commodore

COMPUTING

international

August 1982 £1.00

The independent magazine for Commodore computer users

VICAID

Two sets of Fabulous Utilities in one!

PROGRAMMERS TOOLKIT
Gives extra commands: Auto, Number, Help, Delete, Change, Trace, Step, Light Pen, Break etc.
and
MACHINE CODE MONITOR
Gives Save, Memory Display, Load, Verify etc.
Similar to TIM on PET.

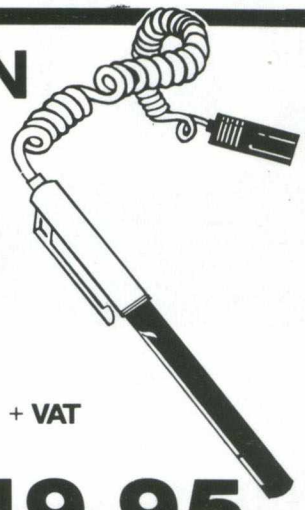
Examine the VICS ROM

Needs DAMS RAM/ROM board or similar

£19.95 + VAT

Business Computers Ltd.
Tel: 051-548 7111 20 lines

LIGHT PEN



VIC LIGHT PEN

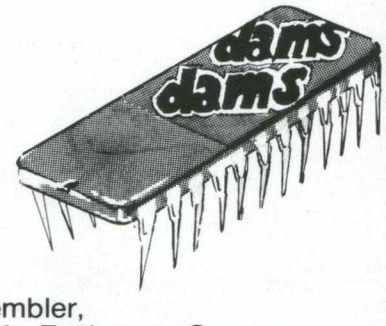
DAMS PRICE ONLY

£17.35 + VAT

FOR PET 12" SCREEN **£19.95** + VAT

VICMON

THE ULTIMATE PROGRAMING AID FOR THE VIC

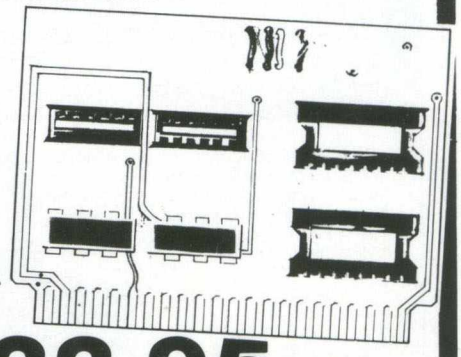


Full machine code package with:
Assembler, Dissassembler,
Fill, Re-locate, Identify, Exchange, Compare,
Printing, Dissassembler etc., etc.

Needs DAMS RAM/ROM board or similar

£19.95 + VAT

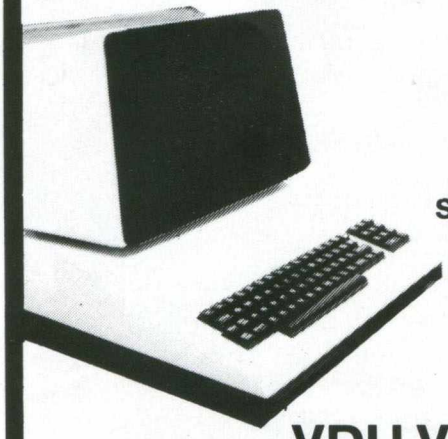
RAM 'N ROM BOARD



3K RAM
In Hires area.
Also space for VICAID and VICMON programming aids.

£22.95 +VAT
(Includes Cover)

BUY THE 3K RAM 'N ROM BOARD WITH VICAID AND VICMON WITH MACHINE CODE MANUAL (WORTH £5.00) FROM MOS TECHNOLOGY FOR ONLY £67.85 + VAT AND GET A FREE VIC LIGHT PEN (WORTH £17.35)
VIC REFERENCE GUIDE R.R.P. £14.95 DAMS PRICE £14.50



VIEW THRU

ANTIGLARE SCREENS FOR PET

40 Column
£17.95
80 Column
£19.95

ALL PRICES PLUS VAT

VDU VIEW THRU

VIC STARTER KIT

VIC 20
C2N Cassette Deck, 10 Blank Cassettes,
User Manual, Vic Programmers Reference Guide,
1 Joystick.
Worth £238.30 ONLY **£214.00** +VAT

(VAT INCL. PRICE = £244.62)

OR VIC 20
With free 3K RAM pack or
Super Cartridge Game **ONLY** **£173.83**
+ VAT

ACCESS & BARCLAYCARD WELCOME

DAMS BUSINESS COMPUTERS LTD.

Gores Road, Kirkby Industrial Estate, Kirkby, Liverpool L33 7UA.
Telephone: 051-548 7111 (20 lines)

Contents

4 ..... **LETTERS** — *Our readers write in with queries, news and views*
5 ..... **NEW PRODUCT NEWS** — *All the latest on the Pet and Vic front*
9 ..... **CLUB NEWS** — *Superscript — history in the making?*
10 ..... **EDUCATION** — *We cover a number of suppliers of educational hardware and software*
14 ..... **LANGUAGES** — *Continuing our dive into Comal*
20 ..... **SOFTWARE REVIEW** — *New educational software for the Vic*
22 ..... **HARDWARE REVIEW** — *Hewlett Packard 7470A Graphics Plotter*
24 ..... **BOOK REVIEWS** — *Structured Programming with Comal, and Pet Interfacing*
25 ..... **GUEST EXPERT** — *Jim Butterfield talking about compilers*
27 ..... **APPLICATIONS STORY** — *Glass Teletype machine code listing*
31 ..... **INTERFACING** — *Dump-Mate, a system for multi-linking of Pets or Vics*
33 ..... **SOUND 'N' VISION** — *CB2 interface, and graphics drawing on the Pet*
36 ..... **PROGRAMMING TIPS** — *Sound Synthesis on the Pet*
43 ..... **BASIC PROGRAMMING** — *Input and Output from the User Port, plus a number of Basic listings*
49 ..... **MACHINE CODE PROGRAMMING** — *System — How to use a SYS call with parameters*

Editorial

Editor
Pete Gerrard

Advertising Manager
Peter Chandler: tel 01-439 3537

Editorial Assistant
Fiona McCormick

Production
Three's Company

Managing Editor
Nick Hampshire

Commodore Computing is published 10 times per year by Nick Hampshire Publications. It is not in any way connected with Commodore Business Machines U.K. Ltd.

Typesetting by Centrepoint Typesetters Ltd, London
Printed by Edwin Snell printers, Yeovil, England.

If you would like to contribute to Commodore Computing, please send articles or programs to:—

Commodore Computing
193 Wardour Street
London WC2

We will pay 10 pounds for each program printed, and 20 pounds for each article published, which should be approximately 1,000 words long.

A magazine such as this cannot afford to be editorially biased in favour of anyone, let alone the company we concern ourselves with all the time, namely Commodore.

If it is true that our news and reviews reflect only Commodore hardware and software, which it is, then this is purely because of the nature of the magazine. We are specialists, in a specialised field. Again, many of the new products that we've covered have been produced by Commodore themselves, or at least marketed by them. Hardware such as the new range of products from the Vic 10 up to the 720 series: software such as the Manager and PetSpeed.

There is good reason for this. The new hardware currently undergoing introduction looks very promising, and should maintain the market position of the company for a long while to come. Software items are covered because they offer particularly relevant features in the current state of the computer industry. If many of these are distributed by Commodore, well, all credit to them for acquiring the product in the first place.

Many of our reviews are favourable. We feel it only fair to bring reasonable products (with due criticism where necessary) to your attention. However, this is not always the case. Where a product is sufficiently 'bad' that to acquire it would be a sheer waste of money, again we think it is fair to point this out. If we subsequently receive a strongly worded letter of complaint from the supplier, which has happened, we are here to weather that particular storm.

Letters

Dear Sirs,

Can you please forward me details as to how I can obtain back issues of the magazine ''Commodore Computing'', and under any of its previous titles. I do currently have a magazine subscription with you.

Thanking you in advance for your kind co-operation in this matter, and looking forward to your earliest reply.

Yours sincerely
M. Preece
West Midlands

Dear Mr. Preece,
This is a query that occurs often enough for it to warrant mention here. Commodore Computing made its debut with the April/May issue, and since then you'll have received issues for both June and July. There was an advertisement placed for a March issue of the magazine, but due to the proverbial 'circumstances beyond our control' this never actually appeared. Consequently, the only back issue we hold are for the April/May, June and July issues.

Earlier that that, and you'll have to go back to Commodore themselves. The magazine was previously called Commodore Club News, and if you addres your query to Margaret Gulliford, at 675 Ajax Avenue, Trading Estate, Slough, Berkshire, I'm sure you'll be able to find the solution to your problem, as Commodore keep telling us.

Dear Sir,

I read with interest the Club News section of the new Commodore Computing magazine. This college has recently set up a CBM User Group to act as a centre for the increasing number of people interested in using Commodore systems particularly for business and educational purposes. We serve largely the North Humberside region and part of North Yorkshire also. I would be obliged if you could tell me whether there is any form of central support or advisory service offered by Commodore or any other source for groups such as ours.

We have a number of CBM computers of various models supported by a number of disc drives, printers and commercial as well as 'home produced' software. We would welcome enquiries concerning our activities from anyone interested and the person to contact is myself at this college.

The college is actively involved in a range of computing and electronics courses at all levels and we do have available 'educational' and 'business' workshops for those who require them.

Yours sincerely
J. L. Stephenson
Head of Computing Section
East Yorkshire College
of Further Education
West Street
Bridlington
North Humberside
YO15 3EA

Dear Mr. Stephenson

Thanks for an informative letter, and news of your new club. If any of our readers are interested, I'm sure Mr. Stephenson would welcome any enquiries.

To answer your query on any support and advice from Commodore, since we took over the magazine the Pet User Club effectively died a death, and Commodore are now quite happy to let ICPUG (Independent Commodore Products Users Group) handle everything. I would strongly recommend getting in touch with them, as they can provide valuable assistance in the early days of setting up and getting the whole show rolling.

Also, as ICPUG have an extremely good 'rapport' with Commodore, they're in a fairly good position to provide the kind of information that can only come from knowing the company well: new product news, advance warning of any major changes coming up, and so on.

Finally, however technically skilled your own people might be, within the auspices of ICPUG are such well known names as Harry Broomhall, Mike Todd, et al., who can usually provide the answers to just about any question that may arise.

The person to contact for more information is Mrs. Eli Pamphlett, at 7 Lower Green, Tewyn, Welwyn, Herts.

Dear Sirs,

I have developed a small program, which may be of interest to your readers with an 8032.

After searching in vain through back issues of all my computer magazines for a short, effective machine code routine for string/substring search and match I decided to tackle the problem myself. The listing below was the result.

The routine tests a main string (S2$) for the presence of a substring (S1$), placing the result in 32767. A result of 0 shows that no match was found. A positive number result indicates that the substring was found in the main string starting at character peek (32767) of the main string. The match takes about 0.01 second and the m/c occupies about 130 bytes once located.

I would be happy to answer any questions from readers on this program.

Yours sincerely
Ron E. Eagle
2 Rusking Crescent
South Shields
Tyne and Wear NE34 9HU

Dear Ron,

Thank you for the letter, and the listing, which we show below. As usual, if any readers come up with an alternative version for the other series of Commodore machines, we'd be delighted to hear from you.

Dear Sirs,

Our school owns a Vic, but unfortunately we cannot afford to buy a printer yet. What I want to do is print out listings of my programs on a Pet printer, but whenever I save a program on tape, then load it into the Pet, although the Pet says 'Searching', 'Loading', and then 'Ready', with the flashing cursor, the Vic program doesn't appear to be there. Am I doing something drastically wrong, or is there a simple solution?

Yours hopefully
John Ives
Birmingham

Dear John,

The problem is essentially that the Vic's internal memory is stored differently from that of the Pet, and this problem is heightened by the fact that the Vic's own memory moves around as you add various memory expansion cartridges. Consequently, one or two Pokes have to be performed before you can convince the Pet that the program is actually there.

On a Vic with no expansion, once the program is loaded into the Pet you need to:—
POKE 4096,0:POKE 41,16:CLR
On a Vic with 3K expansion, no changes are required.
One a Vic with more than 3K expansion, you need to:—
POKE 40,1:POKE 41,18:POKE 4608,0:CLR

Once you've done that you will be able to list the program on the screen, save it to disk, or list it out on the printer. I don't suggest you try and do anything else with it, as Vic programs are not likely to be particularly compatible: the various screen and colour codes may well cause strange things to happen when the program is run.

STRING MATCHING ROUTINE

PROGRAM TO LOCATE M/C @ TOP OF MEMORY
OR LOCATE M/C IN 1ST CASSETTE BUFFER.

```
1 A = 32000 : B = 32119 : REM [OR REPLACE WITH 650 & 769 FOR BUFFER LOCATION]
2 DATA 32,160,189,160,0,177,68,133,177,200,177,68,133,181
3 DATA 32,160,189,160,0,177,68,133,178,200,177,68,133,182,200,177,68,133,183
4 DATA 169,0,141,255,127,141,252,127,162,0,160,0,177,180,209,182,208,1,232,200
5 DATA 196,177,208,244,228,177,240,31,24,173,252,127,105,1,141,252,127,173
6 DATA 252,127,197,179,208,5,162,0,76,97,125,24,165,182,105,1,133,182,76,46
7 DATA 125,224,0,208,8,169,0,141,255,127,76,118,125,24,173,252,127,105,1,141
8 DATA 255,127,96,170
9 FOR I = A TO B : READDA : POKE I , DA : NEXT
10 POKE 53 , 125 : REM [MISS THIS LINE OFF IF USING BUFFER LOCATION]
```

EXPLANATION OF THE PROGRAM

```
20 REM ************************************************************
21 REM ** THIS SUB ROUTINE WILL SEARCH A MAIN STRING FOR A SUBSTRING  **
22 REM ** MATCH AND PLACE THE RESULT IN LOCATION 32767.               **
23 REM **                                                            **
24 REM **  1.   TO  PLACE MACHINE CODE ROUTINE AT TOP OF MEMORY      **
25 REM **                                                            **
26 REM ** COPY BASIC DATA PROGRAM FROM LINES 1 - 10. RUN THE PROGRAM. **
27 REM ** TYPE NEW . YOU NOW HAVE THE ROUTINE AT TOP OF MEMORY & CAN  **
28 REM ** WRITE OR LOAD A BASIC PROGRAM WHICH USES THE ROUTINE.       **
29 REM **                                                            **
30 REM **  2. TO PLACE THE MACHINE CODE ROUTINE IN CASSETTE BUFFER    **
31 REM **                                                            **
32 REM **   IF YOU PREFER TO LEAVE ALL BASIC MEMORY FREE THIS METHOD  **
33 REM ** LOCATES THE M/C IN CASSETTE BUFFER...LEAVING BASIC MEMORY   **
34 REM ** UNAFFECTED.(CAUTION...THIS MEANS YOU CAN'T USE CASSETTES).  **
35 REM **   FIRST REPLACE VALUES OF A & B AT LINE 1 WITH 650 & 769    **
36 REM ** RESPECTIVELY. THEN DELETE LINE 10. NOW PROCEED AS ABOVE     **
37 REM ** AND THE M/C WILL NOW RESIDE IN BUFFER.                     **
38 REM **                                                            **
39 REM **                    SET UP VARIABLES                        **
40 REM **                                                            **
41 REM ** S2$....MAIN STRING WHICH YOU ARE TESTING AGAINST.          **
42 REM ** S1$....SUB STRING WHICH YOU ARE TRYING TO FIND IN S2$      **
43 REM ** RE.....PEEK(32767).....RESULT IS POKED INTO 32767         **
44 REM **                                                            **
45 REM **                    TO CALL ROUTINE                         **
46 REM **                                                            **
47 REM ** USE SYS 32000,S1$,S2$ OR SYS 650,S1$,S2$ (DEPENDS ON LOC.) **
48 REM **                                                            **
49 REM **                    TO TEST RESULT                          **
50 REM **                                                            **
51 REM **   IF THE VALUE OF RE IS 0 THEN NO MATCH WAS FOUND.          **
52 REM **   IF A MATCH WAS FOUND THEN THE VALUE OF RE POINTS TO THE   **
53 REM ** CHARACTER POSITION IN THE MAIN STRING WHERE THE MATCHING    **
54 REM ** SUBSTRING STARTS.                                          **
55 REM **                                                            **
56 REM ** E.G.  S2$="COMPUTERS" S1$="PUT" GIVES RE VALUE OF 4.       **
57 REM **                                                            **
58 REM ************************************************************
```

New Product News

More Training Courses

Training courses abound these days. One University who have just announced a series is the University of Salford, and in particular the Department of Electronic and Electrical Engineering there. They're holding 4 courses in all, in September of this year, covering the Pet for beginners, getting more from your Pet, the Pet in control, and an introduction to Pet machine code programming. Prices for the courses range from 95 pounds for the first two, which are two day courses, to 55 pounds for the latter two (both one day courses).

If you want to know more, the person to speak to is Mrs. S.R. Hill, on 061-736 5843 extension 248.

Of course, there is also Commodore's own training courses, recently taken over by McDowell Knaggs Associates, and being held in Manchester and Worcester, as well as the traditional Heathrow site. The four areas that these cover are Basic for beginners, disk file programming, program planning, and assembler language.

The cost of these three day events is 210 pounds, and this covers tuition, documentation, lunch and refreshments. The emphasis is very much on 'hands-on' experience, and if you wish to know more just ring Worcester 28466.

Whilst still on the subject of training, a novel approach has been adopted by Adda, one of the larger Commodore dealers. Although the fees for their courses are quite high, at the end of the day you take home the equipment you were working on. Based around the Vic 20, and intended primarily for businessmen, further details can be found by ringing 01-579 5845.

INSTALLATION SERVICE

Experience is often very valuable when installing your new Commodore System. Mistakes are frequently very costly and waste valuable time.

ONE DAY SERVICE
For a fee of £85 per day plus expenses a member of staff will help you overcome early difficulties and set you on a suitable path to a successful computerisation.

FULL INSTALLATION SERVICE
This is tailored to your requirements. We can supply extra operations staff, or technical advisors. Extra equipment can be useful if an installation is required by a certain date. As full Commodore Systems Distributors we have experience you will probably need.

MAINTENANCE
Most of the system breakdowns are not hardware faults, but consist of lack of understanding of programs or faults based upon unwise practices. Our staff are trained to assist with system problems, and they are capable of finding the best possible solutions. Maintenance staff will visit your site, diagnose your difficulty and if necessary replace any components needed.

For information concerning this service please contact Brian Homewood or Robert Jones.

PEACH DATA SERVICES LTD.
COMPUTER SERVICE TO BUSINESS
5 HORNINGLOW STREET, BURTON-ON-TRENT, STAFFS. BURTON (0283) 44968

KINGSLEY COMPUTERS LTD.
132 Desborough Road
HIGH WYCOMBE, BUCKS HP11 2PU

CBM BUSINESS SYSTEMS

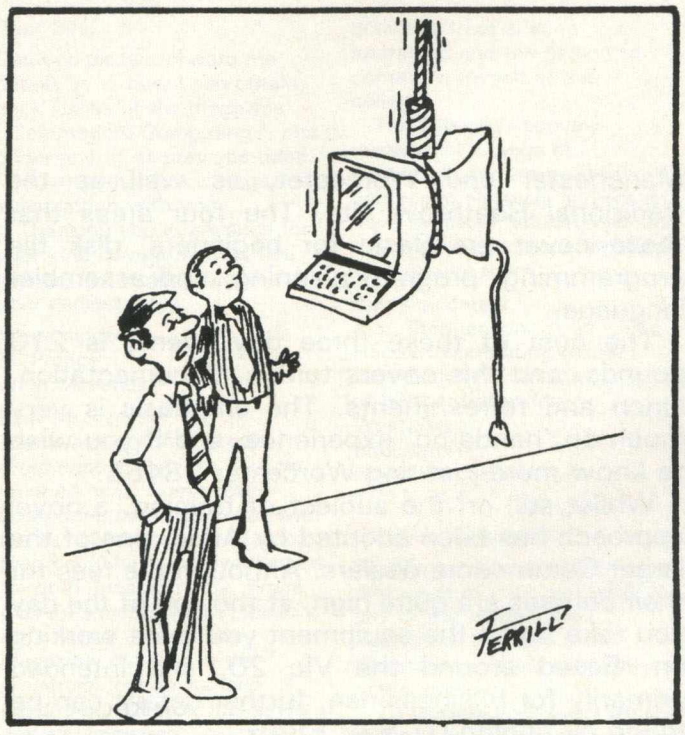
VIC HOME COMPUTERS

COMPUTER ACCESSORIES AND SUPPLIES

AGENTS FOR CBM APPROVED PRODUCTS

COMMODORE SERVICE CENTRE

New Product News



'Obviously, the Work of an Irate User.'

New Books on the Scene

Everyone's talking about COMAL these days, so perhaps it is only fitting that a number of books are now beginning to spring up concerning the subject. First off the mark was Structured Programming with Comal, by Roy Atherton, and published by Ellis Horwood Ltd. Essentially a comparison between Basic and Comal, the book does move onto the subject its title implies it will. An interesting read, and if you're at all attracted by Comal, definitely worth looking into.

Another new arrival is Beginnng Comal (again from Ellis Horwood) by Borge Christensen, founder father and leading light behind the language. Assuming no prior knowledge, it is basically (if you'll excuse the word) a textbook on the language, and indeed is being used as just that in Denmark, where Comal appears to have made its biggest mark. It contains many, many listings, and not just irrelevant four liners either: more like a complete library of Comal programs.

Our final news on Comal is the launch of the Comal bulletin, a bi-monthly publication which promises to deal with "concepts, applications, standards, teaching and implementations". Once more, Ellis Horwood are the people to contact to find out more.

Word Processing on the Vic

There comes a point, with certain types of software, when you have to ask the question "is it a viable proposition?". Word processing on the Vic must surely fall into that category: scrolling screens are all right, but when you can only see 22 characters at a time, it becomes a bit difficult to remember what you typed at the end of the last paragraph.

Still, if you want to do word processing, then the best package to appear so far has to be Wordcraft 20, distributed by Audiogenic in Reading (Tel. 0734 586334). Written by the author of the original Pet version, it has just about all the features of that program, and also incorporates some of the Vics own specialities, such as sound and colour, for extra user feedback. Coming in the form of a plus in cartridge, and priced at 125 pounds, it looks rather impressive. Next month we'll carry a detailed review of the package.

New Vic Games

As well as producting Wordcraft, Audiogenic have recently also brought out a whole host of Vic programs, mainly games, but a data base and a 'toolkit'-like program have reared their heads as well. I don't know who writes the catalogue descriptions for the games, but some of them are quite . . . well, ludicrous! Take this one, from a game called Cloudburst: "Save the Earth from the downpour of Acid Raindrops and the invasion of the mutant Cloud Hoppers!". Almost makes you afraid to go anywhere near the game.

The star of the show though has got to be Renaissance, otherwise known to us as Othello. This is one of the best games I've seen on the Vic. You can change sides and playing level, take back moves, set up special games, save games to tape, read them back later, and all with excellent graphics. Even the lowest level (there are 8 in all) plays an excellent game.

More Vic Product News

Computer World in Holland recently announced the introduction of their TDK 20, essentially a ham interface for the Vic.

Costing 89 pounds ex. VAT, this comes in a cartridge that simply plugs into the back of the Vic, or onto an existing expansion board. It includes a RTTY and morse code convertor, and can be attached to various devices to turn it into a message memory keyboard, to start decoding either RTTY or CW, or transmitting RTTY. All told, a nifty little unit.

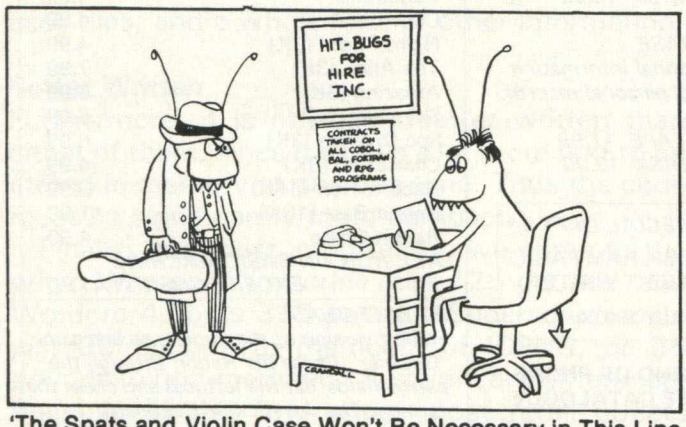
A really exciting one next, and something the

Vic world has been waiting for. As you know, the Commodore disk drive for the Vic is a single unit, which means that making backup copies of disks is somewhat difficult, and quite expensive at 396 pounds. Well, B.G.C.C. down in Brighton (Tel. Brighton 507694) have come up with a dual floppy cartridge drive for the Vic. It is a 3 inch microfloppy system, and should retail for around 395 pounds when it makes its first appearance at the end of September. We hope to carry a full review of this as soon as possible.

Petpourri

A whole host of Pet related software and hardware has appeared over the last couple of months. J.J. Lloyd (Tel. 04895 4221) have brought out a low cost X-Y digital plotter for the Pet, and since it is addressed via the IEEE bus one must assume therefore that it will work with any models in the range. Known as the PD4, it handles A4 size papaer, and you can acquire from J.J. Lloyd a software package in ROM form, which eliminates mucho effort on your part in writing various software routines to drive the beast. Although only one pen is used at a time, it has a compensatory high speed of 600 mm/sec, and at a price of 596 pounds plus VAT it puts this particular plotter firmly in the reach of most engineering, industrial and scientific users, which is where it will probably be of most use.

Netkit II from Yorkshire Microcomputers (Tel. 0723 78136) converts your Pet into a terminal, with the ability to communicate with mini and mainframe computers, telex equipment, hand held data capture terminals, and many other



'The Spats and Violin Case Won't Be Necessary in This Line of Work, Perkins.'

New Product News

scientific and industrial devices. Based on the original Netkit, but significantly upgraded, the entire package comes on board complete with 4K Eprom's worth of software, and fits quite easily into the Pet.

Another from our overseas friends this time, Datatronic in Sweden. A program called Calc Result, which appears to be a significant improvement on the original program Visicalc. The work matrix size is 32 times as big, and it is a lot easier to use. All in all a very smart package. Try, if you can, to get hold of the quaintly worded leaflet from Datatronic. Their programming is an awful lot better than their English!

InfoPro

Professional Software (Tel. 0707 42184) of Wordpro distribution fame, have announced a new product called Inforpro, which they describe as an Information Management System. Essentially another data base, it does have the ability to link with any of the Wordpro family of programs, and indeed can accept information of any sort as long as it has been filed sequentially. It works on either the 8032 or 8096, with a 4040 or 8050 disk drive.

Finally for this month, a new system for indexing on the Pet from Farestead Associates (Tel. 0442 51708). Although they've been running an in-house indexing service for some time now, they're unleashing their program Microindex onto the market at the end of August. Claims made include taking six hours plus proof reading to enter an 800 entry index, and a cost of between 50 to 60 pounds to compile a 750 index.

Nothing to do with New Products, but ....

SPACE INVADERS

```
$0E01 (3585) - sets speed you move and fire ; normally 2
$0E09 (3593) - sets invader firing speed    ; normally 4
$0404 (3598) - sets mother ship speed       ; normally 6

$0623 (1571) contains character after missiles to erase
them.  Try POKEing with 102 for Wall Invaders.
```

80/40 COLUMN TEST

```
WD=80 : POKE 32768+1024, 96  :  IF  PEEK(32767)=96  THEN
WD=40
```

Where WD is the width of the screen. Thanks Jim!

CREAM COMPUTERS

380 STATION ROAD, HARROW, MIDDX. HA1 2DE
Telephone: 01-863 0833

HARDWARE

VIC - 20 Computer (Call now for new low low price)	
C2N Cassette Unit	39.99
1540 Floppy Unit	375.00
1515 Printer	214.95
Memory Expansion	110.75
3K Ram Expansion	27.50
8K Ram Expansion	39.98
16K Ram Expansion	68.99
Super Expander Cart	32.50
Progs Aid Cartridge	34.50
Vicmon Cartridge	30.00
Joystick	6.99
3K Store Board	51.00
8K Ram Chips	39.00
Vickit I	25.00
Vickit II	29.00
VIC - 20 Dust Cover	2.00
C2N Dust Cover	1.70
Light Pen	25.00
Rom Switch Board	42.00
Expansion Board	94.99

*NEW*NEW*NEW*NEW*NEW*NEW*NEW*
PRINTERS FOR YOUR VIC - 20

SEIKOSHA PRINTER –
The Commodore Printer plugs directly into the VIC - 20 uses 8" continuous paper up to 80 chars per line with a speed of 30 c.p.s. 214.95

EPSON MX80 F/T PRINTER –
Uses standard A4 paper or 9.5" paper. 80 characters per line with a speed of 80 c.p.s. (INC. VIC - 20 Interface) 458.99

'DISCOUNT PRICES ON QUALITY PRODUCTS'
THE VIC - 20 CENTRE

****** NEW ******

RABBIT WRITER (Wordprocessor)
Requires 8K or 16K Ram,
*includes - *mailmerge**
left and right hand justification
tape and disk format
prints up to 80 columns
uses 80 columns
uses all function keys for editing etc.
INTRO PRICE TAPE 9.99
DISK 12.50

RABBIT BASE
Requires 16K RAM Personal information master, ideal for mailing, personal records, stock control and filing.
INTRO PRICE TAPE 11.99
DISK 13.99

SPECIAL OFFER* *SPECIAL OFFER
VIC - 20 PRINTER + 16K RAM PACK + RABBIT BASE + RABBIT WRITER
ALL THIS FOR ONLY 290.00

DON'T FORGET TO SEND OR PHONE FOR YOUR FULL FREE CATALOGUE

RABBIT SOFTWARE

Space Storm	6.99
Ski Run	4.99
Dune Buggy	4.99
Super Worm	4.99
Jungle	4.99
Cosmic Battle	4.99
Frogger	7.99
Rab Functions	4.99
Star Wars	6.99
Amok	6.99
Blitz	4.99
Invader Fall	6.99
Masterwits	6.99
Kiddie Checkers	6.99
Simple Simon	6.99
Alien Blitz	7.99
Functions	4.99
Code Breaker	4.99
Night Flight (3K)	4.99
The Alien (3K)	7.99
Amazing (3K)	6.99
Skymath (3K)	6.99
Spc Division (3K)	6.99
Charset 20 (3K)	4.99
Rabbitwriter (16K)	9.99
Rabbit Base (16K)	11.99
Rabbit Chase	4.99

ALL RUN ON BASIC MACHINE EXCEPT WHEN STATED

RABBIT CHASE
Rabbit version of the famous arcade game 'Gobbler'. Chase the Rabbit through the carrot fields, eat the lettuces and chase the farmers. Use Joystick or Keyboard.
only 4.99

ALL PRICES ARE INCLUSIVE OF VAT, POSTAGE & PACKING.
FOR FURTHER DETAILS ON ALL PRODUCTS, PLEASE CALL OR WRITE TO THE ABOVE ADDRESS.

Club News

Pieces of Eight . . .?

