,76,76,76,76,76,76,76,76,73,76,76,72
1665 REM PRIORITY ARRAY POINTERS (16)
1670 DATA 0,4,5,6,1,7,7,7,2,7,7,7,3,7,7,7
1680 REM ROWS OF TWO COUNTERS (8)
1690 DATA 0,0,5,0,0,1,0,0
1700 REM THE MAIN PRIORITIES (64)
1710 DATA  5,10, 1,4000, 1, 4,700,0
1720 DATA  5,10, 1,4000, 1, 5,700,0
1730 DATA  5,15, 5,4000, 5, 6,700,0
1740 DATA  5,15,25,4000, 1,12,700,0
1750 DATA 25,20,40,4000, 1,15,700,0
1760 DATA 25,30,40,4000, 1,15,700,0
1770 DATA 25, 1, 1,4000,10,40,700,0
1780 DATA 25, 1, 1,4000,10,40,700,0
1790 REM ROWS OF TWO PRIORITIES (11)
1800 DATA 0,0,160,180,200,0,20,180,200,200,400
1810 REM PLANES PRIORITIES (64)
1820 DATA   0,  0, 0,5,50,100,200,300
1830 DATA   0,  0, 0,0, 2, 60, 80,100
1840 DATA   0,  0, 0,0, 0,  1, 40, 60
1850 DATA   5,  0, 0,0, 0,  0,  1, 20
1860 DATA  55,  5, 0,0,-5,  0,  0,  0
1870 DATA  60, 50,10,0, 0,-20,  0,  0
1880 DATA 120,100,20,0, 0,  0,-20,  0
1890 DATA 150,120,25,0, 0,  0,  0,-20
1900 REM DIAGONAL MULTIPLIERS (14)
1910 DATA   1,  1,1.5,  1,  1,  1,  1
1920 DATA 1.5,1.5,1.5,1.5,1.5,1.5,1.5
```


TRUMPET, VOLUNTARY

Responding to TIDBITS information or an advertisement?
Say you saw it in PCW
Blow your own trumpet.
Mention your magazine.

VMZ
Sintrom MMT

* Z80 Hardware/Software Development System
* General Purpose Disk Microcomputer
* Business/Education/Industry



The **VMZ** is a solid base on which to build a computer system. Housed within a single attractive case is all the hardware required to implement a disc-based microcomputer. It has a Z80 processor operating at 4MHz. It has 32K of static RAM. It has 630K bytes of storage on twin Micropolis disks. It has an output card for RS232 VDU and parallel printer plus access to the powerful Extended Monitor program. In addition it is supplied with the Micropolis MDOS and BASIC fully configured, not only for the hardware provided, but for a wide selection of peripherals. £ 2300

The **MMT**, ideal for use with the VMZ, is a memory-mapped VDU complete with interface card for the S100 bus. A very full keyboard, including numeric keypad and cursor controls communicates with an 80 x 24 line display. £ 600

Sintrom **MICROSHOP**, formed to supply the rapidly growing market for low-cost microcomputer systems, has emerged as a major force in what is undoubtedly one of the most explosive expansionist markets in the computer industry.
The advent of the low-cost microcomputer is making a significant contribution throughout industry and commerce. For the first time high-powered computerised systems are available to implement a range of individual cost-effective industrial and office applications.

Full technical and sales support is available for the small OEM, systems house and dealer, providing a stable product base of total systems from a single source within the UK.
Prices do not include VAT.

SINTROM GROUP

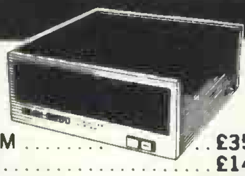
SINTROM MICROSHOP
14, Arkwright Road, Reading, Berks. RG2 OLS
Tel: Reading (0734) 85464 Telex 847395
Cables: SINTROM READING

the BYTE SHOP Ltd

ORDER BY POST FROM THE WIDEST RANGE OF MICRO-COMPUTERS IN BRITAIN... OR VISIT OUR STORES

A WIDE VARIETY OF SYSTEMS AND BOOKS AVAILABLE 'OFF THE SHELF'

SOUTHWEST TECHNICAL PRODUCTS — FOR 6800 USERS



MP-68CPU, 4K RAM	£355
8K additional RAM	£140
560KB twin disk system	£1,400
160KB twin mini floppy system	£860
CT-64 V.D.U. upper/lower case, monitor	£455
PR-40 40-column printer	£250
Cassette interface	£100
GT-61 Graphics Board	£105

A favourite with schools and colleges — low priced system that can grow.

Attractive low-priced software!

FOR THE PROFESSIONAL – CROMEMCO

Z-2 system: fully assembled	£575
Z-2 system: Kit	£395
Disk system from	£1,425
CS-3 System 3, 32K, dual disk, expandable	£4,175
4 MHZ Single card computer	£345
Bytesaver board with PROM Programmer	£135
16K PROM card	£135
Analogue Interface	£135
Disk — BASIC, FORTRAN, ASSEMBLER, COBOL, etc. etc...	each — £85

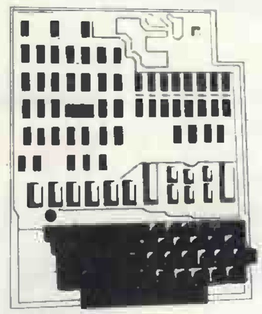
A rugged well-engineered system designed for hard use. Expandable to 512K,21 Boards.