This month's club news was going to be about the Watford group of ICPUG, a look at what they're doing, and a report on a club night with them. However, events are such that now we have to turn our attention elsewhere (sorry Stephen: maybe next month), and take a look at events that are potentially of great importance for the computer industry. It's the old story of the big guys against the little guys.

Not too long ago, but long enough to be of significance, one of the Independent groups (IPUG South East) produced a word processing program of astonishing versatility. The author, Simon Tranmer, who works as Technical Support for General Automation, apparently wrote the program because his wife wanted to send a letter to a long lost aunt in Australia, and he decided that what she needed was a word processor. So, he wrote one.

Other Packages

Presumably, Simon had seen the two major packages already in existence for the Pet, namely Wordpro and Wordcraft. Both have their distinctive features, but both have their failings as well. What one will do the other will not, and vice versa. Equally, there are a number of functions which neither of them will do. Rather like taking a shot of gin and a bottle of tonic, the ideal would be to blend to two together, and add a slice of lemon to set the whole thing off.

This is what Simon did. The program, Superscript, combines the best of Wordpro with the best of Wordcraft, and has a number of its own very powerful commands. It is extremely easy to use, it will read files created from either Wordpro or Wordcraft, will read program listings, asoii files, and a whole host of other information.

Better Written

Furthermore, it is obviously better written than either of them, since it allows a lot more text to be stored in memory at any one time. Thus the code must be significantly more compact.

Finally, and most importantly, we come to the price. Whereas Wordcraft cost 425 pounds, and Wordpro 4 costs 395 pounds, Superscript costs just 30 pounds to members of ICPUG, or 35 pounds to non-members. A significant difference! Because of this low price, most retail outlets would probably not take the package: not enough profit in it for them. Consequently it was being sold via mail order from an address in London.

You will note that I said 'was' being sold from an address in London. It is NOT being sold at the time of going to press, pending legal action. A company who, as they say 'shall remain nameless', have taken out an injunction to prevent Superscript continuing its trail blazing path.

Rip-Off?

This brings us to the importance of the whole issue. Why should anyone take out an injunction? Software piracy perhaps? To say that Superscript is 'ripping-off' another program is the same as saying that a Ford Cortina is a rip-off of a Sherman tank. They both perform the same functions i.e. they get you from A to B, they can go round corners, they have steering wheels etc., but they can hardly be compared.

Fear of competition then? No-one who is in the computer industry can afford not to be afraid of competition, but you do not immediately involve the law because you're afraid of losing business.

Is that the problem, fear of losing business? It would seem immediately obvious that a program selling at thirty pounds which is better than a program selling at round about 400, is going to sell a lot more copies. Through the medium of the ICPUG newsletters punters were going to be kept informed of any changes on the Superscript front, and the backup service normally performed by dealers would have been dealt with that way.

Conclusion

If it is the case that Superscript can not be sold through fear of competition, or indeed for whatever reason someone's decided to take out an injunction, it is a great shame and an indictement on the state of the computer add-on industry. In many cases, prices of software packages are being kept artificially high: if Superscript can do it, why can't everyone else?

Let's hope that this injunction is dropped, and that Superscript re-assumes its place on the market front. By the time you read this, that might already be the case. Will software people take heed and lower their prices? Time alone will tell.

Next month we'll hopefully be back to normal again!

If you have any news of your own user club, details of activities, meetings etc., I'd be grateful to hear from you. Just drop me a line at the address on the masthead.

Education

Educational Software

One of the reasons why the Pet and Vic have made such extensive inroads into the educational world is the amount of good quality software readily available: a computer is only as good as the software that runs on it. Here we take a brief look at just some of the many suppliers around the country.

School Software

Some of these suppliers are in fact schools in their own right, which gives you a fairly good guarantee that the software will be of a high standard, as you can safely assume that it has already been field tested in the school prior to release. One such is the Houghton County Primary School down in Cambridgeshire.

Their programs are, not surprisingly, aimed mainly at the primary school level, and cover such topics as word matching, sentence value, getting to grips with early numeracy skills, and so on. Software for this level is possibly the most difficult to write: children of this age group tend to get tired of doing the same thing very quickly, and so programs have to be carefully structured in order to avoid this.

Happily, Houghton's programs cater for this, and are worth exploring further. They can be reached at Houghton, Huntingdon, Cambridgeshire.

Specialisation

Then there are the more specialised companies, like ESM (Educational Software for Microcomputers). They produce a whole range of programs, covering many topics commonly encountered in the school curriculum. Although principally involved with literary and numeric skills, there are a number of interesting programs covering the biology angle. These latter (and indeed, most of the others) are written by one Russell Wills (with various accomplices), who was the mainstay behind the educational side of Commodore's old PetPack series of cassette programs.

ESM state that the programs can cover both primary and secondary schools, but a read through of their catalogue (and a knowledge of many of the programs) leaves one with the distinct impression that they would be of most use at a secondary level. Either way, further information can be gleaned by telephoning 0945-63444.

Garland Computing are another company who 'specialize in programs for education'. Their latest handout gives details of biological programs, with the promise of animated graphics. Happily, the subjects they cover include nothing more potentially dangerous than Animal Physiology, Plant Physiology, and so on. For full details and a current price list, you're invited to ring 0752-41287.

Inexpensive Companies

A third source of good software are a host of companies who mainly produce general software for all kinds of purposes, but who in turn have quite a number of educational packages, usually inexpensively priced.

To name but a few, Pedro Computer Services (01-250 1481) supply a low cost Pet to TV interface, useful when giving demonstrations in the class, where not everyone can cluster around the Pet screen, and also a CB2 soundbox.

Qwerty Computer Services (0385-67045) supply about 15 million software and hardware add-ons, most of them at a reasonable price. Many of these are suited to education, and any school teacher who's budget is in danger of crumbling ought to consider getting in touch with them.

Simple Software Ltd (0273-504879), have an annoyingly 'Gang Show' type catalogue, but this must not detract from what is a quite acceptable selection of software and hardware. There's not a lot that is specifically aimed at education, but a browse through the items presented should give you an idea or two.

Audiogenic (0734-586334) and Supersoft (01-861 1166) are two well established companies, who have for a long time now been selling software and hardware for the Pet, and lately the Vic as well. Again not too much directly for educational users, although Audiogenic have one or two useful titles in that field. Again, it's interesting to note that a number of Russell Wills' programs are included here: the man certainly gets around.

Conclusion

We haven't listed everyone here: to do so would take up the vast majority of the magazine. In the end it all comes down to the old adage 'you pays your money . . .'. Hopefully we've been able to point you in one or two of the right directions.

We'll continue to explore the educational aspect of Commodore equipment in next month's issue.

£8.95 plus 55p p+p

Very Important Educational Cassettes

A.S.K. announce the first four programs in a series of educational cassettes for the VIC 20. These programs have been written by a team of teachers and professionally programmed specifically for use in the home.

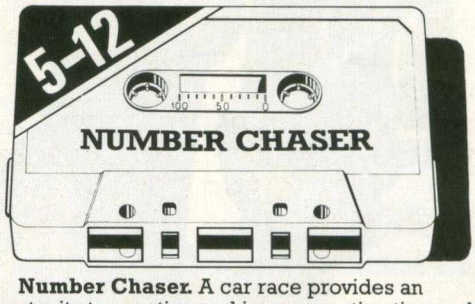
They are of proven educational value, complementing work done at school, yet all the programs are designed to be fun to use – not just once, but over and over again.

We believe that these programs will give you and your family and friends hours of worthwhile enjoyment. They will help your children to learn at home in a relaxed yet stimulating way.

We Want To Count. A program for young children learning to count which involves the numbers 1 to 5. Children often find it easier to recite numbers than to count things correctly. Four different games give the child a variety of objects to count, and are presented in an exciting and stimulating way. Suitable for children aged 3 and upwards.

Twister. A geometric puzzle that will tie you in knots, testing and improving your thinking skills and powers of concentration. The purpose is to rearrange coloured squares so that no row or column contains a repeated colour. Set your own puzzle and test the whole family. Suitable for children aged 8 and upwards.

Facemaker. This program is designed to help improve spelling, expand vocabulary and sharpen observational skills. There are thousands of characterful faces you can make with the program. Perhaps someone you know? Suitable for children aged 5 to 12.



Number Chaser. A car race provides an opportunity to practice and improve estimating and multiplication skills. You can choose the level of difficulty you want making it different every time you play. Suitable for children aged 5 to 12.

Each cassette comes in an attractively labelled box together with a colour booklet which gives detailed loading instructions and tells you how to use the program.

N.B. Because these programs make extensive use of computer memory and colour graphics, a 16K RAM PACK (or 8K RAM PACK for Numberchaser only) and colour T.V. are essential for their operation.

If you do not have a 16K RAM PACK, we will be pleased to supply one at the discounted price of £67.50 with your order for one or more A.S.K. programs.

NO QUIBBLE GUARANTEE

If you are dissatisfied with any A.S.K. program, return it to us within 7 days of delivery and we will give you a full refund without question.

Not convinced? Then see our programs at The Vic Centre, 154 Victoria Road, London W3, opp. North Acton tube.

A.S.K. LIMITED, London House, 42 Upper Richmond Road West, London SW14 8DD

To A.S.K., Freepost, London SW14 8BR (no stamp required)

Please send me:

	Quantity	Unit price inc. VAT + 55p p+p	Total
We Want To Count		£9.50	
Twister		£9.50	
Facemaker		£9.50	
Number Chaser		£9.50	
16K RAM PACK		£67.50	
		TOTAL	

I enclose my Cheque/P.O. for £_____ made payable to A.S.K. LTD

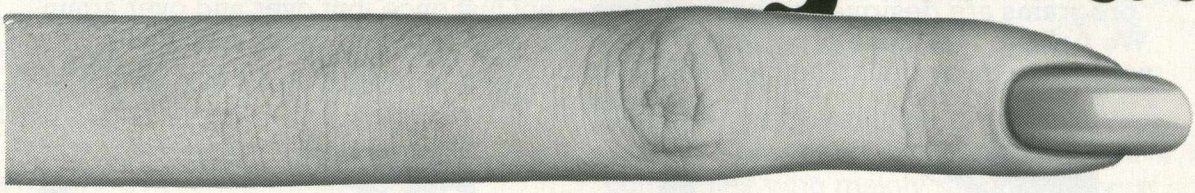
Name _____

Address _____

_____

Please allow 28 days for delivery CC8

Now you can do all accounting with...



without...

the filing, typing and

Silicon Office is the latest microcomputer software program from the Bristol Software Factory.

Designed specifically for use with the Commodore PET 8096, it'll help you run your office with the minimum amount of effort and maximum efficiency.

Think of it like three normal software packages in one, each separate package totally interactive with the other.

For around £4,500, you can have the complete electronic office, the solution to practically all your business problems. The price includes Commodore hardware, a high quality daisy wheel printer and Silicon Office software.

Silicon Office is made up from a flexible information management system which lets you create and maintain an extensive filing arrangement. Allowing you to search quickly through your records, making cross references between files in order to gain the facts you require.

A highly sophisticated word processing program allows you to generate letters, documents and reports. Letting secretaries get on with the more important tasks.

And a fully comprehensive calculator means you can handle all the number crunching you're ever likely to do in a business situation. Leaving the accounts department to concentrate on more profitable things.

But that's not all by any means.

Silicon Office also has a special programmability feature which means you or your dealer can expand and tailor the Silicon Office program to your business.

When Silicon Office is used in an everyday business situation, certain command sequences are inevitably repeated. By writing short, very simple programs which are entered into the computer's memory, Silicon Office can perform the necessary tasks, automatically.

And last, but by no means least is an optional communications facility.

It doesn't take much imagination to see the potential of Silicon Office in virtually any line of business.

So to get a better grasp, send away for our brochure. It'll only cost you a stamp. And it could save you a fortune. Or talk to your local Commodore dealer who has all the facts at his fingertips.

You'll soon see how you're much better off with Silicon Office. Than without.

commodore COMPUTER

I can't wait to get my hands on a free copy of the Silicon Office brochure.

Name_____

Position_____

Company_____

Address_____

_____ Telephone_____

ACC2

I own a Commodore PET (Please tick box) YES ☐ NO ☐

Send to: Bristol Software Factory, PO Box 14, Horley, Surrey.

SILICON OFFICE

Languages

As promised last month, Brian Grainger continues his look at Comal.

11) The command LABEL will cause a label to be placed on a line. For example you can type either of the 2 statements below and the COMAL interpreter will accept it in the same way:
0010 LABEL START or
0010 START:

When listing the program it is the second form that will be listed. We can now use the label START as a reference in a goto statement:
N.B. No other statements can occur on a labelled line.

12) The LIST command under COMAL has 2 special features. Firstly it invokes automatic indenting of lines for readability. It is not necessary to type the lines indented it is done automatically. Secondly one can LIST to a file on the disk or cassette. This file can subsequently be merged into any existing program by the ENTER command:
LIST ''FILENAME'' lists the current program to disk under the name FILENAME.
LIST ''FILENAME'', 1 lists the current program to cassette under the name FILENAME.
LIST 10-50, ''FILENAME'' will list lines 10 to 50 inclusive to disk under the name FILENAME.

13) The RENUM command will remember an entire program. It has the same syntax as the AUTO command.

14) The RUN command is similar to that of BASIC with one notable exception. Variables are NOT reset to zero or null. Thus no assumptions should be made on the values of undeclared variables in a program.

15) In comparing the BASIC PRINT command with the COMAL equivalent. COMAL looks clumsy. In reality COMAL is much more flexible. BASIC has fixed print zones of length 10. In COMAL one can vary the zone length by the ZONE command: e.g. ZONE: = 5 will cause items separated by commas in print lists to be printed in zones of 5-character length.

The default value of ZONE length is 0 and not 10 as in BASIC. Thus a comma will cause no separation of variables until zone length has been redefined. In COMAL a semicolon between print items will ALWAYS print 1 space between those items, whether they are numeric or string. It is a true space unlike BASIC which for numerals will print a cursor right after the value. This appears like a space but in reality is not.

16) In COMAL variable names or labels can be up to 16 characters in length. The first character must be a letter and all 16 are recognised unlike BASIC where only the first 2 are recognised. In COMAL it does not matter if a variable name includes a COMAL keyword. Thus variable names can be chosen which have meaning. One side effect of this is that COMAL keywords must be typed with a following space. Otherwise it will be regarded as a variable name and a syntax error will probably result. One other COMAL restriction on variable names is that the same name CANNOT be used for two different variable types. For example the variables A and A$ cannot appear in the same segment of program.

17) I have mentioned above that one need not type the spaces necessary to cause line indentation. They are added automatically when the program is LISTed. There are other things that need not be typed as well. Here is the complete list:
The ':' preceding the '=' in assignment statements.
The 'OF' in 'CASE . . .OF' statements.
The 'DO' in 'FOR . . .DO' or 'WHILE . . .DO' statements provided the 'DO' is the last word in the statement.
The 'THEN' in 'IF . . .THEN' or 'ELIF . . .THEN' statements provided the 'THEN' is the last word in the statement.
The procedure name following the 'ENDPROC' statement.

In all the above cases the COMAL interpreter will automatically add the relevant words if they have been omitted.

18) The REM' command is accepted in COMAL but the interpreter will replace it with //. REM or // can be used at the end of a COMAL statement or be on a line of its own.

19) It has been stated above that variables cannot be assumed to be null at the start of a program. There is therefore a need for a command which will set a string variable to a blank. The following example will illustrate: SPACE$(1:60):=''''' will set the string SPACE$ of sixty characters length to a blank. Another facility on string handling is that if a single character is to be referenced it can be done simply as the following example shows:
NAME$(5) refers to the 5th character of NAME$. It is not necessary to say NAME$(5:1). Thus when the length value is 1 it may be omitted.

As can be seen from the above COMAL has many features that BASIC does not have. I have

left until last one of the most important features: There is syntax checking on input as there is with ZX80/81 BASIC (wash my mouth out!). The interpreter will even leave the cursor at the point where the error is recognised so that the line can be modified quickly before continuing. This means all the statements of a COMAL program are checked for syntax before it is RUN. In BASIC they are not checked until run time and it is up to the programmer to ensure all the lines are executed and checked. This is not at all easy on long and complicated BASIC programs with many loops and branches. To sum up COMAL is an extremely powerful language compared with BASIC which results in easy to read, easy to maintain programs.

COMAL — BASIC commands without COMAL equivalents

In this article on COMAL I want to do two things. Firstly I want to identify those BASIC commands which are not included in COMAL and do not have COMAL equivalents. Secondly I want to identify some problems I have found in using COMAL (rev 0.11) whether in using the BASIC2 or BASIC4 version.

Here is a list of BASIC commands not supported by COMAL:
BACKUP, CLR, CMD, COLLECT, CONCAT, COPY, GET, HEADER, ON . . .GOTO, RENAME, SCRATCH, VERIFY, WAIT, POS, STR$, TIME, TIME$, USR & VAL.

In addition to the above commands, shortforms are not allowed in COMAL. In particular '?' does not mean PRINT.

It will be seen that most of the commands not implemented are BASIC4 disk management commands. The equivalent BASIC2 sequences also do not appear to exist. I find this somewhat surprising as the means to decode the commands is available in the BASIC ROM.

Here is list of oddities I have found in using COMAL:
1) The command BASIC causes a system error although it is a valid COMAL command.
2) PRINT USING is not implemented. In my opinion the saddest point about using COMAL.
3) SAVE or LOAD to cassette does not work. The commands function but on reloading the file reads as gibberish.
4) When a program with indented lines is ENTERed an error occurs on lines with DIM statements. Just ignore the error. Type return over the displayed line and all will continue normally.
5) When a program is ENTERed from tape an EOF error occurs as a matter of course. It is not really an error.
6) ZONE: = 1 does not work correctly (it works as if ZONE: = 2). Use ZONE: = 0 and it works like ZONE: = 1!
7) Do NOT send disk commands when a disk is not switched on. A system crash will occur if you do!
8) DEBUG, while being a COMAL command causes a system error. If anybody finds any other problems or has any suggestions as to resolving the above or why they happen, please let me know. Write to me at 73, Minehead Way, Stevenage, Herts. SG1 2HZ.

In programs one very often wants to add a value to a variable so that in BASIC one says A = A + B. In COMAL with variable names up to 16 characters long this could get a trifle irritating: —

e.g. VARIABLE: = VARIABLE + VALUE

I have found the above line can be written as VARIABLE: + VALUE. Similarly one can subtract a value (or expression) from a variable by using ': − '. It does NOT work for multiplication or division however. You will see some examples of this in the COMAL program following this article.

To finish off this short piece I must mention another bug I have found with COMAL. If one ENTERs a program from tape I found that if one subsequently LISTs the program to tape the PET crashes. The solution is to do a SYS 65511 after the ENTER.

This is it for now except to say that after looking at some utilities from the US COMAL Users Group it would appear that the COMAL OPEN command is more detailed than I have identified so far. I have not investigated fully yet but it IS possible to send disk management commands from COMAL. In the meantime have fun with MAGIC SQUARES!

MAGIC SQUARES

The game is played with a board of nine cells. Each cell will have either a white dot or a white circle in it. You change the contents of cells from one symbol to the other by pressing one of the keys on the numeric keypad.

With the cells numbered as follows:

7	8	9
4	5	6
1	2	3

Pressing 1 changes the contents of cells 1-2-4-5
Pressing 2 changes the contents of cells 1-2-3

Pressing 3 changes the contents of cells 2-3-5-6
Pressing 4 changes the contents of cells 1-4-7
Pressing 5 changes the contents of cells 2-4-5-6-8
Pressing 6 changes the contents of cells 3-6-9
Pressing 7 changes the contents of cells 4-5-7-8
Pressing 8 changes the contents of cells 7-8-9
Pressing 9 changes the contents of cells 5-6-8-9

The game is complete when all the cells except 5 are filled with white dots and 5 is filled with a white circle.

You may give up at any time by pressing 0. You will then be shown the complete quickest solution. You may also see the quickest solution after you have solved the puzzle yourself.

For each game the PET will tell you the average number of moves to solve the puzzle shown. For a real challenge try to find the quickest solution which is always two moves less than the par score.

```
0010 //
0020 // COMAL MAGIC SQUARES
0030 //
0040 // BY BRIAN GRAINGER
0050 //   JANUARY 1982
0060 //
0070 DIM ANSWER$ OF 1, SPACE$ OF 39
0080 DIM SOLUTIONMOVE(9), CELL(9), VECTOR(9)
0090 FINISHED:=FALSE; OLDSTART:=0
0100 SPACE$:="    <39 sp>     "
0110 FOR COUNTER1:=1 TO 9 DO READ VECTOR(COUNTER1)
0120 DATA 229,63,397,219,341,438,355,504,334
0130 WHILE NOT FINISHED DO
0140  EXEC GETSTARTPOSITION
0150  EXEC FILLCELLS
0160  EXEC SOLVEIT
0170  PRINT "<clr>PAR FOR THIS PUZZLE IS";PAR;"MOVES"
0180  EXEC PRINTBOARD
0190  SOLVED:=FALSE; GIVENUP:=FALSE; NUMBEROFMOVES:=0
0200  WHILE NOT SOLVED AND NOT GIVENUP DO
0210   EXEC HITKEYTOGO
0220   EXEC GETMOVE
0230   IF NOT GIVENUP THEN EXEC ALTERCELLS
0240  ENDWHILE
0250  IF SOLVED THEN
0260   PRINT "<clr>YOUR STANDARD OF PLAY IS";
0270   CASE NUMBEROFMOVES OF
0280   WHEN PAR-2
0290    PRINT "PERFECT"
0300   WHEN PAR-1
0310    PRINT "EXCELLENT"
0320   WHEN PAR,PAR+1
0330    PRINT "VERY GOOD"
0340   WHEN PAR+2,PAR+3
0350    PRINT "GOOD"
0360   WHEN PAR+4,PAR+5
0370    PRINT "FAIR"
0380   WHEN PAR+6,PAR+7
0390    PRINT "PATHETIC"
0400   OTHERWISE
0410    PRINT "OF SOMEONE WHO IS GUESSING"
0420   ENDCASE
0430   REPEAT
0440    PRINT "DO YOU WISH TO SEE FASTEST SOLUTION"
0450    PRINT "(<rvs>Y<off> FOR YES <rvs>N<off> FOR NO)"
0460    INPUT ANSWER$
0470   UNTIL ANSWER$="Y" OR ANSWER$="N"
0480   IF ANSWER$="Y" THEN EXEC SHOWSOLUTION
0490  ELSE
0500   PRINT "<clr>YOU CHICKENED OUT. NOW SEE THE SOLUTION."
0510   EXEC SHOWSOLUTION
0520  ENDIF
0530  REPEAT
0540   PRINT "<clr>DO YOU WISH TO PLAY AGAIN"
0550   PRINT "(<rvs>Y<off> FOR YES <rvs>N<off> FOR NO)"
0560    INPUT ANSWER$
0570  UNTIL ANSWER$="Y" OR ANSWER$="N"
0580  IF ANSWER$="N" THEN FINISHED:=TRUE
0590 ENDWHILE
0600 //
0610 //
0620 PROC GETSTARTPOSITION
0630  REPEAT
0640   NEWSTART:=RND(0,511)
0650  UNTIL NEWSTART<>OLDSTART AND NEWSTART<>495
0660  OLDSTART:=NEWSTART
0670 ENDPROC GETSTARTPOSITION
0680 //
0690 //
0700 PROC FILLCELLS
0710  DECIMALNO:=NEWSTART
0720  FOR BIT:=1 TO 9 DO
0730   BINARYTODEC:=DECIMALNO MOD 2^BIT
0740   CELL(BIT):=SGN(BINARYTODEC); DECIMALNO:-BINARYTODEC
0750  NEXT BIT
0760 ENDPROC FILLCELLS
0770 //
0780 //
0790 PROC SOLVEIT
0800  NUMBEROFMOVES:=0
0810  FOR MOVE:=1 TO 9 DO
0820   IF CHECKMOVE(MOVE) THEN NUMBEROFMOVES:+1;
       SOLUTIONMOVE(NUMBEROFMOVES):=MOVE
0830  NEXT MOVE
0840  PAR:=NUMBEROFMOVES+2
0850 ENDPROC SOLVEIT
0860 //
0870 //
0880 PROC CHECKMOVE(MOVE)
0890  DECIMALNO:=VECTOR(MOVE); MOVECOUNT:=0
0900  FOR BIT:=1 TO 9 DO
0910   BINARYTODEC:=DECIMALNO MOD 2↑BIT
0920   DECIMALNO:-BINARYTODEC
0930   IF BINARYTODEC<>0 THEN
0940    IF BIT=5 THEN
0950     IF CELL(BIT)=1 THEN MOVECOUNT:+1
0960    ELSE
0970     IF CELL(BIT)=0 THEN MOVECOUNT:+1
0980    ENDIF
0990   ENDIF
1000  NEXT BIT
1010  CHECKMOVE:=MOVECOUNT MOD 2
1020 ENDPROC CHECKMOVE
1030 // [in lines 1070, 1110, 1130 & 1160 use shifted
1040 //  equivalents to those characters in quotes - Ed]
1050 PROC PRINTBOARD
1060  PRINT "<hcsr><7dn>"
1070  PRINT TAB(13),"@@@@@2@@@@2@@@@."
1080  FOR COUNTER1:=1 TO 14 DO
1090   CASE COUNTER1 OF
1100   WHEN 5,10
1110    PRINT TAB(13),"+@@@@[@@@@[@@@@3"
1120   OTHERWISE
1130    PRINT TAB(13),"]    ]    ]    ]"
1140   ENDCASE
1150  NEXT COUNTER1
1160  PRINT TAB(13),"-@@@@1@@@@1@@@@="
1170  EXEC PRINTCELLS
1180 ENDPROC PRINTBOARD
1190 //
1200 //
1210 PROC PRINTCELLS
1220  FOR COUNTER1:=1 TO 9 DO
1230   IF CELL(COUNTER1)=1 THEN
1240    CHARACTERPOKE:=81
1250   ELSE
1260    CHARACTERPOKE:=87
1270   ENDIF
1280   SCREENPOS:=33582-((COUNTER1-1) DIV 3)*200+
       ((COUNTER1-1) MOD 3)*5
1290   POKE SCREENPOS,CHARACTERPOKE
1300   POKE SCREENPOS+1,CHARACTERPOKE
1310   POKE SCREENPOS+40,CHARACTERPOKE
1320   POKE SCREENPOS+41,CHARACTERPOKE
1330  NEXT COUNTER1
1340 ENDPROC PRINTCELLS
1350 //
1360 //
1370 PROC GETMOVE
1380  REPEAT
1390   PRINT "<hcsr><dn>"
1400   PRINT SPACE$
1410   PRINT SPACE$
```

PAMPER YOUR PET WITH Codewriter©

Write your data base applications programs in a fraction of the time usually required to do so.

Microsystems introduce to PET owners the CODEWRITER, a superb program generator for the 8000 series PET with dual disc drive unit.

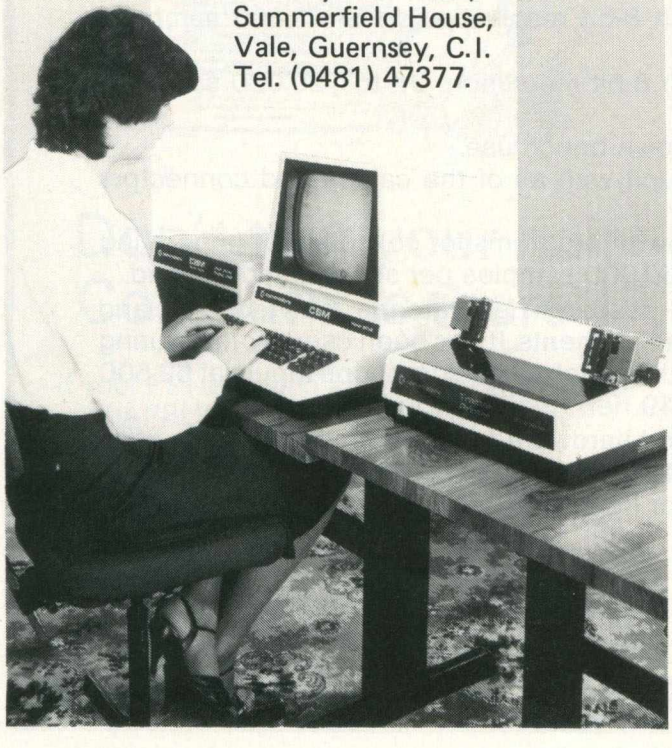
Screen layout, data entry validation, screen display of user-defined error messages, screen calculations, searching by any field - all are child's play to CODEWRITER 1. CODEWRITER DISC 2 provides printed reports and menu generators.

Codewriter 1 - £125

Codewriter Disc 2 - £65

Dealer enquiries welcome.

Pamper your PET promptly, write or 'phone
Microsystems Ltd.,
Summerfield House,
Vale, Guernsey, C.I.
Tel. (0481) 47377.



MIDLANDS

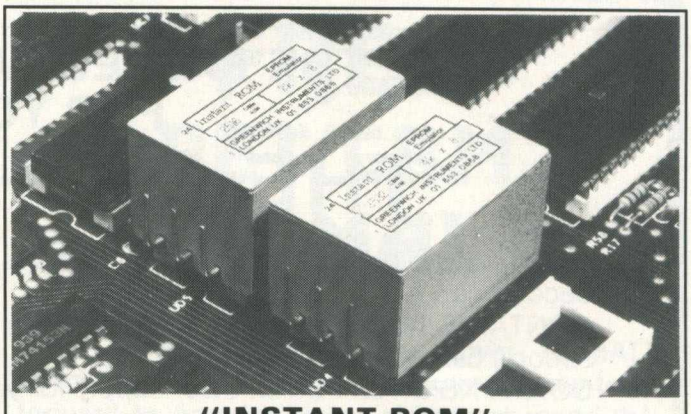
COMMODORE PET SERVICE CENTRE

Phone Anne on 021-772 8181 about our

1. WORKSHOP & FIELD REPAIRS
2. BUSINESS SOFTWARE
3. STATIONERY & SUPPLIES

CBS CONSULTANTS
COMPUTER BUSINESS SYSTEMS

75 Watery Lane, Birmingham B9 4HW.
Telephone: 021-772 8181 (7 Lines)



"INSTANT ROM"

"Instant Rom" ROM/EPROM EMULATORS contain CMOS RAM with internal battery backup. When the power is switched off, data is retained for up to 10 years.
In the PET, a 4K INSTANT ROM can be fitted in the $9000 or $A000 socket. Machine-code (and Basic) programs can be stored, and are available at switch-on.
INSTANT ROM saves time. It can be used for long periods; when the program is finally "bug-free", an EPROM can be programmed.

4K INSTANT ROM (ROM socket replacement).....£56.00
2K INSTANT ROM (character generator replacement).....£39.00
Adaptor GA1 (essential for PET users).....£6.00

"G-ROM E"

G-ROM E is a 4K EPROM which will Auto-run, at switch-on, any Basic or Machine-Code program stored in INSTANT ROM. Basic programs can be stored with a few quick key-strokes. No skill is needed. Programs can now be run without a tape or disk unit, and can be changed without cost to the user. Diagnostic aids are included.

G-ROM E (specify type of PET).....£25.00

Postage (£1.00) and VAT are extra. Leaflets are available.

"INSTANT ROM" and "PETCLOCK" are COMMODORE APPROVED PRODUCTS.
GREENWICH INSTRUMENTS LIMITED, 22 BARDSLEY LANE, GREENWICH, LONDON SE10 9RF, UK. Tel: 01-853 0868. Telex: 896691 Attn. GIL.

```
1420   PRINT SPACE$
1430   PRINT "<hcsr><dn>"
1440   PRINT "WHAT IS YOUR MOVE"
1450   PRINT "TYPE 1,2,3,4,5,6,7,8,9 OR 0 TO GIVE UP"
1460   INPUT ANSWER$
1470   MOVE:=ORD(ANSWER$)-48
1480  UNTIL MOVE>=0 AND MOVE<=9
1490  IF MOVE=0 THEN
1500   GIVENUP:=TRUE
1510  ELSE
1520   NUMBEROFMOVES:+1
1530  ENDIF
1540 ENDPROC GETMOVE
1550 //
1560 //
1570 PROC ALTERCELLS
1580  CASE MOVE OF
1590  WHEN 1
1600   CELL(1):=1-CELL(1); CELL(2):=1-CELL(2)
1610   CELL(4):=1-CELL(4); CELL(5):=1-CELL(5)
1620  WHEN 2
1630   CELL(1):=1-CELL(1); CELL(2):=1-CELL(2)
1640   CELL(3):=1-CELL(3)
1650  WHEN 3
1660   CELL(2):=1-CELL(2); CELL(3):=1-CELL(3)
1670   CELL(5):=1-CELL(5); CELL(6):=1-CELL(6)
1680  WHEN 4
1690   CELL(1):=1-CELL(1); CELL(4):=1-CELL(4)
1700   CELL(7):=1-CELL(7)
1710  WHEN 5
1720   CELL(2):=1-CELL(2); CELL(4):=1-CELL(4)
1730   CELL(5):=1-CELL(5); CELL(6):=1-CELL(6)
1740   CELL(8):=1-CELL(8)
1750  WHEN 6
1760   CELL(3):=1-CELL(3); CELL(6):=1-CELL(6)
1770   CELL(9):=1-CELL(9)
1780  WHEN 7
1790   CELL(4):=1-CELL(4); CELL(5):=1-CELL(5)
1800   CELL(7):=1-CELL(7); CELL(8):=1-CELL(8)
1810  WHEN 8
1820   CELL(7):=1-CELL(7); CELL(8):=1-CELL(8)
1830   CELL(9):=1-CELL(9)
1840  WHEN 9
1850   CELL(5):=1-CELL(5); CELL(6):=1-CELL(6)
1860   CELL(8):=1-CELL(8); CELL(9):=1-CELL(9)
1870  ENDCASE
1880  SOLVED:=TRUE
1890  FOR COUNTER1:=1 TO 9 DO
1900   IF COUNTER1=5 THEN
1910    IF CELL(COUNTER1)=1 THEN SOLVED:=FALSE
1920   ELSE
1930    IF CELL(COUNTER1)=0 THEN SOLVED:=FALSE
1940   ENDIF
1950  NEXT COUNTER1
1960  EXEC PRINTCELLS
1970  PRINT "<hcsr><dn>"
1980  PRINT SPACE$
1990  PRINT SPACE$
2000  PRINT SPACE$
2010  PRINT "<hcsr><dn>
2020  PRINT "LAST MOVE WAS";MOVE
2030  PRINT "NUMBER OF MOVES TO DATE IS";NUMBEROFMOVES
2040 ENDPROC ALTERCELLS
2050 //
2060 //
2070 PROC HITKEYTOGO
2080  REPEAT
2090   PRINT "<hcsr><3dn>
2100   PRINT SPACE$
2110   PRINT <hcsr><3dn>
2120   INPUT "PRESS 'RETURN' TO CONTINUE ": ANSWER$
2130  UNTIL ANSWER$=""
2140 ENDPROC HITKEYTOGO
2150 //
2160 //
2170 PROC SHOWSOLUTION
2180  EXEC FILLCELLS
2190  EXEC PRINTBOARD
2200  FOR NUMBEROFMOVES:=1 TO PAR-2 DO
2210   EXEC HITKEYTOGO
2220   MOVE:=SOLUTIONMOVE(NUMBEROFMOVES)
2230   EXEC ALTERCELLS
2240  NEXT NUMBEROFMOVES
2250  EXEC HITKEYTOGO
2260 ENDPROC SHOWSOLUTION
```


STILL THE FASTEST ANALOGUE INTERFACE FOR YOUR PET!

THE ORTHOLOG 181

CONTAINS eight analogue input channels with 8-bit resolution. Up to 62,500 samples/second can be fed directly into the Pet.

CONTAINS two analogue output channels with 8-bit resolution. Up to 120,000 samples/second can be fed out from the Pet.

CONTAINS a +5, —5 volt ower supply for your own bench use.

IS SUPPLIED as a completely self-contained unit with all of the cables and connectors needed.

IS SUPPLIED with a comprehensive handbook with programs for collecting or generating data at speeds between 1 sample per hour to 50,000 samples per second and beyond.

THE ORTHOLOG 181 is in use in industrial research laboratories, polytechnics and universities, and in medical and government establishments. It has been used for measuring thermocouple outputs at one sample/minute and for analysing microphone inputs at 62,500 samples/second. It costs only **£324** cwo or **£349** nett m/a, all-inclusive.

Ortholog Ltd. also supply supporting software and hardware for use with the 181. We offer programs which let you turn your Pet into a sampling and storage oscilloscope for the 0.01 to 20,000 Hz frequency range and pre-amplifiers for measurements down to 10 microvolts or less. Others items are listed in our brochures.

FOR FURTHER DETAILS OR FREE DEMONSTRATION CONTACT:

ORTHOLOG LTD., P.O. BOX 72, EDGWARE, MIDDLESEX HA8 6RD
Tel. 01-952 2459

Give your PET a home.... Buy it a PETDESK!

A Commodore approved product.

Specially designed to take any Commodore Pet system.

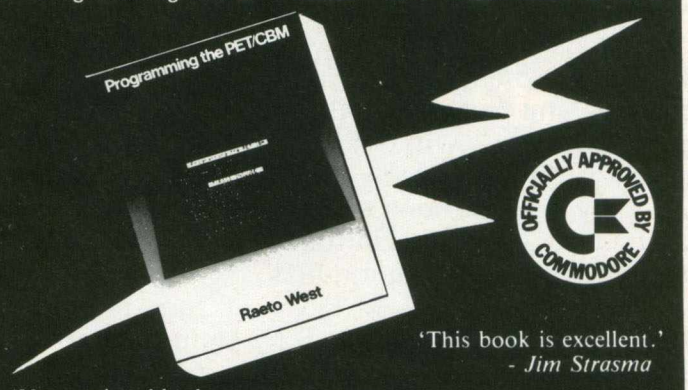
Black leathercloth top and Black metal frame.
Paper feed tray, top extension shelf. Concealed cables and 4 way 13 amp plug socket.
Mounted on castors. Size 1470 x 560 x 675 mm.
Delivered flat packed.

Price £189.50 includes VAT and delivery.

This offer available UK only. Cheques with order to:
Tirith Ltd, Pear Tree House,
Woughton on the Green, Milton Keynes
MK6 3BE. Telephone: (0908) 679528

NEW

Programming the **PET/CBM**



'This book is excellent.'
- Jim Strasma

'Unquestionably the most accurate and comprehensive reference I have seen to date.'
- Jim Butterfield

Bestseller — comprehensive teaching and reference book on all software aspects of Commodores 2000, 3000, 4000 and 8000 microcomputers and peripherals.

Many programs, charts and diagrams.
17 chapters, appendices, and index.
iv + 504 pages, 19 × 26 × 2½cm.
Paperback, ISBN 0 9507650 0 7.
Price in UK and Europe £14.90 each (incl. post and heavy-duty packing).
Five or more books at £12.90 each.
Soft plastic covers 45p each.
Dealer enquiries invited.
LEVEL LTD., PO Box 438, Hampstead, London NW3 1BH. Tel: 01-794-9848.

Cut out or copy coupon, or write to:
LEVEL LTD. (CC), PO Box 438, Hampstead, London NW3 1BH.
Send ..... copy/ies of *Programming the PET/CBM* at £14.90 (post free)

I enclose cheque/P.O. for £.....or official order.
NAME .....
ADDRESS .....
.....

Fast Service — same day despatch CC582

Don't tear your hair Compufix is there !

If your PET is having a nervous breakdown or your APPLE has bitten it's last byte. Then you need us!!
We are offering fast repairs to faulty systems.
Maintenance contracts are available.

COMPUFIX

Swindon (0793) 694066, 612341/2

FREE PET/CBM COMAL

"The excitement in Europe (over COMAL) seems to be growing by the hour and we look forward to America being able to share in the good fortune of having an easy-to-use, structured, planning language at last."

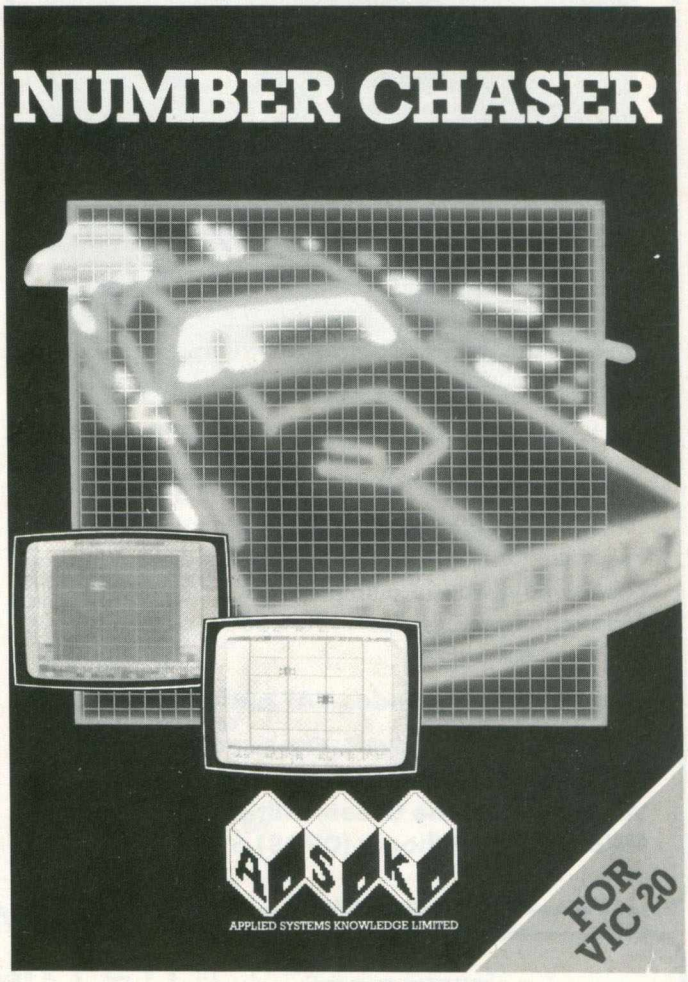
The power of PASCAL and the ease of BASIC can now be yours with Commodore COMAL, a new programming language from DENMARK. It is being distributed in the USA by the COMAL USERS GROUP. To find out more about COMAL and how you can get a free disk copy of Commodore COMAL, send a large self-addressed stamped (35 cents) envelope to:

COMAL USERS GROUP
5501 GROVELAND TER., MADISON, WI 53716.
Outside USA please add $2.00 for airmal and handling.
*PET & CBM are trademarks of Commodore Business Machines.

Software Review

To quote Professor Tom Stonier, of the School of Science and Society at Bradford University, ''A good computer without any software is like a good stereo without any records''. In other words you may have got one of the best systems in the land, but one which is virtually useless due to the lack of accompanying material.

This month's review looks at four products from a company called A.S.K. Limited (Applied Systems Knowledge), whose aim it is to provide good quality educational software for the Vic.



New Programs

There are a large number of programs available on cassette for the Vic, but unfortunately most of them tend to be on the games side of things, and the few educational packages that do exist are mainly 'one-off' lessons that the pupil or user would rapidly lose interest in. Similarly, the programs are usually aimed at existing tutorial lessons, thus simply repeating what the teacher has already gone through, rather than being aimed at reinforcing what has already been learnt.

Therefore, it makes a refreshing change to be presented with a set of programs that not only maintain ones enthusiasm for using them, but also prefer to back up what may have been taught in the school or college.

Since this is a worthwhile and relatively novel approach, it's worth taking a close look at the company behind all this before examining the individual programs in further detail.

A.S.K. Limited

Managing Director of A.S.K. is a gentleman called Peter Lever, an affable but extremely business-like person, whose aim is (reasonably enough) to make the company a very successful one, and also (on a less profit-conscious note) to provide a large range of good, competent educational programs for the Vic.

The whole idea behind the packages is to encourage the learner to work from home, and complement the work that has already been done in the traditional classroom environment. Thus the programs are not just 'one-off' lessons, but give a comprehensive approach to whatever particular area is under consideration.

For instance, the two areas that A.S.K. have originally aimed for are mathematical and literacy skills, and at the time of writing they have two programs in each area. The method by which they go about selecting the topics or the programs is quite interesting, and deserves quick mention. Basically they have a team of authors, most of whom are in the teaching environment already, and thus know the kind of material that is required from the home educational angle.

This team then come up with a list of about 8 or 9 topics in each subject area, and produce a 'script' for each one. This is then presented to a professional team of programmers who come up with the finished goods.

Rampacks

These 'finished goods' are quite sophisticated, and make extensive use of the colour graphics available on the Vic.

As a consequence of this, they majority of the programs will require a 16K rampack before they can actually be used. This raises an interesting point: how many schools or homes will have, or will be willing to buy, a 16K rampack? A fairly expensive commodity.

Still, this is the marketing decision that A.S.K. have made, and one by which they stand or fall. To offset this, they are making available 16K packs themselves, at a reduced cost of 67.50 pounds for each one, provided you buy at least one of their programs.

The Programs

As I said, there are four programs on the market at present, although they hope to expand this up to fifteen by the end of the year, still covering the mathematical and literacy skills.

'We Want To Count' is aimed at the lower end of the scale, for children of three years or upwards. It involves teaching them to count, using the number 1 to 5, and like all A.S.K. programs this challenge is presented in a game, rather than a stern lesson, format. Four different games, and a fair degree of randomness, make this a good and useful program. It is priced at 8.95 pounds, like all of the programs so far released.

'Number Chaser' is the next one up, aimed at the 5 to 12 year market, and is the only program that doesn't require a 16K rampack: it needs an 8K one instead! It all revolves around a car race, and presents the opportunity to improve estimation and multiplication skills. There are a number of levels of difficulty, and thus the program can be made different every time the user comes to run it. As usual, a lot of colourful graphics serve to improve the display, and present 'rewards' for getting correct answers.

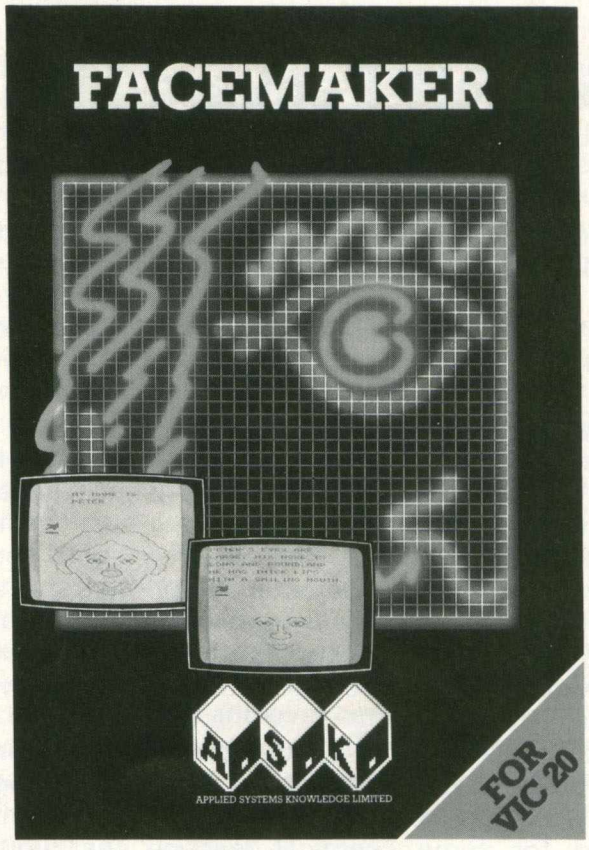
'Facemaker' is again aimed at the 5 to 12 years age bracket, and is one of the programs on the literacy side, looking at spelling, vocabulary and increasing observational skills. The idea is to make up faces on the screen, and with skillful use of the high resolution graphics in the program, some quite reasonable displays can be made. As the number of faces is virtually limitless, this is a game that the user can come back to again and again.

'Twister' is the other program they have on the market at present. This is being pointed at a higher level than the others, for 8 year olds and upwards. In effect it is a 2 dimensional version of the well known Rubik Cube, involving moving coloured squares around until you end up with no rows or columns with a repeated colour. As far as this reviewer was concerned, this one was the pick of the crop, but of course the games are all aimed at people at the secondary or junior school level.

Future Plans

The two main driving forces behind the company are the aforementioned Professor Stonier, and Doctor Mike Thorne at the department of Computing Mathematics at Cardiff University.

Tom Stonier has said that by the turn of the century every home will either have, or will have access to, a computer. Thus it is important that there is a good ground base of software, and



A.S.K. are aiming to get there before the end of this century.

The aim of all their programs is to provide interactive entertainment as well as good tuition. A computer is an infinitely patient thing: it, unlike the average teacher, will not lose patience, switch off, have a bad day or whatever, but rather will keep on going ad infinitum (barring intervention from the C.E.G.B.!). Learning at home can also be a lot more fruitful, with the private atmosphere providing the ideal background to absorb information.

As time goes on and more and more programs appear from the company, they anticipate moving onto C.S.E. O-Level topics, as well as maintaining the original scheme of junior subjects. The programs will still have the same kind of format i.e. combining education with entertainment, which is really a very good way of tackling this kind of software.

Summary

Four very good programs, which are well worth your examining in greater detail. With the promise of more to come, it looks like A.S.K. could become a major force on the educational software front, but of course, as with all these things, time alone will tell.

Meanwhile, for further information they can be contacted on 01-876 0102.

Hardware Review

Hewlett Packard Line Plotter

There are a number of graphics plotters around at present that are capable of being interfaced to the Pet. Here we turn our attention to what might be termed the 'Rolls-Royce' of plotters, the Hewlett-Packard 7470A.

Why Use a Graphics Plotter?

There are many ordinary printers for the Pet, the sort which produce printouts of listings, screen contents, for use with word processing packages, and so on. You can even, with some ingenuity, produce characters on printers that are not normally accessible from the Pet keyboard. However, there comes a point where you cannot reproduce a hard copy of the information you would like.

For instance, you want to produce a coloured bar chart showing your sales figures, you might like a comparison (and hence colour for clarity) chart for some chemical reaction that you're monitoring, or produce a very accurate reproduction of some component diagram. Whatever the reason, you will need a graphics plotter of one kind or another.

First Impressions

A first look at the Hewlett-Packard 7470A is indeed impressive. Securely bound and contained, complete with copious documentation, warranty card etc., it certainly looks the part. I know it's only a small point, but the mains lead even has a plug on it! Taking everything into consideration, the package as a whole sets off to a favourable start in any reviewers (or indeed users) eyes.

Connecting it up to the Pet is simplicity itself. It just plugs straight into the IEEE connector at the back of the disk drive (assuming you're using a complete system), and you're ready to go straightaway.

Documentation

One has always to consider documentation when taking a look at any product. It can sometimes make the difference between a good piece of hardware/software and a merely average piece. The documentation accompanying the Hewlett-Packard plotter is sufficient for the job. It does not go overboard, and assumes that you have a fair knowledge of programming: a reasonable enough assumption if you've just spent over 1,000 pounds on buying the product.

Rather, it gives you the basics of how to connect the device up, a complete rundown on all the extra plotting commands now at your disposal, various hints and tips on setting up (what paper to use, what pens to use etc.), and overall does enough to get you going.

Also included is a full list of service and support centres, warranty cards, a business reply form to tell them what you think of the documentation and hardware, and all of this combined serves to give you the (very probably correct) impression that you are dealing with an extremely professional company.

Software Accompaniment

Also provided, at our request rather than an integral part of the package (although I should imagine a quick telephone call would probably do the trick for you) was a disk of some sample demonstration software which put the plotter through its paces. Our thanks go to Andy Palmer of Hewlett-Packard for giving us this, and a word of praise for the help we had from them whilst reviewing the product: the staff were courteous, polite, and always ready to help.

That software was a great help in designing and setting up ones own programs. The plotter can accept something in the order of 60 graphics commands, and seeing these demonstrated rather than going through the often tedious process of trying them all out oneself was a rapid step forward in getting the best out of the device.

The Hardware

The particular version of the plotter we were examining retails at 1,021 pounds excluding VAT, although there are two other versions also suitable for the Pet: these are approximately 60 pounds more expensive. Comparable in size to the old Commodore 3022 printer, it is rather lighter than that particular printer, and thus portability is very easy.

Connection to the Pet is via the IEEE bus, and the plotter is programmed using a mixture of Basic, and its own graphics programming language HP-GL. This consists mainly of two letter codes which bear some resemblance to the required action. For example, PU initiates Pen Up, and so on.

Two colour plotting is possible by use of two

pens: the plotting arm grabs the relevant one as and when required. This action can be controlled either via hardware or software. Plotting is done using a scale of 'plotter units'. Each unit is 0.025 millimetres in length, and is the smallest move the plotter can make. Using this system of measurement, the plotter has a plotting space of 10,900 units in the X direction and 7,650 units in the Y direction, using A4 paper.

A nice feature when plotting is that, if for some reason you're about to begin plotting points outside the range of the paper, the arm will hold the pen until your plot comes back into range again, and off it will start once more.

Using HP-GL, all plotting is relative to two scaling points P1 and P2. These have their default values at power up, but can easily be altered again either from hardware or software. Reasons for altering them at all would be, for example, producing two equal plots by starting off at two equal scaling points, doing reduced size copies of plots by redefining a new plotting area, and so on.

Finger Tip Control

As well as being programmed from software, there is an extensive range of controls built into the machine itself.

There is a panel of 13 switches to the right of the plotting area (you can see it in the photograph). These perform a variety of functions, and can be used to draw plots when the plotter is not connected up to the Pet. Buttons exist to move pens forwards, backwards, left and right, and pressing two buttons simultaneously will cause the pen to move diagonally. A further button controls the speed of movement, and two more determine whether the pen is up or down.

Buttons marked P1 and P2 move the pen to the scaling points mentioned earlier. Many of these buttons have dual purposes. For instance, the one which controls the speed of movement can also be used to halt program execution whilst plotting, for as long as the button is held down. However, this is where I found the one fault with the plotter. It certainly halted execution, but on releasing the button again the program did not start from where it left off: there was a slight jump away from the last point plotted.

The final two of the 13 switches are an Enter button, wich has a variety of useful purposes, and a View button, which allows you to suspend printing and get a complete view of the plot so far.

Ease of Use

We found the plotter very easy to install and use, both from a programming point of view and a hardware point. There was a power up test (or confidence test as they call it in the manual) just to check that everything was working correctly. Programming, once you'd got used to the two letter commands, was quite straightforward, and made easier by the programmer's reference card provided. This gave a summary of all the commands available. Looking at the sample programs given also helped.

Interfacing routines to ones own programs was simple enough, and programming the 7470A soon became second-nature. Certainly as easy as any of the existing Commodore printers, or indeed any other printer that I've come across.

Technical Overview

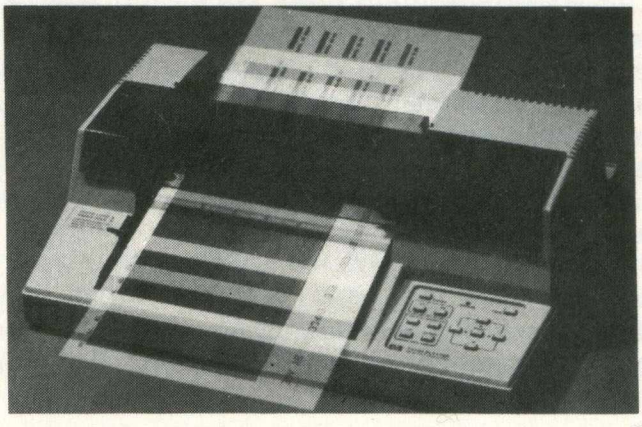
We've already covered most of the points of interest. Only two other main areas need to be covered here. The plotter is designed to work both in the States and over here, and as you know Stateside they work to a different convention as regards paper size. Consequently there is a rocker switch at the rear of the plotter which can be set to either US standard or A4 size.

Five further rocker switches are used to determine the plotter address value, set at 05 when the printer leaves the factory. You can place the plotter in listen-only mode by setting all five switches to 1: in this mode the plotter doesn't have an address, but listens to all data transmitted on the bus. Not particularly useful.

Summary

An easy to use peripheral for any Commodore computer with an implemented IEEE port. At 1021 pounds it is perhaps a trifle expensive, but you are paying for an excellent machine, well documented and well backed-up in terms of servicing and technical help. Hewlett-Packard have a reputation for producing good, reliable, robust equipment: the 7470A graphics plotter lives up to that reputation.

Highly recommended.



Book Review

This month a look at two books covering very different aspects of Pet usage, namely Structured Programming with Comal, by Roy Atherton, and Pet Interfacing, by James Downey and Steven Rogers.

Structured Programming with Comal

Both last month and this we've had a feature on Comal, so you should have a fair idea now of what the language is all about. Roy Atherton, who wrote this book (available from Ellis Horwood/Wiley, at a price of 6.90 pounds for the paperback version: at 18.50 pounds leave the hardback version alone!), is one of the U.K.'s leading spokesmen on Comal.

Before taking a look at the book, a few words on Comal itself. As I'm sure you know, Comal was first designed by Borge Christensen in Denmark, and was intended to combine the simplicity of Basic with the structure of Pascal. In other words, an easy to learn, and also easy to understand, language. Its popularity has grown enormously in the last few months, and now we're beginning to see the first of the books appearing.

Roy has long been respected as a spokesman for the art of the good, sound programming, and now not only in Basic. Comal is the ideal language for a programmer such as this, and his enthusiasm shows in the book. All of the major concepts he introduces are lavishly illustrated with many programmed examples, which makes a change from the usual rhetoric aimed at the user.

The art of Comal

The art of the book is the art of Comal itself, for without the language we would never have had the book. Both follow the same styles: aimed at the beginner, simple enough to understand, no nonsense method of writing, and above all a clarity that would make it an excellent book (and indeed language) for the student to computer programming.

Basically (sorry, but there are times when you can't avoid using the word) Roy goes through a gentle introduction to the language, covers the various statements in the language, moves onto structured programming in Basic (thus showing how Comal scores in this area), and then finally a number of chapters on general programming ideas.

Summary

Learning about programming, and learning about good programming at the same time, are two topics that very rarely go hand in hand. With this book, thankfully they do. If you're interested in Comal programming (and more and more people are becoming so), then this is well worth investigating. Computer teachers and lecturers especially ought to take note.

Pet Interfacing

This particular book is published by Howard W. Sams and Co., at a cost of 11.85 pounds, and should be available from any major computer bookstore, or indeed any bookshop with a large range of computer books.

Much has been written about the use of the Pet as a controller of other scientific instruments: indeed, this ability to interface with the outside world is one of the main reasons for the success and longevity of the machine. Why change a winning formula?

Commodore Computing (and of course Commodore Club New of old) published many articles on the use of the Pet in this field. This particular issue, for instance, sees a lengthy article by Allan Potten on some of the many uses of the User Port.

Inner Contents

The book covers both the IEEE port and the User port, and drops off at a number of interesting topics en route, leaving you with a fair sprinkling of ideas worthy of further development.

The first couple of chapters provide an introduction to what the Pet consists of, and also gives instructions for the building of a breadboard for access to the User Port. This board is subsequently used as the basis for all the user port subjects subsequently tackled. Many of these are extremely interesting, and cover much more than the usual serial input/output and analogue to digital conversion.

The memory expansion port comes next, with an equally large number of projects for you to tackle, before moving onto a general chapter on interfacing per se. The final part of the book takes a detailed look at the IEEE port, including using the Pet as an IEEE controller, and as a listener/talker.

To round off an excellent book there are copious appendices, giving flow charts, assembler listings etc.

Summary

A very good book, well and clearly written. For anyone who wants to dive around the Pet, getting it talking to all manner of devices, and who can handle a soldering iron, this is a must.

Butterfield

Compiler Comments

I don't want to become involved in the Great Debate about compilers. On the other hand, it's almost irresistible to dive in and add a few footnotes. You'll find no product reviews here. Just a little talk about what's involved.

For BASIC?

Some languages were designed for compilers. In fact, the compiler was designed first, and whatever it turned out you had to type in ended up as the language. FORTRAN started more or less this way. To put compilers in perspective, we have to do a little historical work.

Once, long ago, there were no interactive computers. You punched up a deck of cards and if you were lucky an operator would run them sometime that week. Most of the results came back saying something like SYNTAX ERROR (does that sound familiar?). There was no point in having an interpreter language; you wouldn't be there to watch it happen. We had FORTRAN and COBOL and others . . .

The first FORTRANS, for example, were tricky. If you used a variable called DIGIT, it would turn out to be a floating-point number; on the other hand a variable called NUMBER would be fixed-point. Heaven help you if you typed TOTAL=TOTAL+1; you'd get a ?MIXED MODE error notice and have to recode TOTAL=TOTAL+1.0 to fix it. To input or output you needed to give more than the command: an extra line called FORMAT was needed, written in advanced gibberish. Honest.

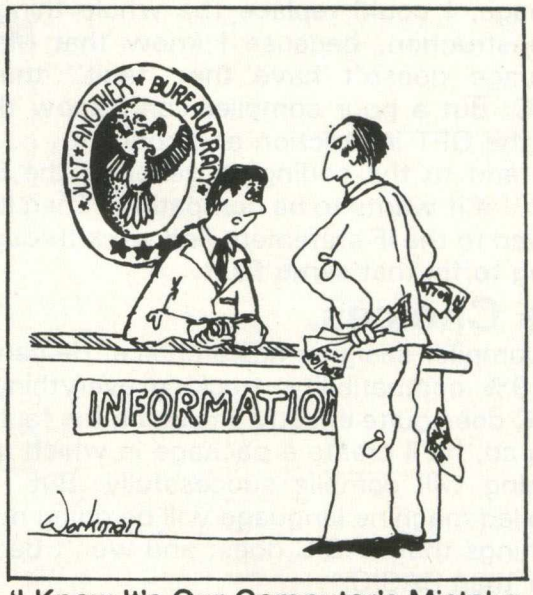
Many of these problems have been fixed up over the years — you did know that there was more than one FORTRAN, didn't you? — but the style remains. The programmers have to adapt to the machine, and interactive is still an alien concept.

And Now, BASIC . . .