Excellent software support CP/M AVAILABLE

OHIO SCIENTIFICS

Challenger IP	£395
C2-4P	£585
C2-P 20K + 5" disk	£1,560
C3-SI 32K + two 8" disks	£3,250
C3-A 48K + two 8" disks	£5,850
C3-B 48K + 74 M-byte hard disk	£12,350
Dual-sided disk enhancement	£682.50

Ohio Scientifics

Super board II



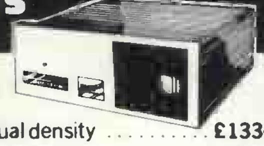
4K user RAM
Full 8K basic
Built and tested

£263.84

COMMODORE SYSTEMS

PET 2001-8	£550
PET MKII — Full Keyboard Plus Basic	£675
Pet Cassette Deck	£55
PET Disk Drives	£740
PET Printer	£550
Software from	£3.00
Kim-1	£99.95

NORTH STAR COMPUTER PRODUCTS



NEW HORIZON-I dual density	£1330
Additional mini floppy	£325
Additional 16K static memory 4MHZ	£315
Printer interface	£70

Extended BASIC and DOS included in above

Double Density Disks now available separately	
CP/M Horizon	£130

NEW SYSTEMS

ITT 2020 EX STOCK

16K RAM integer basic	£950
32K RAM integer basic	£1114
48K RAM integer basic	£1278
DISK II floppy subsystem	£425
EXIDY SOCERER 32K	£850
S100 expansion unit	£275
Disk Drive & Control Card	£699
Nascom-1 kit	£165.00
Buffer board	£32.50
32k RAM	£200.00
16k RAM	£140.00
8k RAM	£85.00
P.S. Unit	£24.50

Astounding Colour Graphics

AN AMAZING SYSTEM BUT A COMPUTER IN ITS OWN RIGHT! DISK DRIVE BUILT IN!

MAIL ORDER

Please send me ....
....
....
(if preferred, attach order on separate sheet to coupon)
Name ....
Address ....
....
....

☐ I enclose cheque for £ ....
☐ Please debit my Diners Club/American Express/Access/Barclaycard Account no. ....

Please add 8% VAT to all items except books

The Byte Shop Ltd
426-428 Cranbrook Road, Gants Hill, Ilford Essex. Tel: (01) 554 2177

All prices correct at time of going to Press

VISIT US AT: LONDON — 48 TOTTENHAM COURT ROAD. (01) 636 0647
BIRMINGHAM — 94-96 HURST ST., BIRMINGHAM NOTTINGHAM — 92A UPPER PARLIAMENT ST. 0602-40576 OPENING SHORTLY AT: LEEDS, MANCHESTER, GLASGOW.

CT-82 Terminal

- ★ Software function controls
- ★ 56-key "Cherry" keyboard
- ★ 12 key numeric or cursor control pad
- ★ 128 control functions
- ★ Graphics capability
- ★ User programmeable character sets
- ★ Software selectable Baud rates (50-38, 400)

£550 + VAT

The CT-82 has 128 different functions all of which are software controlled. Too numerous to list in their entirety they include 18 cursor control commands, 6 ways to configure the cursor, 11 erase functions and graphics capability including point-to-point plotting, field protection, printer pass through, paged edit mode and light pen position reading.

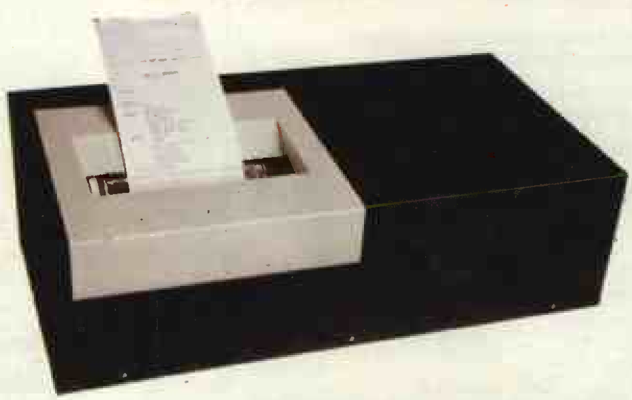
Computer Workshop 3rd Anniversary Offers

4K Memory board (Can be used with Motorola D2 kits as well as SWTPC)	Kit **£45.00** Ass. **£50.00**
Tubes of 25 x 2102-L RAM	**£20.00**
6575 Character Generator	**£4.00**
Set of 20 male and female Molex 10 way connectors	**£6.00**
MP-A processor board containing 6800 MICBUG system, clock and 128 byte scratchpad RAM	Ass. **£50.00**

All prices include postage, packing and VAT.

All offers subject to availability. Please send cheque with order. Goods or full refund by return.

PR-40 Matrix Printer



This unit can be used with any computer with a parallel printer interface and may be ordered with a special interface for the PET system.

The PR-40 uses an 8 bit parallel interface, with data strobe and acknowledge.

This unit prints the 64 character upper case ASCII set on a 40 character line at a print speed of 55 lines per minute. It uses an impact dot matrix head and prints on standard adding machine paper. It is ideal for development work and applications where a 40 column report is sufficient.

PR-40A (25 way RS connector)	**£250** + VAT
PR-40P (PET connector)	**£250** + VAT

SWTPc
Southwest Technical Products Co.

CW
Computer Workshop

38 DOVER STREET · LONDON · W1X 3RB · Telephone: 01-491 7507 · Telex: 268913