Along came BASIC. It's a loose language: you don't have to dimension some arrays; strings wander all over; sometimes you can have FOR and NEXT items that don't match (bad practice, but it can be done) . . . and interactive users love it.

What's the problem? Things that are not clearly defined by BASIC. Let's look at a few of them.

Strings may be the worst thing that a compiler has to deal with. BASIC doesn't tell the compiler how big any string is likely to be — ever. INPUT X$ gives no hint as to the size of string X$. The poor compiler has a grim choice: allow maximum space for all strings and waste a lot of memory; or bounce the strings around as they change. The first alternative costs you program size; you write this little program that says DIM A$(1000) and the compiler immediately reports OUT OF MEMORY since it tries to allocate 255000 bytes for the array. The second alternative costs you time; no matter what you call it, some sort of garbage collection will have to take place. And then people complain because they expect compilers to produce fast fast code.



'I Know It's Our Computer's Mistake, Mr. Hill, But It Would Be Easier in the Long Run if You Did Change Your Name to ZP4/QE/70K.'

At first glance we think that the whole object of compiling is to get speed. But we don't give the compiler enought information to work up a really fast program. It's obvious that FOR J=1 TO 10 can run faster if we treat J as an integer. Unfortuntely, we're not allowed to code FOR J% . . . so the compiler will have to figure it out for itself. And what will it do with FOR J=A TO B? Until A is computed, we cannot know if it's integer or not.

It's obvious to us. We wrote the program. But the dumb compiler can't read our minds; and BASIC doesn't give enough explicit information to do the job.

One last example. It's one of the annoying things about BASIC that we sometimes have to code things like GET=1,X$: IF X$="" THEN X$=CHR$(0) mostly to cover failings in BASIC itself. If I were hand-coding into machine

language, I could replace the whole thing with one instruction, because I know that Machine Language doesn't have the ''fault'' that's in BASIC. But a poor compiler can't know that. It sees the GET instruction and codes it . . . and it must add to the coding to generate the BASIC ''fault'' if it wants to be compatible. Then it must proceed to the IF statement and work through the coding to fix that same fault.

The Choices.

The compiler designer has a choice. He can code for 99% compatibility, tracking everything that BASIC does quite exactly (including the faults). In doing so, he'll create a package in which almost anything will compile successfully. But — the compiled machine language will be doing most of the things that BASIC does, and won't be much faster than BASIC.

On the other hand, the designer can ask the user to make changes to his program before compilation that will help the process. He may also have things that compile from BASIC in a non-standard manner. He may make arbitrary decisions on BASIC structures — all FOR loop variables will be fixed-point, for example. And the compiler may question the user during compilation: How large is string M$ likely to be? Can J be fixed-point? The user has to work harder, but the end product runs faster.

Either way, the compiled program is not likely to be smaller in size than its BASIC source. It's difficult to code 100 IFJ 5THENPRINT''J IS'';J in less than the 19 bytes that BASIC uses. And good compilers add extra arithmetic — fixed-point addition, for example — that takes up overhead space.

Why Compile?

It's your choice. If you have a program that runs for five hours, you will probably be delighted with a paltry four-to-one compiler speedup. If you want protection against listing, a compiler will do a good job of instant obfuscation.

Don't lose perspective. A program that spends most of its time waiting for an operator or for a printer won't speed up much under compilation.

Machine Language Programmers will be happy to know that they are not yet obsolete. Compilers can do a useful job. But until they get the brains equivalent to a human's judgement, they won't replace hand coding.

How to buy a word processing program...

First, go to your CBM/PET dealer and see at least two wordprocessing programs. Second, make sure that one of those you see is a WORDFORM from LANDSOFT.

We are serious when we say you should see more than one. Everyones wordprocessing requirements are different. You will want to ensure that the package you buy will do all you require. But also you will not want to pay for functions you don't need.

There are two LANDSOFT word-processors — WORDFORM and WORDFORM II. They are both exceptional programs. You may well find that WORD-FORM will do everything you need, but should you ever want to update to WORDFORM II, we will always supply for the difference in price on return of the other program.

So don't be talked into a very expensive program until you have satisfied yourself that one of the WORDFORMs will not do all you want. Buying another program and then becoming aware of the WORDFORM excellence would be most frustrating.

WORDFORM versions for 3032, 4032 and 8032 **£75 + VAT.**

WORDFORM II for 8032 only **£150 + VAT.**

LandSoft

28 Sheen Lane,
London SW14 8LW.
Telephone: 01-399 2476/7

SUPERIOR PROGRAMS FOR THE CBM/PET MICROCOMPUTER

Applications

A Glass Teletype Listing

Basic 4 only

```
0001  0000                ; te letypesrc
0002  0000                ;
0003  0000                ;*****************************************************
0004  0000                ;*                                                   *
0005  0000                ;*                 Simple Glass Teletype             *
0006  0000                ;*                                                   *
0007  0000                ;* For communication in true ASCII, with cursor,     *
0008  0000                ;* control key, delete and escape.                   *
0009  0000                ;*                                                   *
0010  0000                ;* Enhanced versions of this program, with disk dump,*
0011  0000                ;* disk to printer, parity error checking (user     *
0012  0000                ;* reconfigurable) & bidirectional screen scroll are *
0013  0000                ;* available from:                                   *
0014  0000                ;*                                                   *
0015  0000                ;*                 Ariadne Software Ltd.             *
0016  0000                ;*                 56 Chandos Road                   *
0017  0000                ;*                 London NW2                        *
0018  0000                ;*                                                   *
0019  0000                ;*                 Tel 01-452-4390                   *
0020  0000                ;*                                                   *
0021  0000                ;*****************************************************
0022  0000                ;
0023  0000                ;(you never saw software with a plug on it before?)
0024  0000                ;
0025  0000                ;rom addresses (basic 4)
0026  0000                ;
0027  0000                strout = $bb1d          ;print string specified by A,Y
0028  0000                close  = $ff2e          ;close file A
0029  0000                open   = $ff63          ;open file specified in page 0
0030  0000                coin   = $ffc6          ;set input channel to X
0031  0000                coout  = $ffc9          ;set output channel to X
0032  0000                clrch  = $ffcc          ;restore normal i/o
0033  0000                inchr  = $ffcf          ;input from current channel
0034  0000                outch  = $ffd2          ;output to current channel
0035  0000                cgetl  = $ffe4          ;get chr from keyboard queue
0036  0000                ;
0037  0000                ;i/o
0038  0000                ;
0039  0000                portb  = $e822          ;PIA # 2 port b
0040  0000                crb    = $e823          ;PIA # 2 control register b
0041  0000                chrset = $e84c          ;VIA reg controls char set
0042  0000                ;
0043  0000                ; page 0 (tape area used for program variables)
0044  0000                ;
0045  0000                ctrlfg = $b7            ;msb flags control on
0046  0000                arrcol = $b8            ;column posn of up-arrow cursor
0047  0000                lffg   = $b9            ;msb flags line feeds expected
0048  0000                curadd = $c4            ;address of start of cursor line
0049  0000                curcol = $c6            ;column position of cursor
0050  0000                fnmlen = $d1            ;length of file name
0051  0000                fileno = $d2            ;logical file number
0052  0000                secadd = $d3            ;secondary address
0053  0000                devno  = $d4            ;device number
0054  0000                ;
0055  0000                ; alterable parameters
0056  0000                ;
0057  0000                linlen = 40             ;40 column PET
0058  0000                ;
0059  0000                ; change to linlen = 80 for 80 col version
0060  0000                ;
0061  0000                      * = $0401
0062  0401                ;
0063  0401                ;
0064  0401                ; line of basic - 10 sys1037
0065  0401                ;
0066  0401  0b 04               .word basend
0067  0403  0a 00               .word #000a
0068  0405  9e                  .byt $9e,'1037',#00
0068  0406  31 30
0068  040a  00
0069  040b  00 00          basend .word $0000
0070  040d                ;
0071  040d                ;print heading etc
0072  040d                ;
0073  040d  a9 78               lda #<headng    ;'do you expect line feeds?'
0074  040f  a0 05               ldy #>headng
0075  0411  20 1d bb            jsr strout
0076  0414  20 e4 ff     head10 jsr cgetl       ;get reply
0077  0417  a2 00               ldx ##00        ;check for y or n
0078  0419  c9 4e               cmp #'n'
0079  041b  f0 06               beq head20
0080  041d  a2 80               ldx ##80
0081  041f  c9 59               cmp #'y'
0082  0421  d0 f1               bne head10      ;if neither try again
0083  0423  86 b9         head20 stx lffg       ; else set up line feed flag
0084  0425  20 d2 ff            jsr outch       ;echo y or n
0085  0428  a9 c4               lda #<waitt     ;'obtain crx then press any key'
0086  042a  a0 05               ldy #>waitt
0087  042c  20 1d bb            jsr strout
0088  042f  20 e4 ff     wait   jsr cgetl       ;wait for key closure
0089  0432  f0 fb               beq wait
0090  0434                ;
0091  0434                ;open 5,5 to modem etc.
0092  0434                ;
0093  0434  a9 05               lda ##05        ;set up for open
0094  0436  85 d2               sta fileno
0095  0438  85 d4               sta devno
0096  043a  a9 00               lda ##00
0097  043c  85 d3               sta secadd
0098  043e  85 d1               sta fnmlen
0099  0440  20 63 f5            jsr open        ;open 5,5
0100  0443  a2 05               ldx ##05        ;get#5
0101  0445  20 c6 ff            jsr coin
0102  0448  20 cf ff            jsr inchr
0103  044b  20 cc ff            jsr clrch
0104  044e  ad 22 e8            lda portb       ;clear srq interrupt flag
0105  0451  a9 00               lda ##00
0106  0453  85 b7               sta ctrlfg      ;clear control key flag
0107  0455                ;
0108  0455                ;clear screen etc.
0109  0455                ;
0110  0455  a9 0e               lda ##0e        ;set upper/lower case char set
0111  0457  8d 4c e8            sta chrset
0112  045a  a9 ec               lda #<comt      ;'on line'
0113  045c  a0 05               ldy #>comt
0114  045e  20 1d bb            jsr strout
0115  0461                ;
0116  0461                ;main program loop
0117  0461                ;
0118  0461  a5 c6         main   lda curcol      ;calculate arrow position
0119  0463  18                  clc
0120  0464  69 28               adc #linlen
0121  0466  85 b8               sta arrcol
0122  0468  a8                  tay
0123  0469  a9 1e               lda ##1e        ;up arrow cursor (screen rep)
0124  046b  05 b7               ora ctrlfg      ;rvs if in for control mode
0125  046d  91 c4               sta (curadd),y  ;print cursor
0126  046f                ;
0127  046f                ;check for input from modem or keyboard
0128  046f                ;
0129  046f  2c 23 e8     main10 bit crb         ;srq interrupt flag?
0130  0472  10 03               bpl main20
0131  0474  4c fb 04            jmp getchr      ;if so, get char from modem
0132  0477  20 e4 ff     main20 jsr cgetl       ;keyboard input?
0133  047a  f0 f3               beq main10      ;if not, do nothing
0134  047c                ;
0135  047c                ;deal with keyboard input
0136  047c                ;
0137  047c  c9 03         key10  cmp ##03        ;stop key?
0138  047e  d0 0c               bne key20
0139  0480  a9 05               lda ##05        ;if so, close 5 and exit
0140  0482  20 e2 f2            jsr close
0141  0485  a9 18               lda #<donet     ;'off line'
0142  0487  a0 06               ldy #>donet
0143  0489  4c 1d bb            jmp strout
0144  048c                ;
0145  048c  c9 93         key20  cmp ##93        ;clear screen?
0146  048e  d0 06               bne key30
0147  0490  20 d2 ff            jsr outch       ;if so print without sending
0148  0493  4c 61 04            jmp main        ;and check for incoming chr
0149  0496                ;
0150  0496  a2 80         key30  ldx ##80
0151  0498  c9 12               cmp ##12        ;check for rvs on
0152  049a  f0 06               beq key40       ;if so set control flag
0153  049c                ;
0154  049c  c9 92               cmp ##92        ;check for rvs off
0155  049e  d0 07               bne key50
0156  04a0  a2 00               ldx ##00        ;if so clear control flag
0157  04a2  86 b7         key40  stx ctrlfg
0158  04a4  4c 61 04            jmp main        ;& check for chr in
0159  04a7                ;
0160  04a7  24 b7         key50  bit ctrlfg      ;is control on?
0161  04a9  10 12               bpl key60
0162  04ab  29 7f               and ##7f
0163  04ad  c9 41               cmp ##41        ;if so, letter?
0164  04af  90 47               bcc ignkey
0165  04b1  c9 60               cmp ##60
0166  04b3  b0 43               bcs ignkey      ;if not, ignore
0167  04b5  a2 00               ldx ##00        ;otherwise clear control flag
0168  04b7  86 b7               stx ctrlfg
0169  04b9  29 3f               and ##3f        ;mask bit 6
0170  04bb  d0 2e               bne sendch      ;& send (forced)
0171  04bd                ;
0172  04bd  c9 0d         key60  cmp ##0d        ;if not control, is it cr?
0173  04bf  f0 2a               beq sendch      ;if so, send it
0174  04c1  c9 1b               cmp ##1b        ;escape?
0175  04c3  f0 26               beq sendch      ;if so, send it
0176  04c5  c9 14               cmp ##14        ;delete?
0177  04c7  d0 04               bne key65
0178  04c9  a9 08               lda ##08        ;if so change to ctrl h
0179  04cb  d0 1e               bne sendch      ;and send (forced)
0180  04cd                ;
0181  04cd  c9 20         key65  cmp ##20        ;convert PETSCII to ASCII
0182  04cf  90 27               bcc ignkey
0183  04d1  c9 41               cmp ##41
0184  04d3  90 16               bcc sendch
0185  04d5  c9 5b               cmp ##5b
0186  04d7  b0 04               bcs key70
0187  04d9  09 20               ora ##20
0188  04db  d0 0e               bne sendch
0189  04dd  c9 60         key70  cmp ##60
0190  04df  90 0a               bcc sendch
0191  04e1  c9 c1               cmp ##c1
0192  04e3  90 13               bcc ignkey
0193  04e5  c9 db               cmp ##db
0194  04e7  b0 0f               bcs ignkey
0195  04e9  29 7f               and ##7f
0196  04eb  48            sendch pha             ;send character to modem
0197  04ec  a2 05               ldx ##05
0198  04ee  20 c9 ff            jsr coout
0199  04f1  68                  pla
0200  04f2  20 d2 ff            jsr outch
0201  04f5  20 cc ff            jsr clrch
0202  04f8  4c 61 04     ignkey jmp main        ;and check for char in.
0203  04fb                ;
0204  04fb                ;deal with incoming character
0205  04fb                ;
0206  04fb  ad 22 e8     getchr lda portb       ;clear srq flag
0207  04fe  a2 05               ldx ##05        ;obtain character
0208  0500  20 c6 ff            jsr coin
0209  0503  20 cf ff            jsr inchr
0210  0506  48                  pha
0211  0507  20 cc ff            jsr clrch
0212  050a  a4 b8               ldy arrcol      ;kill up-arrow cursor
0213  050c  a9 20               lda ##20
0214  050e  91 c4               sta (curadd),y
0215  0510  68                  pla
0216  0511  29 7f               and ##7f        ;kill any parity bit
0217  0513                ;
0218  0513  c9 08               cmp ##08        ;control h?
0219  0515  d0 08               bne getc40
0220  0517  a9 14               lda ##14        ;if so perform delete
0221  0519  20 d2 ff            jsr outch
0222  051c  4c 61 04            jmp main
0223  051f                ;
0224  051f  c9 0d         getc40 cmp ##0d        ;cr?
0225  0521  d0 0b               bne getc50
0226  0523  a0 00               ldy ##00        ;if so perform cr
0227  0525  84 c6               sty curcol
0228  0527  24 b9               bit lffg        ;line feeds expected?
0229  0529  10 07               bpl linefd      ;if not, perform line feed
0230  052b  4c 61 04            jmp main
0231  052e                ;
0232  052e  c9 0a         getc50 cmp ##0a        ;line feed?
```

```
0233  0530  d0 10           bne getc80
0234  0532  a9 11     linefd lda ##11        ;force scroll if necessary
0235  0534  20 d2 ff        jsr outch
0236  0537  20 d2 ff        jsr outch
0237  053a  a9 91           lda ##91
0238  053c  20 d2 ff        jsr outch
0239  053f  4c 61 04        jmp main
0240  0542                ;
0241  0542  c9 20     getc80 cmp ##20        ;convert ascii to screen
0242  0544  90 2f           bcc ignchr
0243  0546  c9 40           cmp ##40
0244  0548  d0 04           bne getc85
0245  054a  a9 00           lda ##00
0246  054c  f0 15           beq wrtchr     ;(forced)
0247  054e  c9 5b     getc85 cmp ##5b
0248  0550  90 11           bcc wrtchr
0249  0552  c9 60           cmp ##60
0250  0554  f0 1f           beq ignchr
0251  0556  b0 05           bcs getc90
0252  0558  29 bf           and ##bf
0253  055a  4c 63 05        jmp wrtchr
0254  055d  c9 7b     getc90 cmp ##7b
0255  055f  b0 14           bcs ignchr
0256  0561  29 1f           and ##1f
0257  0563  a4 c6     wrtchr ldy curcol      ;put chr on line
0258  0565  91 c4           sta (curadd),y
0259  0567  c8               iny
0260  0568  c0 28           cpy #linlen
0261  056a  d0 07           bne wrtc20
0262  056c  a0 00           ldy ##00
0263  056e  84 c6           sty curcol
0264  0570  4c 32 05        jmp linefd
0265  0573  84 c6     wrtc20 sty curcol
0266  0575  4c 61 04  ignchr jmp main
0267  0578                ;
0268  0578                ; PETSCII messages
0269  0578                ;
0270  0578  93          headng .byte $93,$0e,$12,'Simple '
0270  0579  0e
0270  057a  12
0270  057b  d3 49

0271  0582  c7 4c              .byte 'Glass Teletype',$0d
0271  0590  0d
0272  0591  11                .byte $11,'Ariadne Software Ltd.',$0d
0272  0592  c1 52
0272  05a7  0d
0273  05a8  11                .byte $11,'Do you expect line feeds? ',$00
0273  05a9  c4 4f
0273  05c3  00
0274  05c4  0d          waitt .byte $0d,$11,'Dial, obtain '
0274  05c5  11
0274  05c6  c4 49
0275  05d3  c3 d2              .byte 'CRX, then press any key.',$00
0275  05eb  00
0276  05ec  93          comt  .byte $93,$12,'On line.',$0d
0276  05ed  12
0276  05ee  cf 4e
0276  05f6  0d
0277  05f7  11                .byte $11,'For control-X use '
0277  05f8  c6 4f
0278  060a  d2 d6              .byte 'RVS then X.',$0d,$0d,$00
0278  0615  0d
0278  0616  0d
0278  0617  00
0279  0618  0d          donet .byte $0d,$12,'Off line.',$0d,$00
0279  0619  12
0279  061a  cf 46
0279  0623  0d
0279  0624  00
0280  0625                ;
0281  0625                .end

errors = 0000

symbol table

symbol value
 arrcol   00b8   basend   040b   cgetl   ffe4   chrset   e84c
 close    f2e2   clrch    ffcc   coin    ffc6   comt     05ec
 coout    ffc9   crb      e823   ctrlfg  00b7   curadd   00c4
 curcol   00c6   devno    00d4   donet   0618   fileno   00d2
 fnmlen   00d1   getc40   051f   getc50  052e   getc80   0542
 getc85   054e   getc90   055d   getchr  04fb   head10   0414
 head20   0423   headng   0578   ignchr  0575   ignkey   04f8
 inchr    ffcf   key10    047c   key20   048c   key30    0496
 key40    04a2   key50    04a7   key60   04bd   key65    04cd
 key70    04dd   lffg     00b9   linefd  0532   linlen   0028
 main     0461   main10   046f   main20  0477   open     f563
 outch    ffd2   portb    e822   secadd  00d3   sendch   04eb
 strout   bb1d   wait     042f   waitt   05c4   wrtc20   0573
 wrtchr   0563

end of assembly
```

Hex Dumps

```
40 column version
ready.

c*
      pc   irq  sr ac xr yr sp
.;  b780 e455 35 34 33 38 f8
.
.:  0400 00 0b 04 0a 00 9e 31 30
.:  0408 33 37 00 00 00 a9 78 a0
.:  0410 05 20 1d bb 20 e4 ff a2
.:  0418 00 c9 4e f0 06 a2 80 c9
.:  0420 59 d0 f1 86 b9 20 d2 ff
.:  0428 a9 c4 a0 05 20 1d bb 20
.:  0430 e4 ff f0 fb a9 05 85 d2
.:  0438 85 d4 a9 00 85 d3 85 d1
.:  0440 20 63 f5 a2 05 20 c6 ff
.:  0448 20 cf ff 20 cc ff ad 22
.:  0450 e8 a9 00 85 b7 a9 0e 8d
.:  0458 4c e8 a9 ec a0 05 20 1d
.:  0460 bb a5 c6 18 69 28 85 b8
.:  0468 a8 a9 1e 05 b7 91 c4 2c
.:  0470 23 e8 10 03 4c fb 04 20
.:  0478 e4 ff f0 f3 c9 03 d0 0c
.:  0480 a9 05 20 e2 f2 a9 18 a0
.:  0488 06 4c 1d bb c9 93 d0 06
.:  0490 20 d2 ff 4c 61 04 a2 80
.:  0498 c9 12 f0 06 c9 92 d0 07
.:  04a0 a2 00 86 b7 4c 61 04 24
.:  04a8 b7 10 12 29 7f c9 41 90
.:  04b0 47 c9 60 b0 43 a2 00 86
.:  04b8 b7 29 3f d0 2e c9 0d f0
.:  04c0 2a c9 1b f0 26 c9 14 d0
.:  04c8 04 a9 08 d0 1e c9 20 90
.:  04d0 27 c9 41 90 16 c9 5b b0
.:  04d8 04 09 20 d0 0e c9 60 90
.:  04e0 0a c9 c1 90 13 c9 db b0
.:  04e8 0f 29 7f 48 a2 05 20 c9
.:  04f0 ff 68 20 d2 ff 20 cc ff
.:  04f8 4c 61 04 ad 22 e8 a2 05
.:  0500 20 c6 ff 20 cf ff 48 20
.:  0508 cc ff a4 b8 a9 20 91 c4
.:  0510 68 29 7f c9 08 d0 08 a9
.:  0518 14 20 d2 ff 4c 61 04 c9
.:  0520 0d d0 0b a0 00 84 c6 24
.:  0528 b9 10 07 4c 61 04 c9 0a
.:  0530 d0 10 a9 11 20 d2 ff 20
.:  0538 d2 ff a9 91 20 d2 ff 4c
.:  0540 61 04 c9 20 90 2f c9 40
.:  0548 d0 04 a9 00 f0 15 c9 5b
.:  0550 90 11 c9 60 f0 1f b0 05
.:  0558 29 bf 4c 63 05 c9 7b b0
.:  0560 14 29 1f a4 c6 91 c4 c8
.:  0568 c0 28 d0 07 a0 00 84 c6
.:  0570 4c 32 05 84 c6 4c 61 04
```

```
.:  0578 93 0e 12 d3 49 4d 50 4c
.:  0580 45 20 c7 4c 41 53 53 20
.:  0588 d4 45 4c 45 54 59 50 45
.:  0590 0d 11 c1 52 49 41 44 4e
.:  0598 45 20 d3 4f 46 54 57 41
.:  05a0 52 45 20 cc 54 44 2e 0d
.:  05a8 11 c4 4f 20 59 4f 55 20
.:  05b0 45 58 50 45 43 54 20 4c
.:  05b8 49 4e 45 20 46 45 45 44
.:  05c0 53 3f 20 00 0d 11 c4 49
.:  05c8 41 4c 2c 20 4f 42 54 41
.:  05d0 49 4e 20 c3 d2 d8 2c 20
.:  05d8 54 48 45 4e 20 50 52 45
.:  05e0 53 53 20 41 4e 59 20 4b
.:  05e8 45 59 2e 00 93 12 cf 4e
.:  05f0 20 4c 49 4e 45 2e 0d 11
.:  05f8 c6 4f 52 20 43 4f 4e 54
.:  0600 52 4f 4c 2d d8 20 55 53
.:  0608 45 20 d2 d6 d3 20 54 48
.:  0610 45 4e 20 d8 2e 0d 0d 00
.:  0618 0d 12 cf 46 46 20 4c 49
.:  0620 4e 45 2e 0d 00 aa aa aa
.
.
ready.
```

80 column version

```
ready.

c*
      pc   irq   sr ac xr yr sp
.;  b780 e455 35 34 33 38 fa
.
.:  0400 00 0b 04 0a 00 9e 31 30
.:  0408 33 37 00 00 00 a9 78 a0
.:  0410 05 20 1d bb 20 e4 ff a2
.:  0418 00 c9 4e f0 06 a2 80 c9
.:  0420 59 d0 f1 86 b9 20 d2 ff
.:  0428 a9 c4 a0 05 20 1d bb 20
.:  0430 e4 ff f0 fb a9 05 85 d2
.:  0438 85 d4 a9 00 85 d3 85 d1
.:  0440 20 63 f5 a2 05 20 c6 ff
.:  0448 20 cf ff 20 cc ff ad 22
.:  0450 e8 a9 00 85 b7 a9 0e 8d
.:  0458 4c e8 a9 ec a0 05 20 1d
.:  0460 bb a5 c6 18 69 50 85 b8
.:  0468 a8 a9 1e 05 b7 91 c4 2c
.:  0470 23 e8 10 03 4c fb 04 20
.:  0478 e4 ff f0 f3 c9 03 d0 0c
.:  0480 a9 05 20 e2 f2 a9 18 a0
.:  0488 06 4c 1d bb c9 93 d0 06
.:  0490 20 d2 ff 4c 61 04 a2 80
.:  0498 c9 12 f0 06 c9 92 d0 07
.:  04a0 a2 00 86 b7 4c 61 04 24
.:  04a8 b7 10 12 29 7f c9 41 90
.:  04b0 47 c9 60 b0 43 a2 00 86
.:  04b8 b7 29 3f d0 2e c9 0d f0
.:  04c0 2a c9 1b f0 26 c9 14 d0
.:  04c8 04 a9 08 d0 1e c9 20 90
.:  04d0 27 c9 41 90 16 c9 5b b0
.:  04d8 04 09 20 d0 0e c9 60 90
.:  04e0 0a c9 c1 90 13 c9 db b0
.:  04e8 0f 29 7f 48 a2 05 20 c9
.:  04f0 ff 68 20 d2 ff 20 cc ff
.:  04f8 4c 61 04 ad 22 e8 a2 05
.:  0500 20 c6 ff 20 cf ff 48 20
.:  0508 cc ff a4 b8 a9 20 91 c4
.:  0510 68 29 7f c9 08 d0 08 a9
.:  0518 14 20 d2 ff 4c 61 04 c9
.:  0520 0d d0 0b a0 00 84 c6 24
.:  0528 b9 10 07 4c 61 04 c9 0a
.:  0530 d0 10 a9 11 20 d2 ff 20
.:  0538 d2 ff a9 91 20 d2 ff 4c
.:  0540 61 04 c9 20 90 2f c9 40
.:  0548 d0 04 a9 00 f0 15 c9 5b
.:  0550 90 11 c9 60 f0 1f b0 05
.:  0558 29 bf 4c 63 05 c9 7b b0
.:  0560 14 29 1f a4 c6 91 c4 c8
.:  0568 c0 50 d0 07 a0 00 84 c6
.:  0570 4c 32 05 84 c6 4c 61 04
.:  0578 93 0e 12 d3 49 4d 50 4c
.:  0580 45 20 c7 4c 41 53 53 20
.:  0588 d4 45 4c 45 54 59 50 45
.:  0590 0d 11 c1 52 49 41 44 4e
.:  0598 45 20 d3 4f 46 54 57 41
.:  05a0 52 45 20 cc 54 44 2e 0d
.:  05a8 11 c4 4f 20 59 4f 55 20
.:  05b0 45 58 50 45 43 54 20 4c
.:  05b8 49 4e 45 20 46 45 45 44
.:  05c0 53 3f 20 00 0d 11 c4 49
.:  05c8 41 4c 2c 20 4f 42 54 41
.:  05d0 49 4e 20 c3 d2 d8 2c 20
.:  05d8 54 48 45 4e 20 50 52 45
.:  05e0 53 53 20 41 4e 59 20 4b
.:  05e8 45 59 2e 00 93 12 cf 4e
.:  05f0 20 4c 49 4e 45 2e 0d 11
.:  05f8 c6 4f 52 20 43 4f 4e 54
.:  0600 52 4f 4c 2d d8 20 55 53
.:  0608 45 20 d2 d6 d3 20 54 48
.:  0610 45 4e 20 d8 2e 0d 0d 00
.:  0618 0d 12 cf 46 46 20 4c 49
.:  0620 4e 45 2e 0d 00 aa aa aa
.
.
ready.

sys 4 to monitor, enter program
as listed, then
.s "0:teletype",08,0401,0625
```

T.A.L.
COMPUTER DIVISION

AN INTELLIGENT CHIP

D.O.S. commands as per Universal Wedge program, and a sequential file reader.

?INTELLIGENT CHIP? Yes. The same chip will work in $9000, $A000 and $B000 locations with Pet Basic 2 and up.

£20 C.W.O.

11 High Street, Leighton Buzzard, Beds.
Tel: (0525) 372114

STOP PRESS: Amateur radio operators. R.T.T.Y. plug-in module for Pets and Vics. S.A.E. for further details.

Your move

Commodore CBM systems are dependable, efficient, versatile, easy to maintain. That's why so many companies, big and small make their move through

ALPHA Business Systems

Electron House
Industrial Estate
Church Street
Ware SG12 9ES

Telephone Ware (0920) 68926/7

from small systems.....

PET HARD DISKS

The Small Systems HARDBOX acts as an intelligent controller for up to Corvus Winchester drives.

- PET DOS 1 and 2 compatibility
- Multi user capability on Corvus Drives
- 16 Megabyte max file size
- 65535 max records per relative file
- Over 2000 files on 5 MB drive

Price..... £495

PET MINI-WINI

The MW-1000 Mini-Winchester is a compact desk-top unit that just plugs into the PET – gives you up to 12 Mb of hard disk storage under CP/M or PET DOS or both!!

At the flick of a switch, this amazing unit allows you to have:-

- The whole disk under CP/M (plus 60K of RAM and Z80)
- The whole disk under PET DOS
- Half of the disk under PET DOS and half under CP/M!!

No changes to the PET or ROMS are required – just plug in and go.

3Mb..... £2538
6Mb..... £2837
12Mb..... £3360

Prices include CP/M Utilities, 60K RAM and Z80.

PET CP/M

SOFTBOX and CP/M SOFTWARE

SOFTBOX allows the PET to run the worlds most popular operating system for micros. Operates with PET floppies and/or a Hard disk system.

Comprehensive range of CP/M software available – ask for our Catalogue.

Softbox (with RS232 + Hard disk interfaces as standard) ..... £495 *new low price!*

IEEE-488

IEEE-488 SERIAL INTERFACE TYPE C..... £120

IEEE-488 SERIAL INTERFACE TYPE B300..... £186
40 char input buffer

IEEE-488 ADDRESSABLE PARALLEL INTERFACE TYPE A100..... £106

TYPE G.P.I. AP MICROPROCESSOR BASED IEEE-488 BI DIRECTIONAL INTERFACE..... £275
IK input buffer standard 59K max.

TV/VIDEO MONITOR INTERFACES..... £46

RS 232C TO 20mA CURRENT LOOP ADAPTER..... £17.50

PETSPEED

Optimising Basic Compiler..... £240

DEVELOPMENT TOOLS

S10/CP/M

8048 family in circuit emulator..... £550
8748 programmer..... £395
Prom emulator..... £295
Range of cross assemblers for most popular micros..... £95
8048 BASIC compiler..... £195

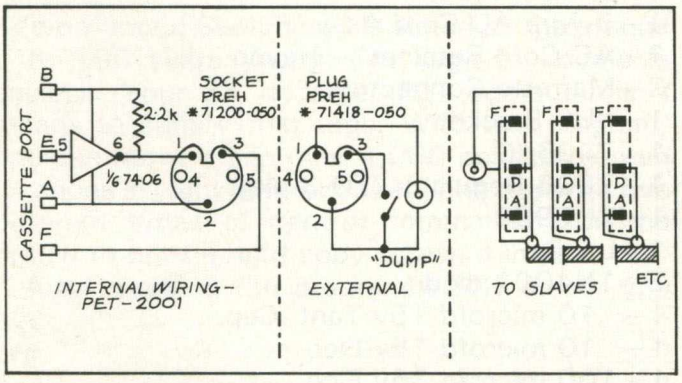
COMPLETE PET and HP-85 SYSTEMS

small systems engineering limited
2-4 Canfield Place, London NW6 3BT. Telephone: 328 7145 Telex 264538

Interfacing

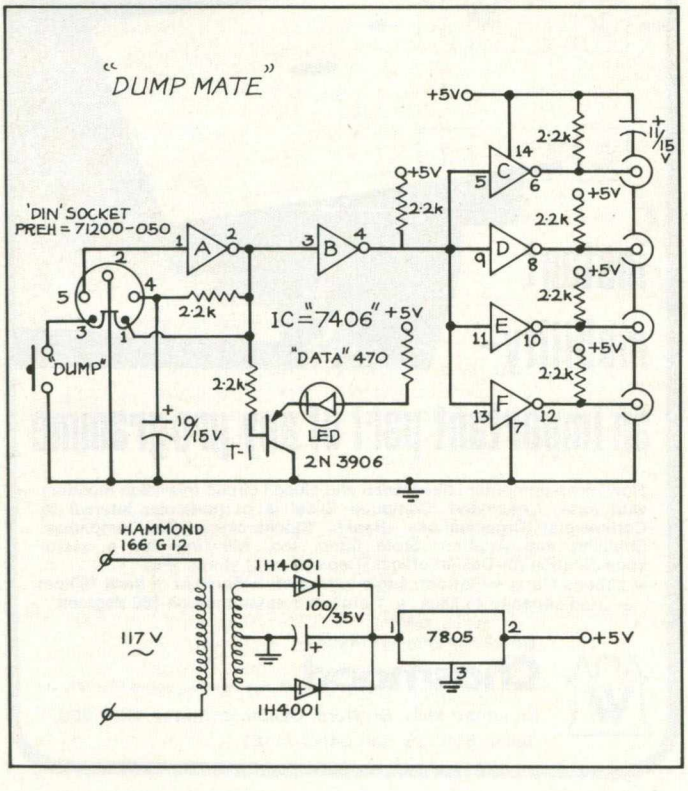
Dump Mate: A Multi-Load System

The original multi-load system was part of our AV-8101 video-audio interface for the Commodore 2000 series computers, as shown below.

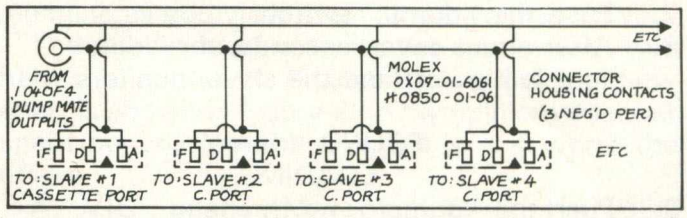


By means of the spare inverter-driver on this board, programs could be dumped from the master computer to about twenty slave units. In order to increase its capability to load programs to up to sixty slaves, when so required, the first "Dump-Mate", a multi-output driver, was built.

However, with the introduction of the Commodore 8032 and 4032 (12" screen), the multi-load system used in the 2001 was no longer possible, as all six inverters of the 7406 I.C. were now required for the video interface. This problem was overcome by the redesign of the "Dump-Mate" into a self-contained, external type multi-loader.



Each of the four outputs can be connected to up to twenty "slave" computers by means of the cassette-ports interface assembly shown below.



Connection between the input of the Dump-Mate and the output of the master computer is made by a short length of five-conductor cable with "DIN" plugs (PREH = 71418-50) on both ends.

The output socket at the computer end is wired as per diagram below:

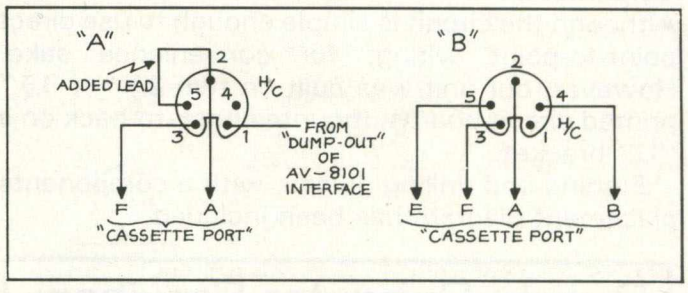
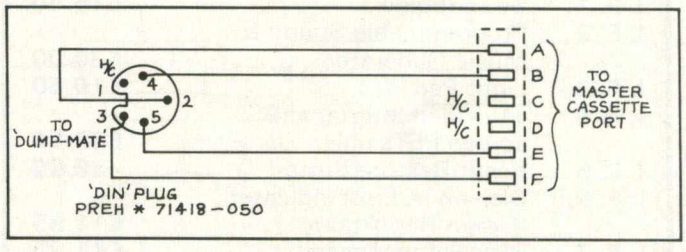


Figure "A" is used for PET 2001 series with the AV-8101 interface and dump circuit, while figure "B" is the wiring required for use with the regular 2000, 4000 and 8000 series computers.

Another way of connecting the Dump-Mate to the computer is shown below:



In this manner, any PET computer can be utilized as the master unit, however, the cassette port will not be available for program loading.

The following is a short "how to" guide:

1. Be sure that the power to all equipment is OFF before connecting or disconnecting cables.
2. When everything is in place, switch on all units, including the Dump-Mate.
3. LOAD a program into the master computer.
4. The slave computers requiring this program should now type:

   NEW return
   LOAD return

5. The monitors of these units should now show:
   SEARCHING
6. On the master unit, type:
   SAVE ''name'' return
7. Push the ''dump'' switch.
8. After about seven seconds, the ''data'' light will go off and the slave monitors will show:
   FOUND ''name''
   LOADING
9. Push the ''dump'' switch again.
10. The ''data'' light will stay on until the program is loaded, at which time READY. and flashing cursor should appear on all monitors.
11. Typing RUN return will execute the program.

Construction

Although the circuit is simple enough to use direct point-to-point wiring, for convenience sake. However, our unit was built on two $2\frac{1}{2}'' \times 1\frac{3}{4}''$ printed circuit boards, mounted back-to-back on a ''U''-bracket.

Etching and drilling guides, with a components placement diagram has been included.

1—Hammond 1454G Case
1—Hammond 166G12 Transformer
1—Preh 71200-050 Socket
4—Switchcraft 3501-FP Connectors
1—N/O pushbutton — Grayhill
1—L.E.D. Mount
1—3-wire AC Cord Assy.
1—AC Cord Retainer — Heyco
2—Marrette Connectors
1—''U'' Bracket
1—7406 IC
1—7805 Regulator (TO-3 pkg)
1—2N3906
1—L.E.D.
2—1N4001 diode
1— 10 microfd 15v Tant. Cap.
1— 10 microfd 15v Elco
1—100 microfd 35v Elco
1—470 ohm resistor
6—2200 ohm resistor
1—22 K-ohm resistor
—Miscellaneous Mounting Hardware

Editor's Note

Dump-Mate was built originally for PET/CBMs, but it will also work with the VIC-20 since the cassette interface is identical to the PETs.

Qwerty Computer Services

20 Worcester Road, Newton Hall, Durham

Tel. (0385) 67045

PET PRODUCTS

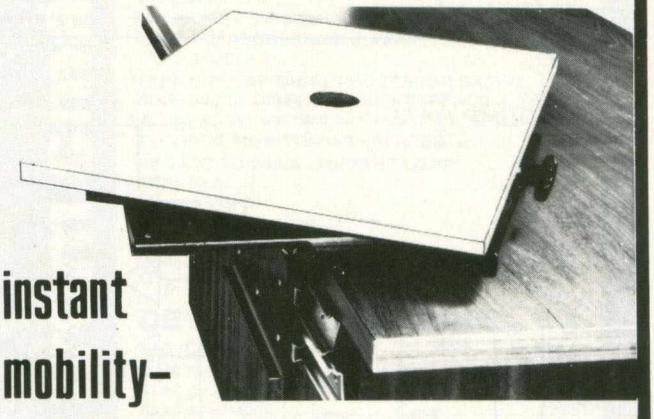
I. F. 1	Soundbox	£16.50
I. F. 2	Programmable Sound & Music Generator	£30.00
I. F. 3	Light Pen	£16.50
I. F. 4	TV/Video Interface (9'' screen PETs only)	£33.50
I. F. 5	Reset/Restore Button	£8.80
I. F. 6	Disk on + Error indicator (Green/Red/Audio)	£11.95
I. F. 7	Disk Safety Device	£11.95
I. F. 8	E–Socket Rom Expansion	£12.95
	Basic 4 Extramon to fit above	£8.95
I. F. 10	EPROM Burner	£35.00
I. F. 11	ROM n' RAM 2K	£25.00
	4K	£30.00
	Battery back-up 4	£6.00
I. F. 14	Extended Basic Rom (2 ROM set)	£30.00
I. F. 15*	Switch Unit	£11.00

* Also available for Vic — £5 per switch.

ADD £1.40 to cover post & packing add 15% VAT: S.A.E. for catalogue.
BARCLAYCARD AND ACCESS ORDERS ACCEPTED
(Also after 6.00 p,m, and weekends)
We reserve the right to alter prices, appearance and specification at any time.

Chasmood ZIEHL

COMPUTER SLIDE



instant mobility–

an important part of any programme

Now move computer linked video and closed circuit television monitors with ease. Chasmood 'Computer Slide' is of particular interest to Commercial Organisations, Banks, Stockbrokers, Oil Companies, Shipping and Aviation. Store fitters too, will find this a useful consideration for Design offices. Departmental stores, etc.
★ 2 metres long ★ Platform angle adjustable ★ Forward or back 100mm.
★ Load capacity 45 kilos ★ Turntable rotates through 360 degrees.

Details on request from:

Chasmood (A division of Vulcanised Fibre (TR) Ltd.)
Broadford Mills, Shalford, Guildford, Surrey, GU4 8EG.
Telex: 859229 Tel: 0483 77161

Sound and Vision

Two Handed Sketching

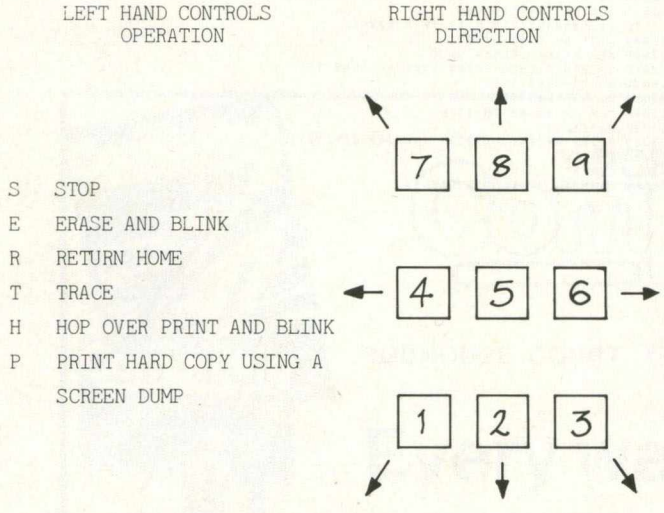
"Two Handed Sketching" is aimed at age groups 5 to 105 years young. The urge to doodle is ageless. Your 8K or larger PET computer is a means to satisfy that urge. With this program, one can move a pen over 4,000 possible screen locations in eight different directions and with five different types of control commands plus the option to print a hard copy screen dump.

After working the program a few times, you suddenly realize you have not scratched the surface of its possibilities. Cubics, curves, figures within figures, dot drawings, faces, machines, chemical, biological, mathematical, graphical, geographic, and just abstract forms that take on meaning as you view them all crowd in upon you.

The principles of drawing the screen with two hands is shown in Figure 1. A simplified flow chart is shown in Figure 2. In my opinion, the hour spent punching in the program is well worth the effort.

A VIC adaptation of "TWO HANDED SKETCHING" is a straight-forward conversion. With a little extra effort, the addition of colour would add spice.

Figure 1



The display screen is 80 columns by 50 lines.

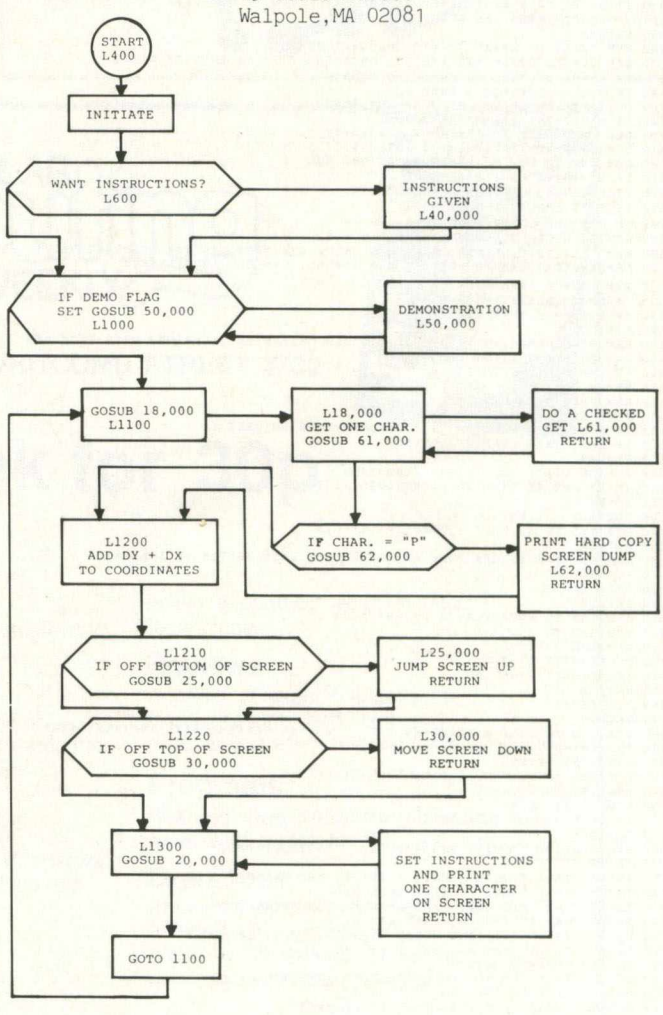
The character printed is a square white dot (4x4 pixels) that is $\frac{1}{4}$ of the area of the standard character.

When the right had directs the dot to move, it will continue moving until the left hand gives the order STOP (S) or the right hand issues a new direction order.

Lines 50005-50015 can be altered so that the demonstration program automatically draws a sketch you have designed without use of the control keys.

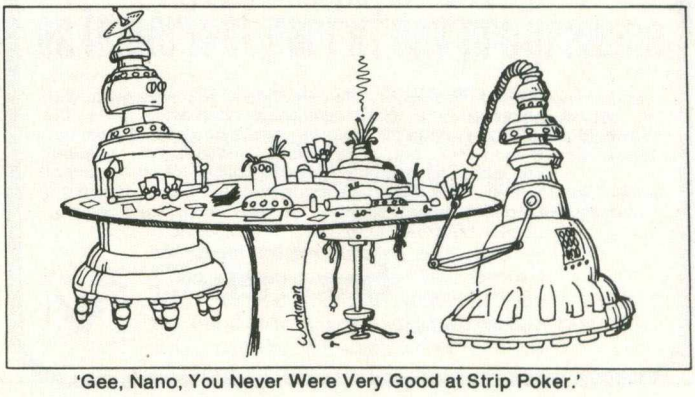
Figure 2

Two Handed Sketching
Preston F.Marshall
19 Gould Street
Walpole,MA 02081



Sound and Vision

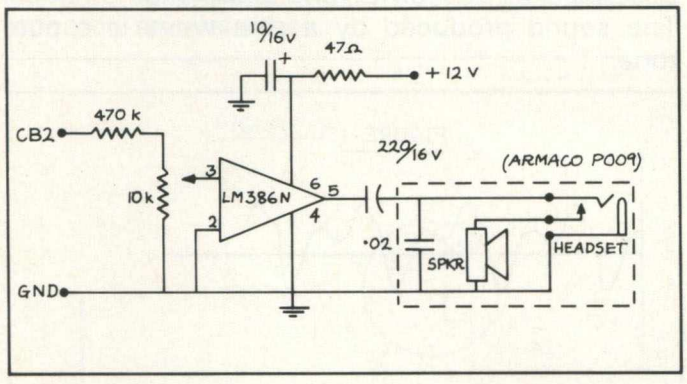
```
400 KB=158:REM KEYBOARD COUNTER NEW ROMS
410 IF PEEK(51234)=0 THEN KB=525:REM OLD ROMS
420 DATA 64,0,128,64,-64,-128,0,-64
430 FOR I=0 TO 7:READ C(I):NEXT I:REM  SETS UP DATA FOR PRINT OUT
500 REM**TWO HANDED SKETCHING******
510 REM**BY PRES MARSHALL**********
511 REM**191 GOULD ST,WALPOLE,MA***
512 REM**02081**********************
513 REM*COPYRIGHT 1982**************
520 REM*******************************
600 PRINT"DO YOU WANT INSTRUCTIONS?"
610 PRINT:PRINT "HIT [Y] FOR YES"
620 PRINT:PRINT "HIT [N] FOR NO"
630 POKE KB,0:WAIT KB,1:GETA$:IF A$="" THEN 630
640 IF A$="Y" THEN A$="":DE=1:GOTO 40000
650 IF A$="N" THEN A$="":   GOTO 700
670 A$="":PRINT "TRY AGAIN":GOTO 630
700 DIM SY(15):REM SYMBOL ARRAY
705 DIM DR$(1)
710 PRINT""
720 PRINT""
800 POKE 59468,12
810 DATA 32,123,126,97,108,98,127,252,124,255,226,236,225,254,251,160
820 FOR I=0 TO 15
830 READ SY(I):REM SET UP SYMBOL ARRAY
840 NEXT I
900 PY%=0:PX%=0
1000 REM START OF PROGRAM
1010 IF DE=1 THEN GOTO 50000:REM IF DEMO FLAG IS SET GO TO 50000
1100 GOSUB 18000
1200 PX%=PX%+DX:PY%=PY%+DY
1210 IF PY%>49 THEN GOSUB 25000
1220 IF PY%<0 THEN GOSUB 30000
1300 GOSUB 20000
1400 GOTO 1100
18000 REM SET UP A SET OF DIRECTIONS TO MOVE THE TARGET DOT WITH A GET
18100 VA$="64283719RSETHP":REM VA=VALID RESPONSES TO A SINGLE LETTER  GET
18110 IF PEEK(KB)=0 THEN GOTO 18240:REMIF NOTHING IN KEYBOARD BUFFER GO ON
18120 GOSUB 61000:REM DO A CHECKED GET
18130 IF C$="6"THEN DX=1:DY=0:REMRIGHT
18140 IF C$="4"THEN DX=-1:DY=0:REMLEFT
18150 IFC$="2"THEN DX=0:DY=1:REMDOWN
18160 IFC$="8" THEN DY=-1:DX=0:REM UP
18170 IFC$="3" THEN DX=1:DY=1:REM SE
18180 IFC$="7" THEN DX=-1:DY=-1:REM NW
18190 IFC$="1" THEN DX=-1:DY=1:REMSW
18200 IFC$="9" THEN DX=1:DY=-1:REM NE
18210 IF C$="T" THEN ER=0:HP=0:REM SETS DOT TO LEAVE ITS TRACE
18220 IF C$="R" THEN PX%=0:PY%=0:DY=0:DX=0:REM SET DOT TO ZERO POSITION
18230 IF C$="E" THEN ER=1:HP=0:REM SETS DOT TO ERASE AS IT MOVES
18240 IF C$="S" THEN GOTO 35000:REM GO TO A FLASHIND DOT SUB
18250 IFC$="H" THEN HP=1:ER=0: REMARK SETS FLAG TO HOP OVER EXISTING DOTS
18260 IF C$="P" THEN GOSUB 62000
18900 RETURN
20000 REM START OF ARRAY TO SET QUADRANT SYMBOLS
21000 REM SUB TO PRINT PX% AND PY% ON A 50 BY 80 COORDINATE GRAPH
21050 RO=2:REM RO= ROW OF QUADRANT SYMBOL
21060 IF PY%/2=INT(PY%/2) THEN RO=1
21100 CO=2:REM CO=COLUMN OF QUADRANT SYMBOL
21110 IF PX%/2=INT(PX%/2) THEN CO=1
21200 REM CHARACTER POSITION ON SCREEN=CP
21210 CP=32768+40*(INT(PY%/2))+INT(PX%/2)
21300 REM SET UP NEW QUADRANT LOCATIONS NQ
21310 IF RO=1 AND CO=1 THEN NQ=2
21320 IF RO=1 AND CO=2 THEN NQ=8
21330 IF RO=2 AND CO=1 THEN NQ=1
21340 IF RO=2 AND CO=2 THEN NQ=4
21350 REM IN ABOVE QUADRANTS READ 1,2,4,8
21500 REM PEEK TO FIND OLD CHARACTER VALUE AND THEN ADD NEW QUADRANT VALUE
21510 PK=PEEK(CP):REM PK =PEEKED OLD SYMBOL. OV= OLD ARRAY VALUE
21530 REM SEARCH ARRAY FOR
61405 ´#NOTSGNENDZWAIT*% I=0 TO 15
21550 IF SY(I)=PK THEN OV=I:GOTO 21600
21560 NEXT I
21600 REM NEW CHARACTER VALUE (NV) IS OLD CHARACTER VALUE(OV) ORED WITH NEW
21610 REM QUADRANT VALUE (NQ).
21615 IF ER=1 THEN GOTO 21710
21620 NV=OV OR NQ
21640 REM POKE NEW CHARACTER
21650 POKE(CP),SY(NV)
21655 IF HP=1 THEN POKE(CP),SY(0):FOR I=1 TO 40:NEXT I
21660 IF HP=1 THEN POKE(CP),SY(OV)
21670 RETURN
21700 REMARK CARRY OUT AN ERASE ROUTINE
21710 IF OV AND NQ THEN NV=OV-NQ:GOTO 21780
21720 NV=OV
21780 POKE(CP),SY(NQ)
21790 FOR I=0 TO 40:NEXT I:POKE(CP),SY(LV)
21800 RETURN
25000 REM SUB FOR SCREEN JUMP AS MOVING DOT HITS BOTTOM OF SCREEN
25010 PRINT
25020 PY%=PY%-2
25030 RETURN
30000 REM SUB TO MOVE SCREEN UP ONE LINE
30010 FOR I=0 TO 959
30020 MK=PEEK(33727-I)
30030 POKE(33727-I),32
30040 POKE(33727-I+40),MK
30050 NEXT I
30060 PY%=PY%+2
30070 RETURN
35000 REM SUB TO HOLD A FLASHING DOT    ON THE SCREEN UNTIL ANOTHER ENTRY
35010 DX=0:DY=0:
35020 POKE(CP),SY(NQ)
35030 FOR I=0 TO 40:NEXT I:POKE (CP),SY(0)
35040 FOR I=0 TO 40:NEXT I
35050 POKE (CP),SY(NV)
35060 IF PEEK(525)=0 THEN GOTO 35020
35070 RETURN
40000 REM ROUTINE TO GIVE PROGRAM USE   INSTRUCTIONS.
40005 PRINT ""
40010 PRINT "THIS PROGRAM MOVES A DOT BY THE USE OF"
40020 PRINT " TWO HANDS TO DRAW SKETCHES
40030 PRINT "THE LEFT HAND IS HELD OVER THE KEYBOARD"
40040 PRINT "CHARACTERS [S,E,R,T AND H]"
40050 PRINT "THE RIGHT HAND WORKS WITH [1,2,3,4,6,7,8,9,]"
40060 PRINT "S=STOP DOT MOVEMENT"
40070 PRINT "E=ERASE DOTS"
40080 PRINT "R=RETURN DOT TO UPPER LEFT SCREEN"
40090 PRINT "T=TRACE DOTS ON SCREEN"
40095 PRINT "H=HOP OVER THE EXISTING TRACES"
40100 PRINT:PRINT:PRINT"FOR THE RIGHT HAND THE NUMBER 5 IS"
40110 PRINT "CONSIDERED THE HUB.THE NUMBERS 1 TO 9"
40120 PRINT "RADIATE OUT FROM THIS HUB TO MOVE THE "
40130 PRINT "DOT OUTWARD FROM THIS HUB IN ANY OF 8"
40140 PRINT "DIRECTIONS"
40200 PRINT:PRINT:PRINT "TO GO TO NEXT INSTRUCTION HIT ANY KEY"
40400 POKE KB,0: WAIT KB,1:GET A$:IF A$="" THEN GOTO 40400
40410 A$=""
40420 GOTO 40500
40500 PRINT"DOUBLE KEYING"
40510 PRINT:PRINT "HOLD ONE KEY DOWN AND INTERMITTANTLY "
40520 PRINT " PRESS ANOTHER KEY "
40530 PRINT:PRINT "EXAMPLE:HOLD DOWN E  [ERASE KEY]"
40540 PRINT "AND INTERMITTANTLY PRESS T [TRACE KEY]"
40550 PRINT "IF THE DOT IS MOVING SPACED DOTS WILL"
40560 PRINT "WILL BE LEFT ON THE SCREEN."
40570 PRINT:PRINT:PRINT "EXPERIMENT TRY OTHER COMBINATIONS."
40580 PRINT"SOME WILL WORK TOGHETHER SOME WILL NOT."
40590 PRINT "DEVELOP YOUR OWN TECHNIQUE."
40595 PRINT:PRINT:PRINT "HIT P FOR HARD COPY"
40600 PRINT:PRINT:PRINT "TO START HIT ANY KEY
40900 POKE KB,0:WAITKB,1:GETA$:IF A$=""THEN GOTO40900
40910 A$="":GOTO 700
50000 REM SUB TO RUN DEMONSTRATION PROGRAMS
50005 DR$="R333333333336669998887774441112222333333669988774441122336666666"
50010 DR$=DR$+"RE22222222222222T333222222222688888889963222222336669988912226"
50015 DR$=DR$+"6669988844426666999889999111122222233669E888877T777"
50020 PRINT "  DEMO      LAST KEY PRESSED="
50030 FOR K=1 TO LEN(DR$)
50040 C$=MID$(DR$,K,1)
50050 DD=ASC(C$)
50060 IF DD>64 THEN DD=DD-64
50070 POKE(32798),DD
50080 GOSUB 18130:REMARK PLOT
50082 PX%=PX%+DX:PY%=PY%+DY
50085 GOSUB 20000:REM PLOT
50090 NEXT K
50100 PRINT"TO GO ON HIT ANY KEY"
50105 POKE(KB),0:WAIT(KB),1:GET A$:IF A$="" THEN GOTO 50105
50120 PRINT "WANT TO SEE IT AGAIN?"
50130 PRINT "HIT [Y] FOR YES OR [N] FOR NO"
50140 POKE(KB),0:WAIT(KB),1:GET A$:IF A$="" THEN GOTO 50140
50150 IF A$="Y" THEN DE=1:REM SET DEMONSTRATOR FLAG TO 1
50160 IF A$="N" THEN DE=0:REM RESET DEMO FLAG
50170 IF A$<>"Y" AND A$<>"N" THEN PRINT"TRY AGAIN":GOTO 50130
50180 PY%=0:PX%=0
50190 PRINT ""
50210 DY=0:DX=0
50220 GOTO 1000
61000 REM SUB TO DO A CHECKED GET
61010 T1=TI
61020 CC$=C$
61030 GETC$:IFC$="" AND (TI-T1)<10 THEN 61030
61040 IF C$<>"" THEN GOTO 61060
61050 IF C$="" AND (TI-T1)>10 THEN C$=CC$:RETURN
61060 FOR G=1 TO LEN(VA$)
61070 IF C$=MID$(VA$,G,1) THEN RETURN
61080 NEXT G: GOTO 61030
62000 REM AXIOM SCREEN DUMP
62010 OPEN 4,4:CMD4:PRINT CHR$(8)+CHR$(15);
62020 R$(0)=CHR$(146)
62030 R$(1)=CHR$(18)
62040 FOR I= 32768 TO 33767
62050 P=PEEK(I)
62060 PRINT R$(INT(P/128)) +CHR$(P+C(P/32));
62070 NEXT I
62075 C$="R"
62080 PRINT#4,"":CLOSE4:RETURN
```

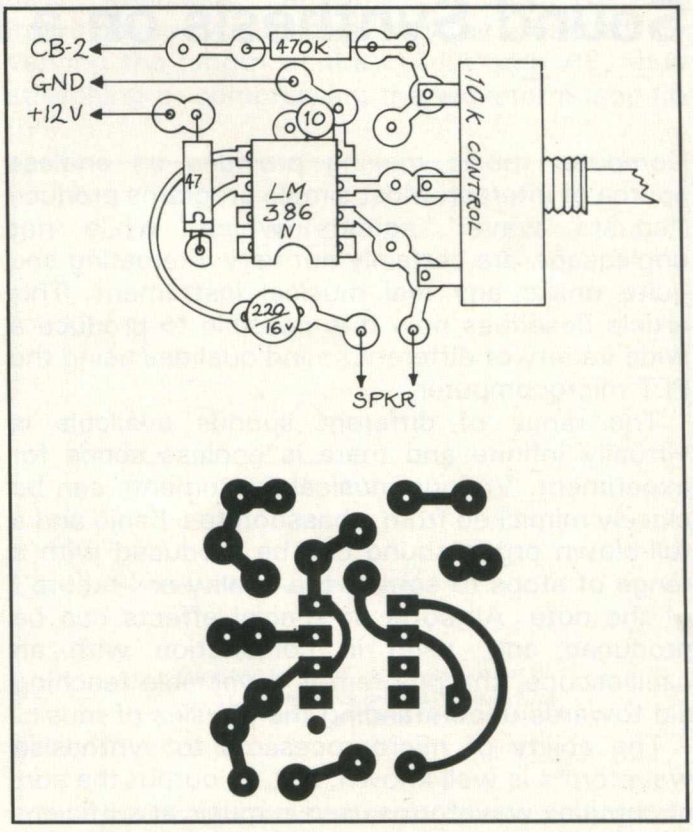


'Gee, Nano, You Never Were Very Good at Strip Poker.'

CB2 Amplifier

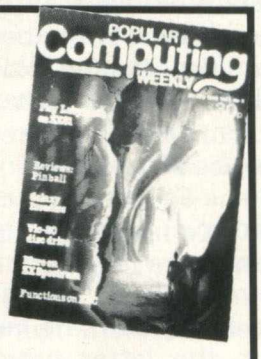
This tidy little circuit came from Ted Evers of Toronto. Connect it to the User Port CB2 line, ground, and one of the 12 volt pins inside the machine, and you've got CB2 sound (with optional headphones jack to prevent raging parents, teachers and wives).





POPULAR Computing WEEKLY

HOBHOUSE COURT, 19 WHITCOMB STREET WC2



Every week for 30p

Available at your newsagent – NOW

Competitions
To tease and squeeze the best out of your rapidly developing computer brain.

Programming
To help you build your knowledge of sophisticated techniques. Step by step the secrets of the pros will be unfolded.

Clubs
What's happening inside Britain's mushrooming micro club world. Interviews and profiles of the new computing breed.

News
A really fast up-to-the-minute analysis of all that's going on in the expanding world of micro technology.

Reviews
A panel of experts assess the latest products both soft and hard as and when they become available.

Programs
Readers' own programs published each week with prizes for the best entries.

Questions
Answered by an expert who knows . . . *In short, it's a package you can't afford to be without. Make sure, place an order with your newsagent.*

Programming Tips

Sound Synthesis on a Pet

Computer music making provides an endless source of interest. Most simple programs produce ''square wave'' sounds which, while not unpleasant, are certainly not very interesting and quite unlike any real musical instrument. This article describes how it is possible to produce a wide variety of different sound qualities using the PET microcomputer.

The range of different sounds available is virtually infinite and there is endless scope for experiment. Various musical instruments can be closely mimicked from a bassoon to a banjo and a full-blown organ sound can be produced with a range of stops to control the quality or ''timbre'' of the note. All sorts of special effects can be produced and, used in conjunction with an oscilloscope, the program is a valuable teaching aid towards understanding the physics of music.

The ability of microprocessors to synthesise waveforms is well known, but, to output the sort of complex waveforms used in music at sufficient frequency, an array of numerical values must be produced in advance — a tedious and time-consuming task. However, the formidable combination of a powerful high-level language such as BASIC with the speed of machine language subroutines, makes it possible to produce an enormous variety of complex waveforms with a good range of audio frequencies.

The result is a ''music machine'' which can be played either from the keyboard or under program control, the latter providing scope for playing intricate combinations of notes at dazzling speed. It is also possible through the use of random numbers and some control routines for the computer to compose and play his own music.

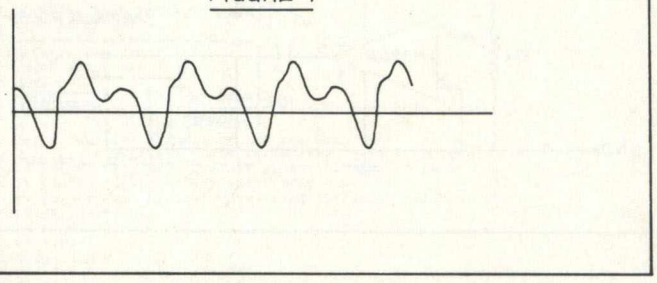
An 8 bit digital to analogue (D/A) convertor will be required for connection to the user port e.g. I/C ZN425E available from Radiospares. A good sound can be obtained by feeding the output direct to an audio amplifier although further refinement is possible through the use of tone filters.

Physics of Music

The waveforms produced by a musical instrument can be extremely complex but a simple example, that of a flute, is shown in fig (i). Each instrument has its unique waveform and it is this which gives it a particular quality or timbre.

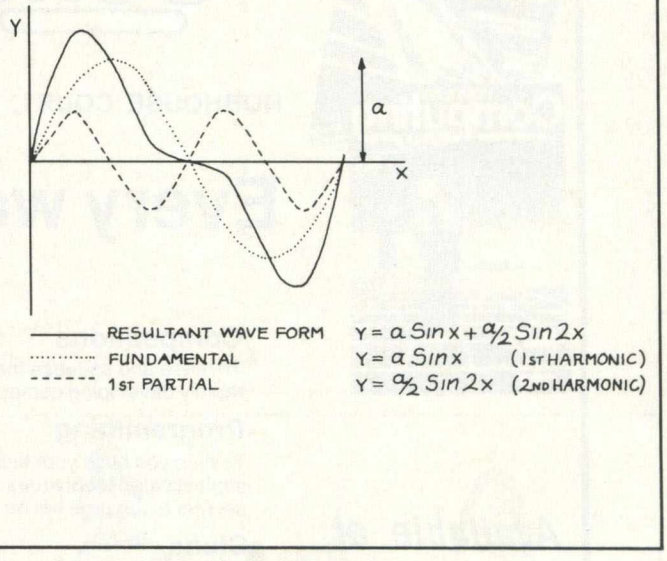
It can be shown that it is possible to build these complex waveforms by addition of simple waves called sine waves so called because they are described by the SINE function in mathematics. The sound produced by a sine wave is a pure tone.

FIGURE 1



An example of the addition of sine waves is given in fig (ii). A typical musical sound contains a rich mixture of harmonics in various proportions and sometimes the fundamental may be virtually absent. In music the frequency of a note is called the pitch.

FIGURE 2

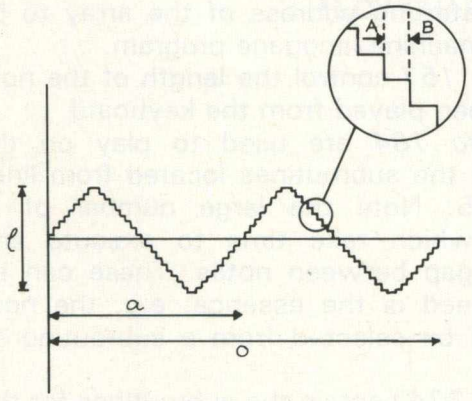


Digital Synthesis

It is possible to produce a waveform by digital means via a D/A convertor. A number at the O/P port produces a voltage proportional to that number but only discreet steps are possible. 256 different voltages can be produced by an 8 bit output.

A waveform produced digitally is shown in Fig (iii). To produce complex waveforms a large number of steps are required. Notice the distinction between the machine output cycle and the audio cycle. The output cycle may contain 1 audio cycle for a low note or several audio cycles for a high note. There must be a whole number of audio cycles in an output cycle since to produce a note of reasonable length the output cycle is repeated several times.

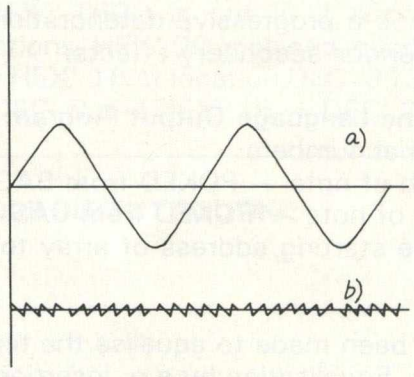
FIGURE 3



a – AUDIO CYCLE
o – OUTPUT CYCLE CONTAINING 256 STEPS
ℓ = 256 POSSIBLE LEVELS

The digitally produced waveform can be regarded as the combination of an audio frequency wave of large amplitude and the stepping frequency wave which is of small amplitude, as shown in fig (iv). It is important to ensure that the stepping frequency is beyond the audio range (greater than 15000 Hz) but since the amplitude is small some compromise is possible. Each output step (AB in fig (iv)) takes a finite time which for the PET is a minimum of about 15 μs and if we choose 256 steps per output cycle (a figure convenient for the internal architecture of the microprocessor) this gives a time per O/P cycle of about 4 ms or a frequency of about 250 Hz. If the output cycle contained 1 audio cycle a note corresponding to middle C on the piano would be produced. Higher frequencies must be achieved by having a greater number of audio-cycles per output cycle but the resolution suffers as there are then fewer steps per audio cycle. In addition these must be multiples of the basic frequency i.e. 500 Hz, 750 Hz etc. Intermediate frequencies may be obtained by varying the length of the output step AB, thus stretching or compressing the waveform (see fig (iv)).

FIGURE 4



a) AUDIO WAVE OF LARGE AMPLITUDE
b) STEPPING WAVE OF SMALL AMPLITUDE, HIGH FREQUENCY AND VARIABLE SHAPE

However in order to "tune the instrument" i.e. produce a certain desired note, the frequency must not only be variable but capable of fine adjustment. This means that a large number of possible times for AB must be available to achieve a number of closely spaced notes. These should ideally be separated by a semitone which is the smallest interval used in Western music, corresponding to the interval between a white and a black note on the piano.

The output step AB must therefore be made considerably longer.

Since the minimum possible change in AB is 2 μs, AB must be at least 100 μs in length giving around 40 possible variations in frequency with a basic frequency (1 audio cycle per output cycle) of around 40Hz. This is about right for a really low note. However, the stepping frequency is then down to about 10 k Hz which is in the upper audio range but is accepted since it is of very small amplitude.

When a machine language output routine is designed taking these considerations into account it proves possible to produce nearly two octaves of a well-tuned major scale. Readjustment of the pitch values can produce a good minor scale but it is impossible to produce a full range of semitones while retaining other desirable qualities. The lowest notes (having the longest step time AB) suffer a little from high

frequency interference from the stepping previously mentioned but it is of low amplitude and in practise not noticeable some distance from the speaker. With the inevitable lengthening of AB previously described, for middle range notes, 4 audio cycles must be included in one output cycle. This cuts the number of steps per audio cycle to 64 but is still sufficient for quite complex waveforms although once more than 4 harmonics are added there is a progressive deterioration in quality due to various secondary effects.

Notes on Machine Language Output Program
HEX' Hexadecimal numbers
"L" gives length of note — POKED from BASIC
"P" gives pitch of note — POKED from BASIC
"A" & "B" give starting address of array to be used

No attempt has been made to equalise the times for output steps. Equalisation by e.g. insertion of extra No Ops, would inevitably slow up the output cycle which would necessitate a reduction in the resolution of the waveform. This lack of equalisation is a cause of background noise on the output but in practise this is hardly noticeable.

The PET uses a 60 Hz interrupt routine which suspends program execution every 1/60 of a second while internal "housekeeping" such as scanning the keyboard takes place. This must be disabled during output or a strong 60 Hz tone is produced. Disabling the interrupt results in a loss of control by the operator while a note is being produced i.e. note length must be preset and cannot be controlled by holding down a key.

Notes on the BASIC program
Lines 5 to 24 load the machine language program into the second cassette buffer which is untouched by BASIC.
Line 25 contains variables for the starting address of arrays. The first array occupies the first cassette buffer and part of the second. This can be used even if all the normally available RAM is taken by a BASIC music program.
Lines 26 to 40 load the arrays which control the quality or timbre of the note.
Line 30 contains a pure tone (1st harmonic or fundamental).
Line 31 combines 1st and 2nd harmonics.
Line 32 combines 1st, 2nd, and 3rd harmonics.
Line 33 produces very low notes.
Line 34 gives an oboe-type sound.
Line 35 contains a simple harmonious chord (containing a third and a fifth).

The numbers in front of the SIN functions must not add up to more than 127. These numbers represent the amplitudes of the wave components. The number 128 at the end represents the datum level around which the output oscillates from 0 to 255.
Line 45 sets the user port to the output state and places the starting address of the machine language program in RAM locations 1 and 2 for use by the A = USR (0) routine.
Line 48 and 49 contain the variables representing the pitch of the notes (doh, ray, me etc.).
Line 50 clears the screen and GETS a character. If desired, instructions for the user could be printed on the screen at this point.
Lines 51 to 56 control the timbre of the note by placing the starting address of the array to be used in the machine language program.
Lines 154 to 157 control the length of the note produced when played from the keyboard.
Lines 160 to 184 are used to play on the keyboard via the subroutines located from lines 200 to 225. Note the large number of IF statements which take time to execute and introduce a gap between notes. These can be pruned if speed is the essence; e.g. the note quality could be selected from a subroutine as required.
Lines 200 to 224 contain the subroutines for the notes produced. Q controls the length of the note and the numbers can be replaced by variables if extra speed is needed. P and L are the addresses in memory of the numbers controlling the pitch and length of the notes. Any value from 1 to 256 can be placed at L but only the range 152-191 is allowed for P.

Once the timbre of the note has been selected, if the RUN/STOP key is pressed followed by GOTO 1000, PET will play "Auld Lang Syne" and will "sing" the words on the screen. (Note acquaintance has a c before the q unlike the printed program).

GOTO 2000 produces the "Sailor's Hornpipe" with a range of speeds available from slow to faster than the "Last Night of the Proms".

The "Hornpipe" is a much more economical program than "Auld Lang Syne" making use of the same subroutines as the "keyboard play" part of the program. This represents a later stage of program development. Note Q is only changed when the note length changes.

The program although perfectly useable as it stands is still only in skeleton form and considerable development is possible.

After typing in the program you are strongly advised to save it on cassette before attempting to RUN. With the use of machine language,

program errors often result in a loss of machine control with the consequent necessity for retyping. If the cassette is used after running the program must be RUN again before use as it makes use of the cassette buffers.

To calculate the starting address of arrays

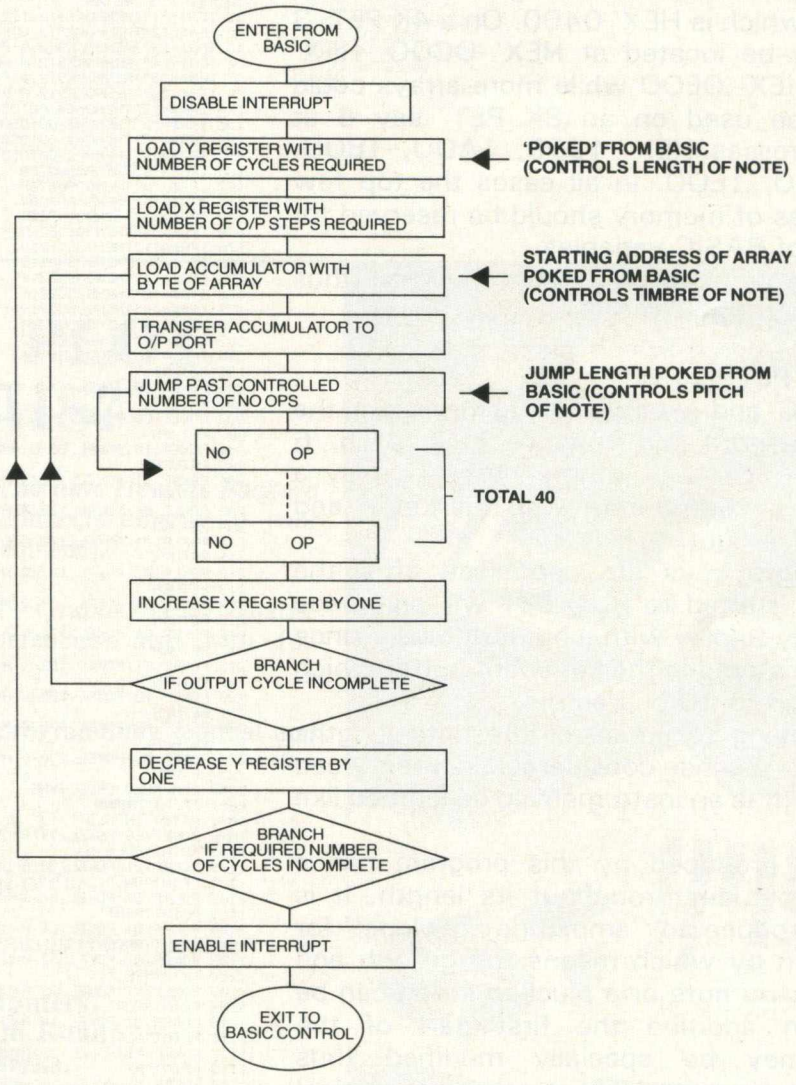
In order to decide the starting addresses of arrays as in line 25 and to POKE the starting address of a particular array as in lines 51 and 56 it is necessary to have some knowledge of the way instructions are stored in memory.

Each memory location contains an 8-bit binary code which is conveniently represented in Hexadecimal notation. HEX numbers use 16 symbols i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, for the numbers decimal 0 to 15. Decimal 16 is then represented as HEX' 10, decimal 17 as HEX' 11 etc. Decimal 256 is HEX' 100 and decimal 4096 is HEX' 1000.

As an example take the array whose starting location in 5632 in line 25. This is 1600 in HEX.

HEX' 1600 is put in 2 successive memory locations, HEX' 00 is placed at location DEC' 912 and HEX' 16 at location DEC' 913. Now HEX' 00 = DEC' 0 and HEX' 16 = DEC' 22 so to load the

FLOWCHART FOR MACHINE CODE OUTPUT ROUTINE



starting address of the array, POKE 912,0 : POKE 913,22.

One array consists of 256 (HEX' 100) Bytes and therefore the next array can be located at HEX' 1700 which is DEC' 5888. To load this address POKE 912,00 : POKE 913,23.

When choosing the starting addresses for arrays other than the one in the cassette buffers, the memory limitations of the machine must be considered.

Total Memory	*Memory limit decimal*	*Memory limit HEX*
4K	4096	1000
8K	8192	2000
16K	16384	4000
32K	32768	8000

On all machines the first 1K is taken by the operating system and Basic Programs start after DEC' 1024 which is HEX' 0400. On a 4K PET, 3 arrays might be located at HEX' OCOO, HEX' ODOO and HEX' OEOO while more arrays could reasonably be used on an 8K PET, say 6 at starting addresses HEX' 1900, 1AOO, 1BOO, 1COO, 1DOO, 1EOO. In all cases the top few hundred bytes of memory should be reserved for the storage of BASIC variables.

Note each array takes $\frac{1}{4}$K of memory which thus cannot be used for a BASIC program.

Playing your PET

Keys Z, X, C and V are used to preselect the length of the note and keys 1, 2, 3, 4, 5, 6 preselect the ''timbre'' of the note. To play a scale on the keyboard start with the key R and progress to the right.

If some note keys are depressed after the program has started to RUN PET will announce that it is ready to play with a burst of music since the notes are stored in the keyboard buffer which can contain up to 10 characters.

Despite having to preselect the note lengths required, with practise considerable dexterity can be achieved. (It is an instrument to be learned like any other).

Each note produced by this program has a constant amplitude throughout its length. It is possible to produce an ''amplitude envelope'' for the waveform by which means the growth and decay of a piano note or a plucked string can be simulated. In addition the first part of the waveform may be specially modified thus providing the characteristic attack of a musical instrument. Virtually any sound can be synthesised from bubbling water to birdsong or space-age effects to simple speech sounds.

SOUND SYNTHESIS ON A PET MICROCOMPUTER
BY DAVID G. BROWN JAN 1980
36 PARKHEAD CRESCENT SHEFFIELD S119RD

```
READY.
 5 DIMV(62)
 10 DATA   120,160,64,162,0,189,122,2,141,79,232,76,176,3,234
 11 DATA234,234,234,234,234,234,234,234
 12 DATA234,234,234,234,234,234,234,234
 13 DATA234,234,234,234,234,234,234,234
 14 DATA234,234,234,234,234,234,234,234
 15 DATA234,234,234,234,234,234,234
 17 DATA232,208,204,136,208,199,88,96
 20 FORN=1TO62
 22 READV(N)
 23 POKE905+N,V(N)
 24 NEXT
 25 AA=634:BB=5632:CC=5888:DD=6144:EE=6400:FF=6656:GG=6912:HH=7168:JJ=7424
 26 FORX=0TO2*πSTEP2*π/256
 30 POKEAA+A,INT(127*SIN(4*X))+128
 31 POKEBB+A,INT(64*SIN(4*X)+63*SIN(8*X))+128
 32 POKECC+A,INT(42*SIN(4*X)+42*SIN(8*X)+42*SIN(16*X))+128
 33 POKEDD+A,INT(97*SIN(2*X)+20*SIN(4*X)+10*SIN(8*X))+128
 34 POKEEE+A,INT(20*SIN(4*X)+30*SIN(8*X)+30*SIN(12*X)+40*SIN(16*X))+128
 35 POKEFF+A,INT(60*SIN(4*X)+20*SIN(8*X)+20*SIN(10*X)+20*SIN(12*X))+128
 40 A=A+1:NEXT
 45 POKE59459,255:POKE1,138:POKE2,3
 48 M2=186:R2=184:D2=182:T1=181:L1=179:S1=176:F1=173:M1=171:R1=168:D1=164
 49 T0=162:L0=157:S0=152:P=918:L=908:Q=2
 50 PRINT"J":GETA$
 51 IFA$="1"THENPOKE912,122:POKE913,2:GOTO50
 52 IFA$="2"THENPOKE912,0:POKE913,22:GOTO50
 53 IFA$="3"THENPOKE912,0:POKE913,23:GOTO50
 54 IFA$="4"THENPOKE912,0:POKE913,24:GOTO50
 55 IFA$="5"THENPOKE912,0:POKE913,25:GOTO50
 56 IFA$="6"THENPOKE912,0:POKE913,26:GOTO50
 154 IFA$="Z"THENQ=64:GOTO50
 155 IFA$="X"THENQ=8:GOTO50
 156 IFA$="C"THENQ=4:GOTO50
 157 IFA$="V"THENQ=2:GOTO50
 160 IFA$="Q"THENGOSUB200
 162 IFA$="W"THENGOSUB202
 164 IFA$="E"THENGOSUB204
 166 IFA$="R"THENGOSUB206
 168 IFA$="T"THENGOSUB208
 170 IFA$="Y"THENGOSUB210
 172 IFA$="U"THENGOSUB212
 174 IFA$="I"THENGOSUB214
 176 IFA$="O"THENGOSUB216
 178 IFA$="P"THENGOSUB218
 180 IFA$="↑"THENGOSUB220
 182 IFA$="<"THENGOSUB222
 184 IFA$=">"THENGOSUB224
 185 GOTO50
 200 POKEP,S0:POKEL,77/Q:A=USR(0)
 201 RETURN
 202 POKEP,L0:POKEL,85/Q:A=USR(0)
 203 RETURN
 204 POKEP,T0:POKEL,96/Q:A=USR(0)
 205 RETURN
 206 POKEP,D1:POKEL,102/Q:A=USR(0)
 207 RETURN
 208 POKEP,R1:POKEL,115/Q:A=USR(0)
 209 RETURN
 210 POKEP,M1:POKEL,128/Q:A=USR(0)
 211 RETURN
 212 POKEP,F1:POKEL,137/Q:A=USR(0)
 213 RETURN
 214 POKEP,S1:POKEL,154/Q:A=USR(0)
 215 RETURN
 216 POKEP,L1:POKEL,171/Q:A=USR(0)
 217 RETURN
 218 POKEP,T1:POKEL,192/Q:A=USR(0)
 219 RETURN
 220 POKEP,D2:POKEL,205/Q:A=USR(0)
 221 RETURN
 222 POKEP,R2:POKEL,230/Q:A=USR(0)
 223 RETURN
 224 POKEP,M2:POKEL,230/Q:A=USR(0)
 225 RETURN
 1000 REM AULD LANG SYNE
 1001 PRINT"■■■■
 1091 PRINT"SHOULD ";
 1098 POKEP,S0:POKEL,19:A=USR(0)
 1099 PRINT"AULD ";
 1100 POKEP,D1:POKEL,38:A=USR(0)
 1101 PRINT"AQU";
 1102 POKEP,D1:POKEL,13:A=USR(0)
 1103 PRINT"AINT";
 1104 POKEP,D1:POKEL,26:A=USR(0)
 1105 PRINT"ANCE ";
 1106 POKEP,M1:POKEL,32:A=USR(0)
 1107 PRINT"BE ";
 1108 POKEP,R1:POKEL,29:A=USR(0)
 1109 PRINT"FOR";
 1110 POKEP,D1:POKEL,13:A=USR(0)
 1111 PRINT"GOT ";
 1112 POKEP,R1:POKEL,29:A=USR(0)
 1113 PRINT"A";
 1114 POKEP,M1:POKEL,16:A=USR(0)
 1115 PRINT"ND "
 1116 POKEP,R1:POKEL,14:A=USR(0)
 1117 PRINT"NE";
 1118 POKEP,D1:POKEL,38:A=USR(0)
 1119 PRINT"VER ";
 1120 POKEP,D1:POKEL,13:A=USR(0)
 1121 PRINT"BROUGHT ";
```

```
1122 POKEP,M1:POKEL,32:A=USR(0)
1123 PRINT"TO ";
1124 POKEP,S1:POKEL,38:A=USR(0)
1125 PRINT"MIND ";
1126 POKEP,L1:POKEL,128:A=USR(0
1127 PRINT"SHOULD ";
1128 POKEP,D2:POKEL,51:A=USR(0)
1129 PRINT"AULD ";
1130 POKEP,S1:POKEL,58:A=USR(0)
1131 PRINT"■■■■■■■QU";
1132 POKEP,M1:POKEL,16:A=USR(0)
1133 PRINT"AINT";
1134 POKEP,M1:POKEL,32:A=USR(0)
1135 PRINT"ANCE ";
1136 POKEP,D1:POKEL,26:A=USR(0)
1137 PRINT"BE ";
1138 POKEP,R1:POKEL,43:A=USR(0)
1139 PRINT"FOR";
1140 POKEP,D1:POKEL,13:A=USR(0)
1141 PRINT"GOT ";
1142 POKEP,R1:POKEL,29:A=USR(0)
1143 PRINT"FOR ";
1144 POKEP,M1:POKEL,16:A=USR(0)
1145 PRINT"THE ";
1146 POKEP,R1:POKEL,14:A=USR(0)
1147 PRINT"SAKE ";
1148 POKEP,D1:POKEL,38:A=USR(0)
1149 PRINT"OF ";
1150 POKEP,L0:POKEL,11:A=USR(0)
1151 PRINT"■■RULD ";
1152 POKEP,L0:POKEL,21:A=USR(0)
1153 PRINT"LANG ";
1154 POKEP,S0:POKEL,19:A=USR(0)
1155 PRINT"SYNE
1156 POKEP,D1:POKEL,77:A=USR(0)
1160 GOTO1000
2000 REM HORNPIPE
2004 PRINT"□HORNPIPE":PRINT"■■PRESS NUMBER FOR SPEED
2005 S=1
2006 GETS
2007 IFS>1THEN2010
2008 GOTO2006
2010 Q=4*S:GOSUB220
2012 GOSUB218
2014 Q=2*S:GOSUB220
2016 GOSUB206
2018 GOSUB206
2020 Q=4*S:GOSUB214
2022 GOSUB212
2024 GOSUB210
2026 GOSUB214
2028 GOSUB220
2030 GOSUB218
2032 GOSUB220
2034 GOSUB224
2036 GOSUB222
2038 GOSUB220
2040 Q=2*S:GOSUB222
2042 GOSUB208
2044 GOSUB208
2046 Q=4*S:GOSUB208
2048 GOSUB206
2050 GOSUB204
2052 GOSUB208
2054 Q=2*S:GOSUB214
2056 GOSUB214
2058 Q=4*S:GOSUB216
2062 GOSUB218
2064 GOSUB220
2066 GOSUB218
2068 GOSUB216
2070 GOSUB214
2072 GOSUB216
2074 GOSUB214
2076 GOSUB212
2078 GOSUB210
2080 GOSUB212
2082 GOSUB210
2084 GOSUB208
2086 GOSUB206
2088 GOSUB206
2090 GOSUB204
2092 GOSUB202
2094 GOSUB200
2096 GOSUB202
2098 GOSUB206
2100 GOSUB204
2102 GOSUB208
2104 GOSUB206
2106 GOSUB210
2108 GOSUB208
2110 GOSUB212
2112 Q=2*S:GOSUB210
2114 GOSUB206
2116 Q=1*S:GOSUB206
2120 GOTO2000
READY.
```


DO YOU PLOT GRAPHS?

This is the affordable answer. The new *Hewlett Packard 7470A* two pen graph plotter connects directly to *your* Commodore Pet (no interface required).

For full graphic output of Graphs, Pie Charts, Histograms, Overhead Transparencies, also with full Digitising facilities.

Supplied complete with cable, programming manual and sample program listing at:

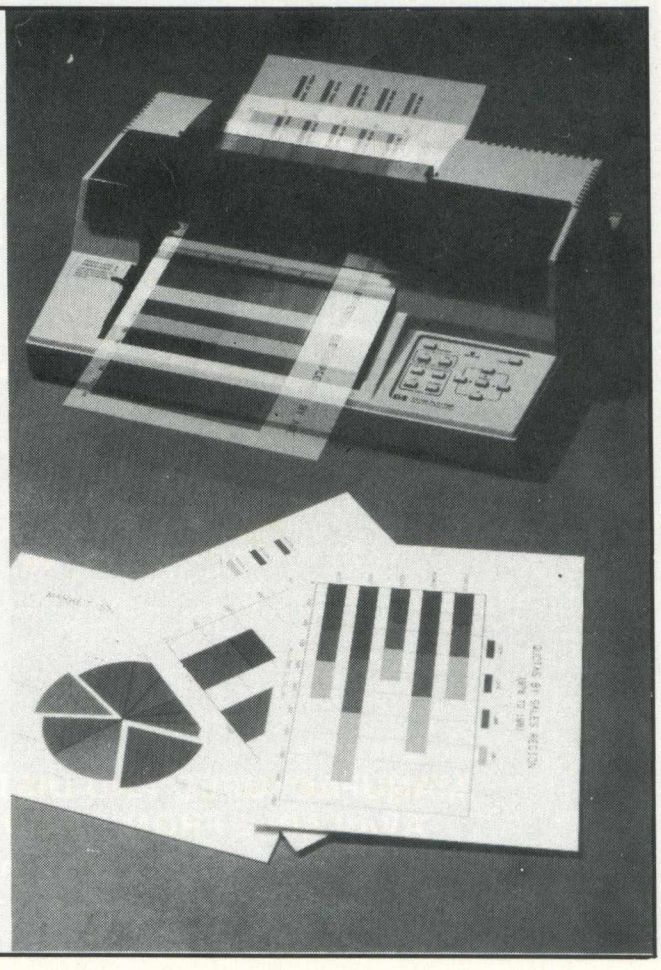
£999 + V.A.T.

10 days approval against official orders.

For further information or a demonstration contact:

C.S.E. (Computers)
12 Wokingham Road
Reading, Berks.
Tel: Reading (0734) 61492

P.S. Also available for Apple.



commodore COMPUTER

PET PACK

BRING TO YOU, NOW ON CASSETTE AND DISK GAMES OF THRILLS & SKILL FOR ALL THE FAMILY

NEW PROGRAMS

Best of Arcade brings together the three most popular Petpack games, Invaders, Cosmic Jailbreak and Cosmiads. These old favourites have been updated to run on 80-column machines also!
MPD 121 BEST OF ARCADE £22.50 - DISK PACK

Best of Treasure Trove gives you twenty of the best games from the Treasure Trove series, including four arcade-type games, Night Drive, Car Race, Breakout and Money Table! There are simulation games, brain-teasers and more, making this package the best value ever in games!
MPD 122 BEST OF TREASURE TROVE £22.50 - DISK PACK

Assembler Tutorial is an extremely well thought out cassette-based package which teaches Assembly Language programming. Now for the first time, you can sit at your computer and learn at your own speed with this self contained course combining lessons with hands-on practice!
MP 124 ASSEMBLER TUTORIAL £50.00

Resident Assembler for all PETs
With excellent documentation and examples.
MP119 RAMP £22.50

Disk Packs available in either 8050 - D8 format or 3040/4040 - D4 format. Please state D8 or D4 when ordering.
Prices include VAT and P&P.

PUB GAMES

This latest disk package brings you five totally new games, never before seen on a PET screen! The programs will all run on 80-column machines also!

DISASTEROIDS - Your mission - pilot your spaceship through the treacherous asteroid belt using your lasers to blast as many asteroids as possible. The PET version of the famous arcade game!

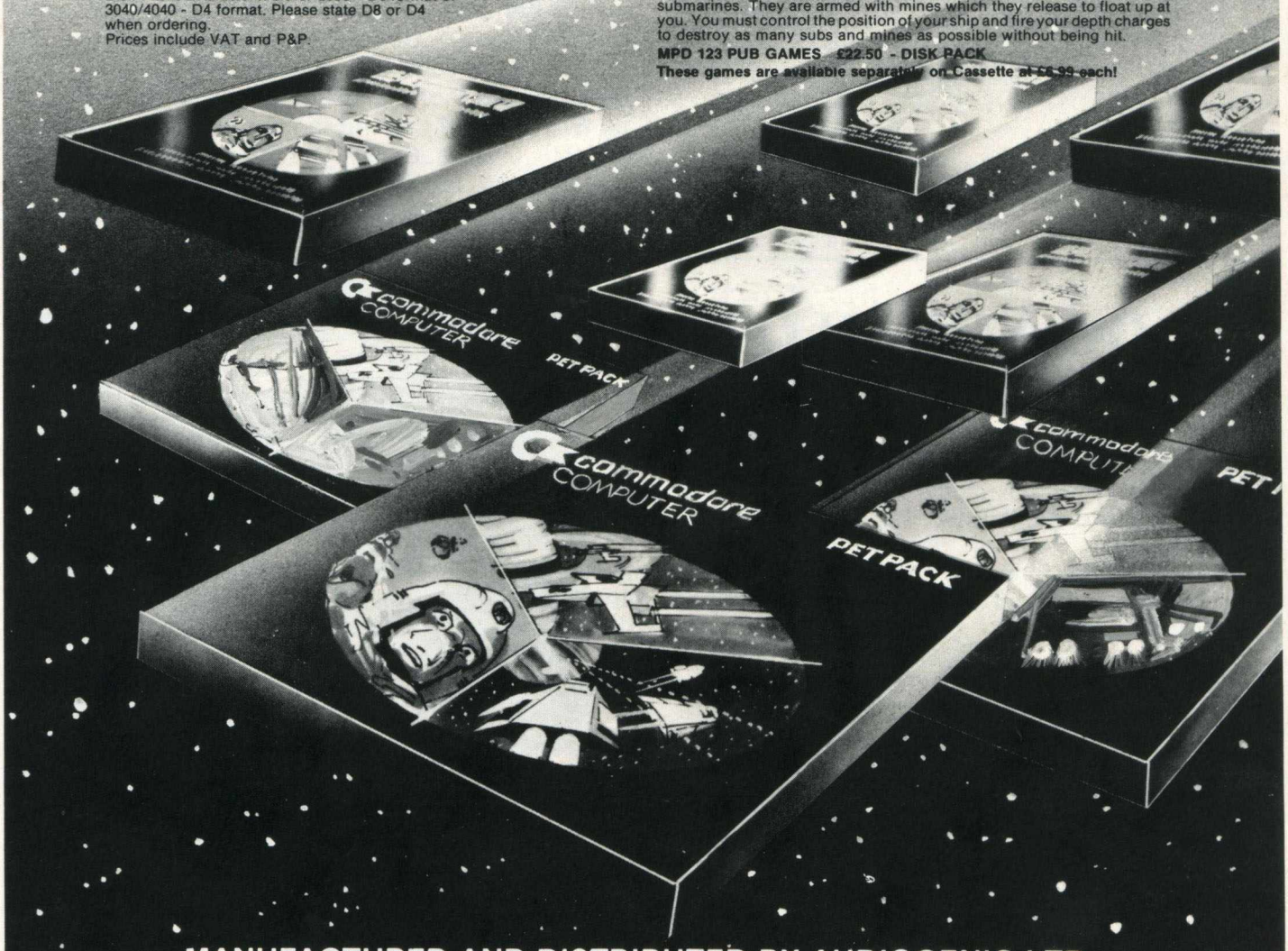
STELLAR WARS - Your spaceship is being pursued by the fighter ships of the evil Empire. You must take control of the ship's laser cannon. Get the fighters in your sights and blast away. The future of the universe depends on your skill and accuracy.

WAR! - You are the captain of the British torpedo boat. You must steer your ship through the minefield to destroy the four shore bases. Unfortunately, you only have two torpedos at a time, so you must get through the minefield again to rearm. Prove that we still rule the waves!

STAR FIGHT - The Alien Invaders are coming again! Control your laser cannon to blast their ships out of the sky and destroy their missiles and bombs. Accuracy and quick reflexes are essential.

DEPTH CHARGE - Your mission - seek out and destroy the enemy submarines. They are armed with mines which they release to float up at you. You must control the position of your ship and fire your depth charges to destroy as many subs and mines as possible without being hit.

MPD 123 PUB GAMES £22.50 - DISK PACK
These games are available separately on Cassette at £6.99 each!



MANUFACTURED AND DISTRIBUTED BY AUDIOGENIC LTD.
AVAILABLE FROM GOOD DEALERS, OR DIRECT FROM:
AUDIOGENIC, P.O. BOX 88, READING, BERKS.
Tel: (0734) 586334

Basic Programs

More Input and Output

This is a follow-up of a previous article published in Volume 3 of CPUCN in two parts, issues Nos. 10 & 11.

Beginners are advised to read these but this articles does contain a brief introduction. It does not deal with the ''internal architecture of the PET and is written for those practical users who wish their micro to communicate with the ''outside world''. It deals mainly with the ''User Port'' but starts with some references to Input/Output in general.

What is meant by Input/Output? The keyboard provides Input only and the screen mainly Output, although a light pen can be said to provide screen input. A list of the more usual types of I/O is given in Table 1. All except the last item, interfaces, will

TABLE 1.

Types of input/Output

```
Cassette units ( via. cassette ports  1 & 2 )
Printers       ( via. the IEEE port usually )
Disk Units     ( via. the IEEE port )
Sound " boxes " .... output only
Light pens  ........ input only
Joysticks   ........ input only
Video output ....... output only
Interfaces  ........ input or output , any port or ports
```

probably be self-contained plug-compatible units which will operate via more or less simple commands, and should in any case be well documented. Some interfaces too will come into this category but in many cases the end use of an interface will be highly specific to the user and he will be expected to write or purchase the necessary software.

Returning briefly to the keyboard this is an input device ''par excellence'' and its functions are quite amazing. It is not just like a typewriter keyboard, as might seem to be the case at first sight, and a few moments reflection on what can be done via the keyboard and how the PET keeps track of keypresses shows that it is a very sophisticated device indeed. Try PRINT PEEK (158) after you have pressed a number of keys (not more than nine) and you will find it knows! Input by keyboard is of course entirely manual . . . or is it? I have made an electromechanical keypresser and such devices are available commercially.

The management of I/O to cassettes, disks, printers, video screens and the like is a fascinating subject but I will only refer to one minor aspect of cassette operation in order to illustrate a very simple form of output.

PROGRAM No. 1.

Cassette motor control.

```
10 PRINT"□PRESS PLAY ON CASSETTE # 1"
20 FORI=1TO2000:NEXT
30 PRINT"□": INPUT"SECS. MOTOR OFF";S
35 INPUT"SECS.MOTOR ON";T
40 POKE59411,60
50 FORI=1TO1000*S:NEXT
60 POKE59411,53
70 FORI=1TO1000*T:NEXT
80 GOTO40
```

Program 1 demonstrates how you can turn the cassette motor on and off at will and is the basis of the various tape positioning programs which enable you to locate a specific place on tape using for example the fast forward control. If however you have an audio cassette unit modified for use with your micro then it will probably have sound output. You can then intersperse screen text or displays with pre-recorded sounds, music or speech in near perfect synchronism.

Since the cassette motor is driven by a 6 volt 250 ma. supply it is also possible to operate any other device controllable by such a voltage and current. You can also switch the supply on with POKE 59411,16 and off with POKE59411,1 but these suggestions are not offered with any guarantees, the risks are yours! Nevertheless I have used these procedures without any undue side effects, as yet.

There is obviously a conflict with normal cassette use and the possible applications are clearly limited, so we come to the User Port and the possibility of eight lines programmable either as imputs or outputs.

The signals which appear on the various lines of the port are TTL ones i.e. nominally 0 or 5 volts, and can supply only small amounts of current. The CBM User Port Cookbook says a maximum of 250 milliamps and this presumably means only a few milliamps per line. The consequences of taking more could be expensive. To avoid such hazards I purchased a Communikit (Mektronic Consultants of Manchester) as described in my previous article. This plugs into the User Port and also needs a supply of between 8 and 24 volts. This supply can be switched on and off via any of the eight channels.

User Port Applications

The first example assumes that you are familiar with the instructions for turning the signals on the

port on and off using POKE 59459,x (sets the Data Direction Register A or DDRA to binary value "x") and POKE 59471,y to put "data" on the port. The "data" i.e. whether the signals are 0 or 5 volts is read by PEEK (59471), which reads the contents of the DATA REGISTER A or DRA.

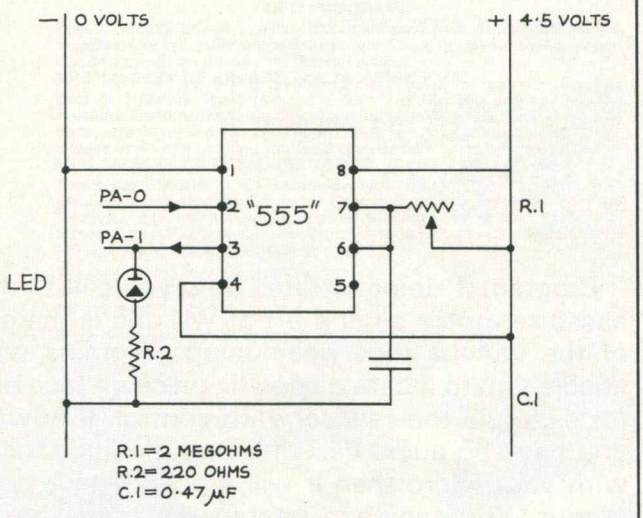


FIGURE No. 1
TIMER CIRCUIT

```
10 REM:::TIMER PROGRAM:::
20 N=59459:X=255:Y=59471
30 POKEN,255:POKEN,X-1
40 FORI=1TO1000:NEXT
50 POKEN,X:FORI=1TO500:NEXT
60 B=TI
70 IFPEEK(Y)>1THEN70
80 C=TI
90 PRINT"TIME ELAPSED = ";C-B
100 GOTO20
```

This is described in greater detail in my previous article so if you find this confusing (I find it confusing!!!) . . . I also described the use of a "555" times circuit see Figure 1. This measures the time constant of the resistance/capacitor network R1/C1, and I suggested a number of possible applications. Since then I have constructed an automatic photo-sedimentometer curve tracer . . . EH? To explain — one way of studying the properties, such as the particle size, of fine powders is to suspend them in water and allow them to settle. Naturally the larger particles settle first and the finer ones will settle slowly. Each particle can be considered to approximate in shape to a sphere and thus to settle at varying speeds according to Stoke's Law

$$v = \frac{r^2(D-d)g}{4.5n}$$

where v is the velocity, r the radius of the particle, D its density and d , n the density and viscosity of the liquid it is suspended in! I leave this as an "exercise to the student" to work out what the settling velocity of the fine clay particles should have been in the following experiment. The sedimentometer consisted of a transparent plastic bottle (a shampoo bottle) containing my suspension of fine clay in water. I used a 6 volt 50 ma. pygmy lamp as the light source and shone it through the suspension onto an ORP-12 to detect the amount of light passing. The lamp and photo-detector were simply taped on to the bottle with black PVC tape as shown in Figure 2.

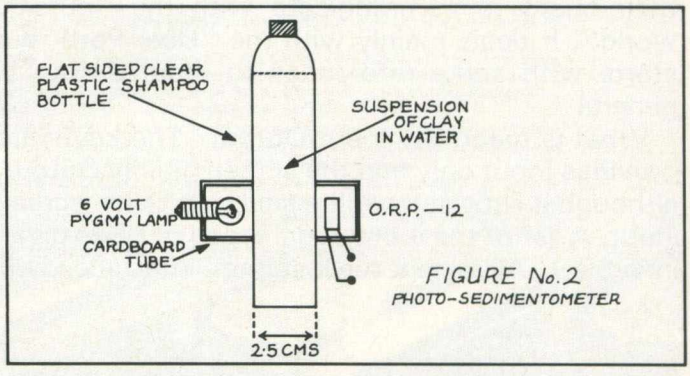


FIGURE No. 2
PHOTO-SEDIMENTOMETER

In this application I used a 25 mfd. capacitor and a 15 k resistor in series with the ORP-12. The program used is given as Program No. 2. This demonstrates the use of an ARROW chip for plotting purposes, hence the strange (?) symbol θG and the odd Z = 1! It also explains the SYS 40960 at the beginning. You can use instead any of the many plotting techniques available such as DEFN(Z) = 32767-40-Y-X and then plotting with POKE Z, 65 or whatever, or make use of the routine at address 59479 e.g. POKE 148, X : POKE 216, Y : SYS 59479 followed by PRINT"." or any symbol you prefer.

The same technique could be used to make an

PROGRAM No. 2.

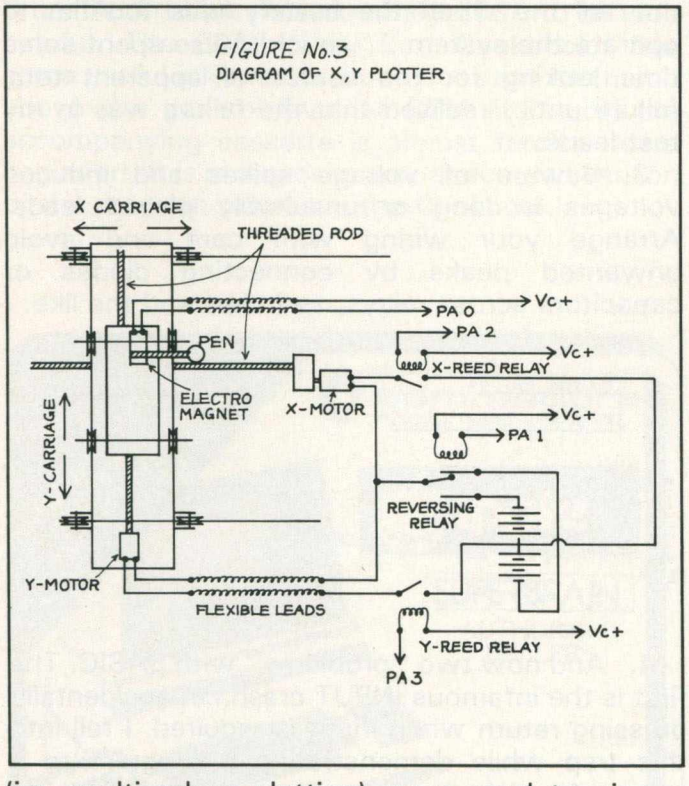
Photosedimentometer curve plotter

```
5 SYS40960
10 PRINT"□ PLOT OF LIGHT INTENSITY VS TIME
11 PRINT"──────────────────────────────────"
15 PRINT"↑ TRANSMISSION
20 FORI=1TO20
25 PRINT"I":NEXT
30 PRINT"──────────TIME──────────────────"
100 L=TI
110 N=59459:X=255:Y=59471:J=J+1
120 POKEN,X:FORI=1TO10:NEXT
130 POKEN,X-1
150 POKEN,X
170 B=TI
180 IFPEEK(Y)>1THEN180
190 C=TI
200 X=(TI-L)/30:Y=C-B
210 Z=1:←G
220 IFJ=80THEN250
240 GOTO110
250 WAIT59410,4,4
260 RUN
READY.
```

automatic titration apparatus and a chemist could I am sure think up many more uses.

My next project was a little too ambitious for my engineering skills, even using Meccano in the best traditions of that early designer of computers Herman Zuse! I constructed an X-Y plotter. This used two low voltage electric motors driving lengths of screwed rod in captive nuts on the X and Y carriages. These motors cost only about 30 to 40 p and the rest of the bits and pieces came from the scrap drawer so this must be one of the cheapest X-Y plotters yet produced. I would not describe it as fast or, in any way I built it, very accurate but it did work and if I can do it almost everyone else could do it better! A diagram is given in Figure 3, and it will be seen that I used pen-lift since the power available was small and the rigidity of the construction poor. Fibre tip pens

FIGURE No.3
DIAGRAM OF X,Y PLOTTER



(i.e. multi-colour plotting) were used to give a series of dots which enabled high definition to be obtained if you could wait long enough. The program I used is Program No. 3.

Analogue/Digital Conversion

You will see that a great variety of input can be carried out fairly simply using switched input methods i.e. the simplest form of digital input. The Communikit can also accept 8-bit parallel input using the control lines CA-1 and CB-2 but this is somewhat outside the scope of this article. However since the outside world deals more frequently in analogue type variables such as voltage, current, temperature, pressure, time and so on i.e. constantly varying parameters it is of interest to examine how these can be handled by a computer. In other words the somewhat daunting subject of data handling.

PROGRAM No. 3 .

X - Y Plotter.

```
100 POKE59459,255:R=59471:POKER,0
110 PRINT:PRINT"□TEST PROGRAM"
120 PRINT:PRINT"
130 PRINT:PRINT"PRESS P TO OPERATE PEN LIFT"
140 PRINT:PRINT"       R TO OPERATE REVERSE"
150 PRINT:PRINT"       X TO RUN X-MOTOR"
160 PRINT:PRINT"       Y TO RUN Y-MOTOR"
165 PRINT:PRINT"TO REVERSE MOTORS PRESS T(FOR XREV.),Z(XREV.)"
170 GETA$:IFA$=""THEN170
180 IFA$="P"THEN500
190 IFA$="R"THEN600
200 IFA$="X"THEN700
210 IFA$="Y"THEN800
220 IFA$="T"THEN900
230 IFA$="Z"THEN1000
500 POKER,1
510 FORI=1TO500:NEXT
520 POKER,0:GOTO170
600 POKER,2
610 FORI=1TO500:NEXTI
620 GOTO170
700 POKER,4
710 FORI=1TO500:NEXT:POKER,0
720 GOTO170
800 POKER,8
810 FORI=1TO200:NEXT:POKER,0
820 GOTO170
900 POKER,10:FORI=1TO200:NEXT
905 POKER,0
910 GOTO170
1000 POKER,6:FORI=1TO500:NEXT
1005 POKER,0
1010 GOTO170
1015 POKER,0
READY.
```

Small amounts of data are readily handled via keyboard entry, more data can be stored as data statements but one finds that such subjects as mailing lists soon exhaust even 32K of memory and one resorts to data files stored on tape and then on disk and then . . .

But analogue data is available in almost infinite amounts so one has to resort to sampling techniques and statistical procedures. There are however relatively simple applications such as finding the position of a potiometer spindle or a joystick control lever.

In fact this is exactly what the timer circuit of Figure 1 does. Two snags which limit the usefulness of this device are (a) the speed of the basic program and (b) the fact that a voltage cannot be input and measured easily. The first snag can be overcome by using a machine code program which enables a measurement to be carried in microseconds rather than milliseconds. A suitable program has been published in the ''PET REVEALED'' providing you are aware of the necessary change (FO 07 to FO F7). A slightly simplified version with the TIM monitor format and BASIC program to accompany it is Program 4. A simple plotting program now gives you realistic screen movements and I was surprised to

find that I could detect mains hum picked up by rather long input leads and thus quite unwittingly plot sine curves! This was all the more surprising since when I first examined the data being fed in it appeared to be varying rapidly and wildly with apparent randomness and it was only when plotted that its true nature was obvious.

PROGRAM No. 4 .

Fast coun ing.

```
B*
     PC  IRQ  SR AC XR YR SP
.;  0005 E62E 30 00 5E 04 F8
.
.   033A 78 A9 00 8D E0 03 A9 FF
.   0342 8D 43 E8 EE E0 03 AD 4F
.   034A E8 C9 02 F0 F6 58 60 00
.?
```

MACHINE CODE & MNEMONICS

```
78        SEI
A9 00     LDA   WITH  0  (DECIMAL)
8D ED 03  STA   @     03E0
A9 FF     LDA   WITH  255(DECIMAL)
8D 43 E8  STA   @     E843(59459)
EE E0 03  INC   03E0
AD 4F E8  LDA   FROM E84F(59471)
C9 02     CMP
F0 F6     BNE
58        CLI
60        RTS
```

These programs may be used in conjunction with a simple BASIC program as shewn below.

```
10 POKE59459,254
20 SYS826
30 PRINTPEEK(59471)
40 GOTO10
```

The second snag is not so easily overcome unless you buy a purpose built A/D converter so I tried a simpler solution, a 75p chip the 4151. The circuit shown in Figure 4 will convert a voltage of 0-10 to a frequency of 0-10 kHz and is obviously only a little more complex than the ''555'' timer circuit. If connected to the input of a timer circuit which has an R/C or frequency constant of greater than 10 kHz the frequency and thus the voltage input may be measured. (A.001 uF capacitor and a 10k resistance is suitable).

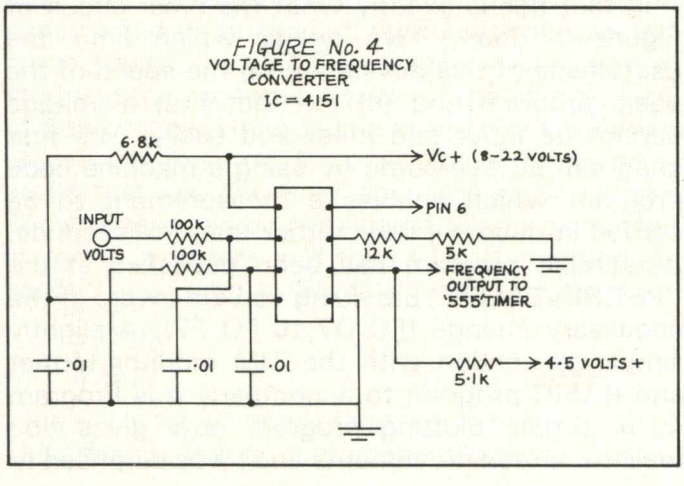


FIGURE No. 4
VOLTAGE TO FREQUENCY CONVERTER
IC = 4151

Problems

If you are thinking of starting up in this field these comments may be of use. Some may seem obvious or elementary but if you are concentrating on one aspect then a trivial problem can seem to be very baffling, for a time at least, and I find myself thinking up all sorts of explanations other than the obvious.

1. If you have peripherals connected to your PET make sure that they are switched on. Strange effects are produced by peripherals which are plugged in but not powered. I had this experience first with a printer and even called in an engineer who was most puzzled. His diagnostic tests indicated a faulty chip and he spent two fruitless hours changing chips without success.

2. I wired in an LED to show that I had my battery supply connected . . . fine . . . but it did not tell me when the battery was too flat to operate the system . . . ouch! I also spent some time looking for the source of apparent total failure until I realised that the failure was in my test leads.

3. Beware of voltage spikes and induced voltages in long or unsuitably placed leads. Arrange your wiring with care and avoid unwanted peaks by connecting diodes or capacitors across relays, switches and the like.

PROGRAM No. 5.

IF A = B THEN .....

```
 5 REM...IS A = B ?
 10 A=.01:B=1/100
 20 PRINTA,B
 30 IFA=BTHENPRINT"YES,A=B":STOP
 40 PRINT"A-B = ";A-B
READY.
```

4. And now two ''problems'' with BASIC. The first is the infamous INPUT crash i.e. accidentally pressing return when input is required. I fell into this trap while demonstrating a program to a group of people (it always happens this way) and then somewhat confused typed a number and again pressed return. Yesss you guessed right I had of course deleted a program line . . . and an essential one at that. POKE 14,1 before INPUT solves this problem.

The other problem which I find even highly experienced programmers suffer from is the IF A = B variety! This is demonstrated in Program No. 5. No prizes are offered for the explanation but one way to avoid the problem is to use greater or less than statements or IF ABS (A-B) .00001 THEN . . .

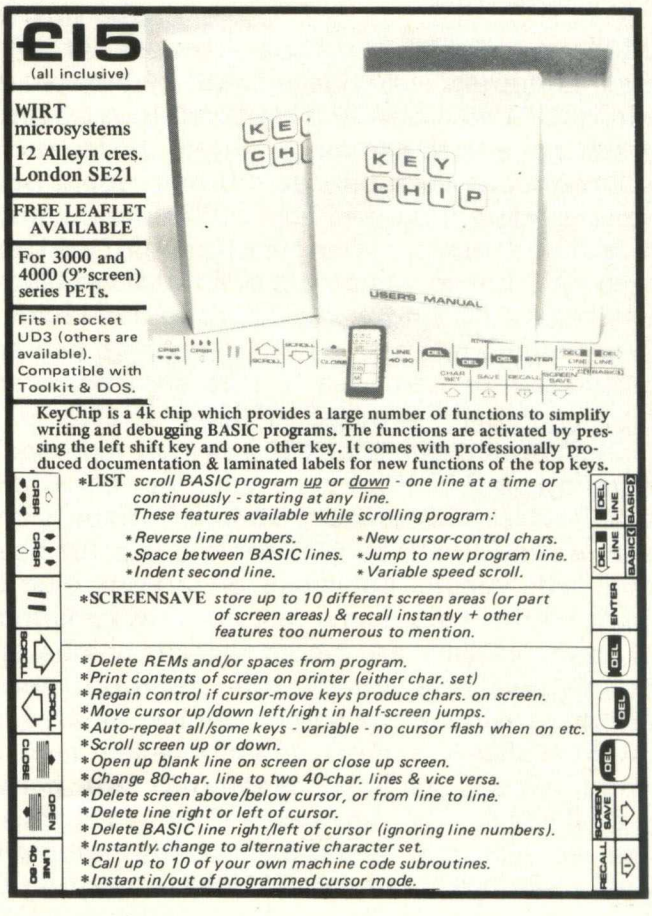
5. Finally I had some difficulty to start with in

getting all the appropriate bits and pieces together and in finding really good books on the subject. Some radio spares shops are remarkably unhelpful if you ask for items for computer use, some computer shops are also unhelpful unless you wish to buy an x-thousand pounds system! Tandy shops are useful suppliers of odds and ends but they too seem to ''act funny'' if you mention a PET! The pages of the computer journals contains more useful information as to suppliers of odds and ends but you should find out if you have a ''computer breaker'' in your neighbourhood . . . an invaluable source of cheap goodies.

As to books your County Library will doubtless have a phenomenal stock of books on computers, mainly unsuitable for our purposes. I have however found ''Micro Computer Interfacing'' by Bruce A. Artwick (Prentice Hall) 1980 to be very helpful. The CBM ''User Port Cookbook'' has some useful information but it does not rise much above the general level of CBM literature, the accompanying cassette is almost farcical for a beginner! (It is in any case only a recorded version of the program printed in the Cookbook).

By contrast the manual accompanying the Communikit is an excellent example of how to produce a manual and I only wish that they would prepare a more comprehensive manual in addition to their listing one.

Of course if you don't wish to bother with all this detail you can buy a ready built unit complete with A/D converters, 16-bit accuracy (.0015%) bi-directional control, relay switching, IEEE and RS 232 interfaces built in, variable baud rates, voltage/current signal shaping, plug in user oriented boards *and* LED or digital displays!



Programs For Your VIC 20

Programs for the Vic

Two games programs for the Vic, both utilising programmable high resolution graphics, colour and sound, and containing all instructions for playing the game within the program. These are written for the unexpanded Vic, so remove those RAM packs before going into action.

```
        SPACE PIRATES - VIC 20


 1 POKE36878,15
 2 I2=PEEK(829):I9=PEEK(826)
 5 IFI9<>0THEN14
 10 VI=36864:POKEVI+5,240:POKEVI+15,219:PRINT"■SPACE BUCCANEERS":PRINT"──────
──────"
 12 T$="■A - BASE 1 FIRE":GOSUB9000:T$="■S - BASE 2 FIRE":GOSUB9000:T$="■D -
BASE 3"
 13 T$=T$+" FIRE":GOSUB9000
 14 POKE56,PEEK(56)-2:POKE52,PEEK(56)-2:POKE51,PEEK(55):CLR
 15 DT=32768:VI=36864:CC=256*PEEK(52)+PEEK(51):FORI=0TO511:POKECC+I,PEEK(DT+I):N
EXT
 16 FORI=464TO496:READJ:POKECC+I,J:NEXT:POKEVI+5,255
 17 DATA129,66,36,0,0,36,66,129,24,24,24,24,60,126,126,255
 18 DATA64,32,31,25,57,127,238,56,2,4,248,152,156,254,119,28
 19 I2=PEEK(829):I9=PEEK(826):S=PEEK(827)*25:IFI9<0THENS=S+(PEEK(826)-1)*6300
 20 S2=5:GOSUB500:V=25:T$="MOTHERSHIP-(=-?MYSTERY":T2$="#1 PIRATE-@-25"
 21 IFI9=0THENI2=50
 22 POKEVI+15,8:PRINT"■■SCORE"S:PRINT"----------":POKE8105,59
 23 PRINT:PRINT:FORI=1TO5:PRINT"■■          ■■  ┌ ┌ ┌ ┌■":PRINT:NEXT:POKE8099,59:PO
KE8102,59
 24 Z(1)=7790:Z(2)=7834:Z(3)=7878:Z(4)=7922:Z(5)=7966
 25 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■PIRATE BOUNTY■":FORT=1TO5:XZ(T)=1.25+RND(TI):N
EXT
 26 FORK=1TO5:IFI3(K)=1THENPOKEZ(K),32:GOTO43
 27 X=RND(K):IFS1=5THENGOSUB500
 28 IFI4=0ANDRND(4)<.02THENI4=1:BU=7724
 29 IFI4=1THENPOKEBU,32:BU=BU+1:POKEBU+1,61:POKEBU,60
 30 IFI4=1ANDBU>=7743THENI4=0:POKEBU,32:POKEBU+1,32
 31 IFX>.5THENPOKEZ(K),32:Z(K)=Z(K)+XZ(K):POKEZ(K),0:POKEZ(K)+30720,5
 33 GOSUB200
 34 IFPEEK(Z(K)+1)=79THENI2=I2-1:PRINT"■■■■■■■■■■■■■CELLS "I2"■ "
 35 IFI2<=0THEN9005
 36 IFZ(K)>=8098THEN9005
 43 NEXT:GET#:IFA#="0"THEN99
 45 GOTO26
 100 POKE36869,240:POKE56,30:POKE52,30:CLR:PRINT"■■SCANNING SECTOR"
 101 PRINT"■■STARDATE "TI$:RUN
 200 Z=PEEK(197)
 202 IFZ=17THENC=8077:GOSUB220
 204 IFZ=41THENC=8080:GOSUB220
 206 IFZ=18THENC=8083:GOSUB220
 210 RETURN
 220 MM=32:FORI=C-22TOC-330STEP-22:POKEI+22,MM:IFPEEK(I-22)=0THEN300
 221 MM=PEEK(I-22):IFMM=61ORMM=60THENGOSUB350
 222 POKEI,33:IFBU=I-22THENPOKEI,32:I=C-330:GOTO224
 223 FORDE=253TO250STEP-1:POKE36875,DE:NEXT:POKE36875,0
 224 NEXT:POKEI+22,32:RETURN
 300 POKE36875,0:POKEI,32:POKEI-22,58:POKEI+30698,2
 301 FORDE=200TO255:POKE36877,DE:NEXT:POKE36877,0:POKEI-22,32
 302 S=S+V:S1=S1+1:PRINT"■SCORE"S:I=C-352:I3(K)=1:S2=S2+1:IFS>=12000THENS=S-1200
0
 303 IFS1/20=INT(S1/20)ANDS1<>0THEN990
 304 IFS2=5THENGOSUB500:V=V+25:FORJ=1TO5:I3(J)=0:NEXT:GOTO22
 305 GOTO223
 350 V=INT(RND(7)*4+1)*50:POKEBU+1,32:POKEBU,58
 351 FORDE=150TO180:POKE36877,DE:NEXT:POKE36877,0:I4=0:POKEBU,32
 352 S=S+V:PRINT"■SCORE"S:POKEI,32:RETURN
 393 IFS1/20=INT(S1/20)ANDS1<>0THEN990
 500 IFS2<>5THENRETURN
 502 FORI=0TO7
 503 READJ
 505 POKECC+I,J
 506 NEXT:S2=0
 509 DATA102,102,60,231,189,60,102,66,66
 511 DATA24,60,126,90,126,90,129,66,0,60,189,102,189,24,36,24
 512 DATA60,126,159,254,62,34,34,102
 514 RETURN
 990 POKE826,1:POKE829,I2:IFS-6300>0THENPOKE827,(S-6300)/25:POKE826,PEEK(826)+1
GOTO100
 991 POKE827,S/25:GOTO100
 9000 FORI=1TOLEN(T$):PRINTMID$(T$,I,1);:FORDE=1TO70:NEXT:NEXT:PRINT:RETURN
 9005 POKE8102,58:POKE8105,58:POKE8099,58
 9006 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■":PRINT"■■■■■■■■CELLS###"
 9007 PRINT"■■■■■DESTROYED#":PRINT"■■■■■■■■■■■■■■■"
 9010 FORDE=255TO120STEP-1:POKE36877,DE:FORD=1TO30:NEXT:NEXT
 9020 POKE36877,0:PRINT"■ANOTHER GO?(Y/N)"
 9023 POKE36879,76:POKE198,0
 9025 GETA$:IFA$=""THEN9025
 9030 IFA$="N"THENPOKE826,0:POKE827,0:POKE829,0:GOTO9036
 9035 POKE826,0:POKE829,0:POKE827,0:GOTO100
 9036 POKE56,30:PRINT"■■":POKE36869,240:POKE36879,27:CLR:END
READY.
```

```
CONQUEST - VIC 20


 1 REM CONQUEST BY B.J.SLADE
 2 FORI=826TO835:POKEI,255:NEXT
 9 POKE36879,93:POKE36869,242
 11 PRINT"■-ONQUEST":PRINT"──────"
 12 PRINT"■OUR AIM IS"
 13 PRINT"■TO KILL ALL THE"
 14 PRINT"■NASTY KNIGHTS "
 15 PRINT"■WHO ARE HIDING "
 16 PRINT"■AROUND THE CASTLE."
 17 GOSUB900
 18 PRINT"■INPUT DIFFICULTY":PRINT"■■(1-9)"
 19 GETDH#:IFDH#=""THEN19
 20 DF=VAL(DH$):IFDF<1ORDF>9THEN19
 21 DEFFNB(X)=INT(RND(4)*150+DF*10):K=197:Z=PEEK(K)
 22 POKE825,DF
 23 DEFFNA(X)=INT(RND(8)*90+TV)
 24 POKE36869,240:TL=7742:TD=7724:Y1=106:Y2=113:Y3=117:BL=8142
 25 KN=83:AR=30:TV=7758
```

Space Pirates is an Invaders type game, whereby rows of aliens wander across the screen, and you have to shoot them down. You have three laser bases from which to do this.

Conquest has you defending a castle, attempting to shoot down the marauding knights with your catapult before time runs out.

```
 30 PRINT"■■■■■■■■■■■■■■■■■■■
```

Machine Code

System

Some time ago, I developed a program written in both BASIC and machine language. The machine language part of the program formed a large library of assembler-coded subroutines, which could be called from a BASIC program by using the SYS command. The SYS command, although providing a useful interface to machine code, is rather limited if used in its standard form. The main problem arises from the fact that entry-points have to be specified as absolute addresses, as in SYS 826, for example. Consequently, any changes made to subroutines at the assembler level, which alter start-addresses, or entry points, involve updating SYS calls in the main BASIC program. Other languages, like FORTRAN, allow the user to call subroutines by name, including subroutines written in assembler. Being able to call a subroutine by name not only eases the task of program maintenance and development, but it also greatly enhances the 'readability' of the program. In a large program, readability and ease of maintenance become significant design criteria, so therefore I decided to implement a call-by-name mechanism within BASIC. By describing the implementation in detail, I hope that others will derive the same benefits as I did from using the call-by-name approach.

But first, for those of you who are not familiar with the subroutine-calling conventions offered by other programming languages, a few words of explanation. As you probably know, a call to a subroutine in BASIC requires the use of the GOSUB command, and any parameters needed by the subroutine have to be established in variables prior to making the call. So, a typical program segment of BASIC code, which calls a cursor-positioning subroutine, for example, might look like this:—

```
1110 X=5: REM SET X COORDINATE
1120 Y=9: REM SET Y COORDINATE
1130 GOSUB 5000: REM-POSITION CURSOR
```

Now if we were using FORTRAN as our programming language, then the same segment of program could be coded as:—

```
CALL CURSOR(5,9)
```

Comparing the two calling sequencies above, it is obvious that the FORTRAN code is far more 'readable' than the equivalent BASIC version for two reasons. Firstly, by using a meaningful name, even without annotating the code, the function of the subroutine can be easily surmised just by looking at the call. By comparison, if the REM statements were removed from the BASIC version, it would be virtually impossible to determine the function of the GOSUB without actually looking through the subroutine code and analysing the program logic. Secondly, parameter passing is considerably more elegant in the FORTRAN version than in the BASIC equivalent. If we now look at how assembler-coded subroutines are called from BASIC, we see that the call to our cursor-positioning subroutine becomes even less readable, as the following example illustrates:—

```
1110 POKE 23569,X
1120 POKE 23570,Y
1130 SYS 23571
```

Imagine how cryptic a BASIC program would look if there were calls to thirty or more assembler subroutines! Surely it would be better if one could call the cursor-addressng routine by:—

```
1110 SYSTEM,''CURSOR'',5,9
```

Well you can, and in this first article I shall start to tell you how.

For the moment, and for the sake of simplicity, let us study a small assembler-subroutine library consisting of just two routines: The first routine, when called, forces the screen display into lower-case mode, and the second routine performs the complimentary function of forcing upper-case, or graphics, mode. Under normal circumstances, we would need to have a uniquely coded SYS call to invoke each of the library subroutines. But, using the call-by-name scheme, we can dispense with the usual style of SYS call. Instead, we route all SYS calls to what is called a 'dispatcher'. The dispatcher figures out which subroutine has been requested, and passes control to the appropriate entry point. To understand how this works, we need to take a close look at the structure of our sample library. As explained already, the library contains two routines, which are used to change the screen display mode. As well as these two 'user' routines, the library also contains the dispatcher routine, which is the only routine that is ever called directly from a BASIC program. When the dispatcher receives control from a BASIC SYS call, it scans the SYS call line for a six-character, space-filled name, and uses the name to perform a 'look-up' in the dispatcher's entry-

point-table. The entry-point-table contains, in addition to the subroutine names, all the subroutine start addresses. When the dispatcher finds the requested name, it pushes the appropriate start address onto the stack. Then, via an RTS instruction, it gives control to the called subroutine. If you look at listing 1, you will see exactly how the dispatcher, the entry-point table, and the library subroutine are put together. Lines 1 through 36 of the listing contain declarations of the ROM routines, constants, variables, and pointers which are used by the dispatcher and library routines. (As a general tip, you will find that code is much easier to read and maintain if you religiously avoid using constant values when coding instructions). Including commentary, lines 37 through 135 contain the actual dispacter routine. The label 'SYSTEM', at line 58, marks the point where control is received from the BASIC SYS call. Following the dispatcher routine is the entry-point-table (E.P.T.). Label 'EPTAB', at line 155, marks the start of the table. For each library subroutine there is a two-line entry in the E.P.T. The first line uses the assembler .BYTE directive to generate the subroutine name. Note well that the name given between single-quotes, if less than six characters, must be space-filled. The second line of each entry uses the assembler .DBYTE directive to generate the subroutine start-address minus one. Why minus one? Well, as stated earlier, the dispatcher passes control to the library subroutine by pushing an address onto the stack, and then executing an RTS instruction. The RTS instruction computes its target address by popping the stack and adding 1, so entry-point addresses are always generated as one less than the actual start address. As new library subroutines are added, it is quite a simple task to code new E.P.T. entries. The last part of the listing, lines 164-207, contains the actual library, or user, subroutines.

Now that we have studied the structure of the library, let us move on to see how a BASIC program accesses the subroutines. Listing 2 shows a simple BASIC program, that illustrates just how easy it is to make calls to the library. Lines 10-30 show a fairly standard method of booting a machine-coded segment, in this case our library. Line 30 assumes that the library has been assembled, loaded, and saved in a file called 'USER-LIB'. Line 50 is interesting — not only does it assign the dispatcher start-address to the BASIC variable 'TEM', but in so doing, allows the more usual 'SYS 24576 to be cosmetically coded as the pseudo-keyword: SYSTEM.

Well, there you have it, a useful call-by-name facility, which requires little effort to implement, and which provides the twin benefits of improved readability and easier program maintenance.

In my next article, I shall describe how the call-by-name technique can be developed to allow 'SYSTEM' calls to include parameters. The article will also include listings of some very useful assembler subroutines.

```
0002  0000                ;
0003  0000                ; *********************************
0004  0000                ; *                               *
0005  0000                ; *          DECLARE              *
0006  0000                ; *                               *
0007  0000                ; *   BASIC 4.0 ROM ROUTINES      *
0008  0000                ; *                               *
0009  0000                ; *********************************
0010  0000                ;
0011  0000                ERRMSG = $BF00        ; NAME NOT FOUND = 'SYNTAX ERROR'
0012  0000                EVALU8 = $BD98        ; EVALUATES A BASIC EXPRESSION
0013  0000                NXTFLD = $BEF5        ; SCANS PAST A COMMA
0014  0000                STRING = $C7B8        ; SETS UP CHARACTER STRING PNTR
0015  0000                ;
0016  0000                ; *********************************
0017  0000                ; *                               *
0018  0000                ; *          DECLARE              *
0019  0000                ; *                               *
0020  0000                ; *    CONSTANTS, VARIABLES       *
0021  0000                ; *                               *
0022  0000                ; *        AND POINTERS           *
0023  0000                ; *                               *
0024  0000                ; *********************************
0025  0000                ;
0026  0000                CASE   = $E84C        ; CASE-MODE SWITCH-ADDRESS
0027  0000                ENTPNT = $0000        ; SUBROUTINE START ADDRESS PNTR
0028  0000                ENTLEN =  0008        ; ENTRY SIZE
0029  0000                LOWER  =  0014        ; LOWER-CASE MODE VALUE
0030  0000                NAMSIZ =  0006        ; SUBROUTINE NAME SIZE
0031  0000                SUBNAM = $001F        ; NAME ON 'SYSTEM' CALL
0032  0000                TABNAM = $0000        ; POINTER TO CURRENT E.P.T. NAME
0033  0000                UPPER  =  0012        ; UPPER-CASE MODE VALUE
0034  0000                ZERO   = $0000        ; ZERO = 0, SEEMS REASONABLE!
```

```
0035  0000                ;
0036  0000                ;
0037  0000                ;
0038  0000                ; ********************************
0039  0000                ; *                              *
0040  0000                ; *                              *
0041  0000                ; * [] D I S P A T C H E R  []   *
0042  0000                ; *                              *
0043  0000                ; * THIS ROUTINE ENABLES BASIC   *
0044  0000                ; * PROGRAMS TO CALL-BY-NAME     *
0045  0000                ; * MACHINE-CODE SUBROUTINES.    *
0046  0000                ; *                              *
0047  0000                ; *  E.G.                        *
0048  0000                ; *                              *
0049  0000                ; * 10 SYSTEM,"CASELO"           *
0050  0000                ; *                              *
0051  0000                ; * 90 SYSTEM,"CASEUP"           *
0052  0000                ; *                              *
0053  0000                ; ********************************
0054  0000                ;
0056  0000                      *=$6000
0057  6000                ;
0058  6000  D8            SYSTEM CLD                ; WE DON'T WANT DECIMAL MODE
0059  6001  20 F5 BE             JSR NXTFLD         ; SCAN TO START OF SUBRTN NAME
0060  6004  20 98 BD             JSR EVALUS         ; EVALUATE CHARACTER STRING
0061  6007  20 B3 C7             JSR STRING         ; THEN SET UP POINTER IN SUBNAM
0062  600A  A9 12                LDA #EPTLEN        ; GET THE LENGTH OF THE E.P.T
0063  600C  4A                   LSR A              ; DIVIDE E.P.T LENGTH BY EIGHT
0064  600D  4A                   LSR A              ; (NUMBER OF BYTES PER ENTRY)
0065  600E  4A                   LSR A              ; YIELDS NUMBER OF ENTRIES
0066  600F  AA                   TAX                ; ENTRY COUNT IS LOOP CONTROL
0067  6010  A0 00                LDY #ZERO          ; INITIALISE CHARACTER INDEX
0068  6012  A9 42                LDA #<EPTAB        ; SET UP E.P.T NAME POINTER LO
0069  6014  85 00                STA TABNAM         ;
0070  6016  A9 60                LDA #>EPTAB        ; SAME FOR NAME POINTER HI
0071  6018  85 01                STA TABNAM+1       ;
0072  601A  B1 1F        NXTCHR LDA (SUBNAM),Y     ; GET NEXT CHAR OF CALL NAME
0073  601C  D1 00                CMP (TABNAM),Y     ; SAME AS E.P.T NAME?
0074  601E  D0 0D                BNE NXTENT         ; IF NOT, THEN BRANCH TO NXTENT
0075  6020  C8                   INY                ; BUMP CHARACTER POINTER
0076  6021  C0 06                CPY #NAMSIZ        ; HAVE WE MATCHED ALL CHARS
0077  6023  D0 F5                BNE NXTCHR         ; IF NOT, GO MATCH NEXT CHAR
0078  6025                ;
0079  6025                ; ********************************
0080  6025                ; *                              *
0081  6025                ; * CONTROL WILL COME HERE WHEN  *
0082  6025                ; * WE SUCCESSFULLY 'LOOK-UP'    *
0083  6025                ; * THE SUBROUTINE NAME.         *
0084  6025                ; *                              *
0085  6025                ; ********************************
0086  6025
0087  6025  B1 00                LDA (ENTPNT),Y    ; LO BYTE OF SUBRTN ENTRY-POINT
0088  6027  48                   PHA               ; AND STACK IT
0089  6028  C8                   INY               ; BUMP TO THE HI BYTE
0090  6029  B1 00                LDA (ENTPNT),Y    ; NOW THE HI BYTE
0091  602B  48                   PHA               ; GETS STACKED
0092  602C                ;
0093  602C                ; NOW THE STACK HAS BEEN PRIMED. SO THE FOLLOWING
0094  602C                ; RTS INSTRUCTION WILL PASS CONTROL TO THE RQUESTED
0095  602C                ; SUBROUTINE.
0096  602C                ;
0097  602C  60                   RTS               ; D I S P A T C H E D !
0099  602D                ;
0100  602D                ; ********************************
0101  602D                ; *                              *
0102  602D                ; * CONTROL COMES HERE WHEN THE  *
0103  602D                ; * CALL NAME AND E.P.T. NAME    *
0104  602D                ; * FAIL TO MATCH. IF THERE ARE  *
0105  602D                ; * MORE NAMES IN THE E.P.T WE   *
0106  602D                ; * TRY AGAIN, ELSE IT'S RATHER  *
0107  602D                ; * BAD NEWS FOR THE USER!       *
0108  602D                ; *                              *
0109  602D                ; ********************************
0110  602D  A0 00        NXTENT LDY #ZERO          ; RE-INITIALISE CHARACTER PNTR
0111  602F  A5 00                LDA TABNAM        ; COMPUTE ADDRESS OF NEXT ENTRY
0112  6031  18                   CLC               ; PREPARE FOR ADDITION
0113  6032  69 08                ADC #ENTLEN       ; ADD ENTRY LENGTH
0114  6034  85 00                STA TABNAM        ; SAVE BACK NEW ADDRESS
0115  6036  A5 01                LDA TABNAM+1      ; MAYBE WE HAD A CARRY
0116  6038  69 00                ADC #ZERO         ; SO CATER FOR THE POSSIBILITY
0117  603A  85 01                STA TABNAM+1      ; JUST IN CASE
0118  603C  CA                   DEX               ; ONE LESS ENTRY TO COMPARE
0119  603D  D0 DB                BNE NXTCHR        ; BRANCH IF MORE ENTRIES
```

```
0120  503F                ;
0121  603F                ; ********************************
0122  603F                ; *                              *
0123  603F                ; * CONTROL COMES HERE IF THE    *
0124  603F                ; * CALLED SUBROUTINE DOES NOT   *
0125  603F                ; * EXIST, IN WHICH CASE THE     *
0126  603F                ; * INTREPID USER WILL BE GIVEN *
0127  603F                ; * BASIC'S VERSION OF THE       *
0128  603F                ; * TWO-FINGERED SALUTE, I.E     *
0129  503F                ; * THE MESSAGE 'SYNTAX ERROR'   *
0130  603F                ; * WILL BE DISPLAYED, AND THE   *
0131  503F                ; * LITTLE OL' PROGGY WILL COME *
0132  603F                ; * TO A GRINDING HALT IN TRUE   *
0133  503F                ; * USER-FRIENDLY FASHION!       *
0134  603F                ; *                              *
0135  503F                ; ********************************
0136  603F                ;
0137  603F  20 00 BF           JSR ERRMSG       ; 'ON YER BIKE!'
0139  6042                ;
0140  6042                ; ********************************
0141  6042                ; *                              *
0142  6042                ; * HERE IS THE DISPATCHER'S     *
0143  6042                ; * ENTRY-POINT-TABLE (E.P.T)    *
0144  6042                ; *                              *
0145  6042                ; * ENTRIES ARE EIGHT BYTES LONG*
0146  6042                ; * AND CONSIST OF A 6-CHARACTER*
0147  6042                ; * SPACE-FILLED NAME, FOLLOWED *
0148  6042                ; * BY THE ENTRY ADDRESS MINUS 1*
0149  6042                ; * (WHICH GETS PUSHED ONTO THE *
0150  6042                ; * STACK FOR A CRAFTY RTS JUMP *
0151  6042                ; *                              *
0152  6042                ; ********************************
0153  6042                ;
0154  6042                ;
0155  6042  43 41        EPTAB  .BYTE   'CASELO'   ;
0156  6048  61 FF              .DBYTE   CASELO-1   ; SWITCH TO LOWER CASE ROUTINE
0157  604A  43 41              .BYTE   'CASEUP'   ;
0158  5050  62 05              .DBYTE   CASEUP-1   ; SWITCH TO UPPER CASE ROUTINE
0159  6052                ;
0160  6052                ; FUTURE ENTRIES CAN BE DEFINED HERE
0161  6052                ;
0162  6052                EPTLEN = *-EPTAB          ; LENGTH OF E.P.T.
0164  6052                ;
0165  6052                ; WARNING - IF 'EPTLEN' ADDRESS IS > THAN $61FF
0166  6052                ; THEN ADJUST THE FOLLOWING LIBRARY START ADDRESS
0167  6052                ;
0168  6052                ; ********************
0169  6052                ; * START LIBRARY ON *
0170  6052                ; *                  *
0171  6052                      *=$6200
0172  6200                ; *                  *
0173  6200                ; *   PAGE BOUNDARY  *
0174  6200                ; ********************
0175  6200                ;
0176  6200                ; ********************************
0177  6200                ; *                              *
0178  6200                ; *      L I B R A R Y           *
0179  6200                ; *                              *
0180  6200                ; *    R O U T I N E S           *
0181  6200                ; *                              *
0182  6200                ; *       S T A R T              *
0183  6200                ; *                              *
0184  6200                ; *         H E R E.             *
0185  6200                ; *                              *
0186  6200                ; ********************************
0187  6200                ;
0188  6200                ;       ***************
0189  6200                ;       *             *
0190  6200                ;       * C A S E L O *
0191  6200                ;       *             *
0192  6200                ;       ***************
0193  6200                ;
0194  6200  A9 0E        CASELO LDA #LOWER
0195  6202  8D 4C E8           STA CASE
0196  6205  60                RTS
0197  6206                ;
0198  6206                ;       ***************
0199  6206                ;       *             *
0200  6206                ;       * C A S E U P *
0201  6206                ;       *             *
0202  6206                ;       ***************
0203  6206                ;
0204  6206  A9 0C        CASEUP LDA #UPPER
```

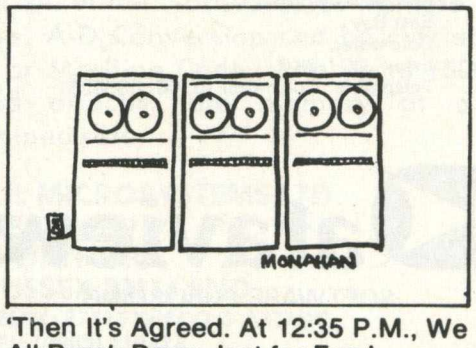
```
0205  6208  8D 4C E8          STA CASE
0206  620B  60                RTS
0207  620C                ;
0208  620C                ;     *************
0209  620C                ;     *           *
0210  620C                ;     * W A T C H *
0211  620C                ;     *           *
0212  620C                ;     *  T H I S  *
0213  620C                ;     *           *
0214  620C                ;     * S P A C E *
0215  620C                ;     *           *
0216  620C                ;     *************
```

LISTING 2.

```
 10 POKE 53,112   : REM PROTECT USER-LIB
 20 X=X+1         : REM BOOTSTRAP FLAG
 30 IF X < 2 THEN  LOAD "USER-LIB",8
 40 :
 50 TEM=24576     : REM DISPATCHER START
 60 FOR I=1 TO 12
 70 PRINT"■     I'M GETTING GOOD VIBRATIONS
 80 NEXT
 90 REM AND NOW WE CALL THE LIBRARY
 100 SYSTEM,"CASEUP" :
 110 SYSTEM,"CASELO"
 120 GOTO 100
READY.
```

Just for fun ... Snoopy on a Vic20!

```
 100 rem pet benelux
 110 rem exchange
 120 rem netherlands
 130 poke56,peek(56)-2:run140
 140 cs=peek(56)*256
 150 c$=chr$(34)
 160 poke36879,42:poke36869,255
 170 print"[clr,ctrl2,8cd,7sp]@abcdef
 180 print"       hijklmn
 190 print"       pqrstuv
 200 print"         z[\]
 210 print"        !"c$'#$%&
 220 print"       ()*+,-."
 230 readx:ifx=-1then260
 240 fori=xtox+7:reada:pokei,a:next
 250 goto230
 260 geta$:ifa$=""then260
 270 print"[clr,ctrl7]";:poke36879,27
 275 poke36869,240:poke56,peek(56)+2
 278 end
 280 data7424,0,0,0,0,0,0,0,0
 290 data7168,0,0,0,0,0,3,12,16
 300 data7176,3,4,9,19,247,23,11,12
 310 data7184,192,48,200,228,247,231
 315 data200,48
 320 data7192,0,0,0,0,3,252,0,64
 330 data7200,0,0,7,56,192,0.0,0
 340 data7208,0,0,192,48,12,3,0,0
 350 data7216,0,0,0,0,112,136,232,248
 360 data7232,32,64,128,131,132,132
 365 data132,68
 370 data7240,15,24,48,225,65,33,33,33
 380 data7248,192,128,128,0,0,0,0,0
 390 data7256,64,0,31,32,64,128,128
 395 data128
 400 data7264,0,0,255,0,0,0,0,0
 410 data7272,0,3,252,0,0,0,6,57
 420 data7280,240,0,0,0,0,0,0,128
 430 data7296,66,33,16,12,3,0,0,0
 440 data7304,33,192,0,0,0,224,31,0
 450 data7312,0,128,64,48,15,0,255,0
 460 data7320,64,32,31,0,31,248,68,250
 470 data7328,0,0,224,64,129,2,4,8
 480 data7336,74,105,99,132,7,8,16,32
 490 data7344,64,224,32,192,128,0,0,0
 500 data7376,1,1,3,3,6,6,9,9
 510 data7384,209,160,96,100,194,225
 515 data208,168
 520 data7392,208,56,7,0,0,0,128,64
 530 data7400,64,128,0,224,24,4,2,1
 540 data7432,0,0,0,0,0,0,1,254
 550 data7440,18,18,38,38,76,148,20,40
 560 data7448,228,230,232,103,49,30,1
 565 data1
 570 data7456,120,8,48,16,224,2,2,2
 580 data7456,120,8,48,16,224,2,2,2
 590 data7464,1,1,2,4,56,8,8,7
 600 data7472,0,0,0,0,0,0,0,252
 610 data7488,1,63,15,255,0,0,0,0
 620 data7496,0,255,128,255,0,0,0,0
 630 data7504,40,200,16,224,0,0,0,0
 640 data7512,63,64,128,127,0,0,0,0
 650 data7520,1,0,0,255,0,0,0,0
 660 data7528,255,0,36,255,0,0,0,0
 670 data7536,2,146,252,128,0,0,0,0
 680 data-1
ready.
```



'Then It's Agreed. At 12:35 P.M., We All Break Down Just for Fun.'

PET ASSEMBLER & DISASSEMBLER

Available for all PETs 8K and upwards, 2001 to 8032.

Full details, CPUCN July '81.

Assembler handles data tables in Decimal, Hex., or ASCII.

On Cassette:

Assembler	£15.00
Dissembler	£10.00

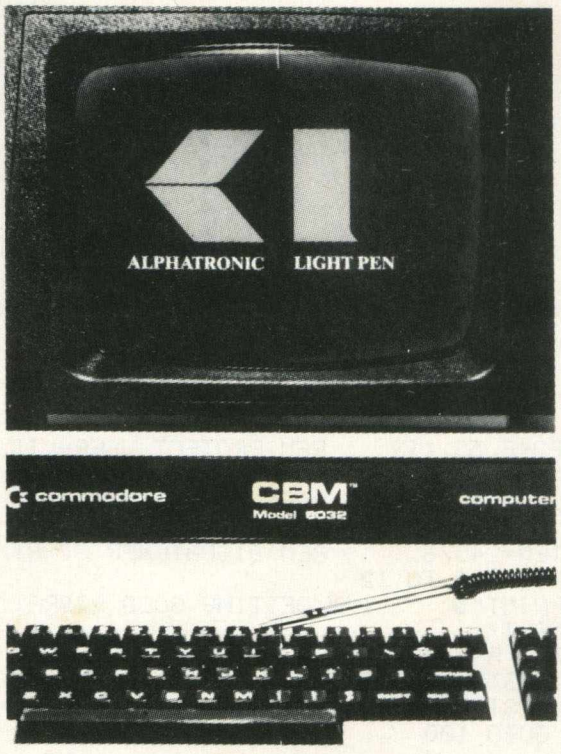
On Disk (8050 only, BASIC 4):

Both Programmes	£28.00

Specify machine type

Patrick Walshe,
Loughteague, Stradbally, Laois,
Ireland

FOR EXCELLENCE AND DEPENDABILITY AND USE ON MOST MICROCOMPUTERS



ALPHATRONICS MICROPROCESSOR APPLICATIONS LTD.,
Norfolk House, 435 Silbury Boulevard, Central Milton Keynes MK9 3HB
Telephone: (0908) 663906 Telex: 825016 ROCOCO G

PROGRAMMER
PRODUCT SUPPORT
Colchester

We are both Software publishers and Hardware suppliers, having achieved an annual turnover in sales approaching £3m over the last 2½ years.

Our requirement is for a Programmer with experience of CBM operating systems and the 6502 Assembler to join our Software Support Group. Excellent salary and conditions can be negotiated.

For full details, please phone or write to:

Mr. Sid Newman,
General Manager,
Dataview Ltd.,
Portreeves House,
East Bay,
Colchester,
Essex, CO1 2XB
Telephone: Colchester (0206) 865835

Dataview
SOFTWARE PUBLISHING

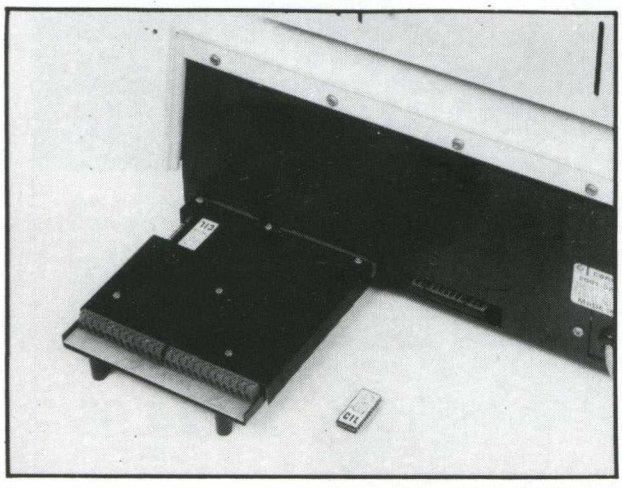
ADVERTISER'S INDEX

30 .....Alpha Business Systems
54 .....Alphatronic
11 .....A.S.K.
42 .....Audiogenic
12, 13.....Bristol Software Factory
17 .....C.B.S.
32 .....Chaswood
55 .....C.I.L.
19 .....Comal Users Group
19 .....Compufix
41 .....C.S.E.
2 .....D.A.M.S.
54 .....Dataview
17 .....Dynatech
17 .....Greenwich Instruments
11 .....Healey
31 .....Impetus
9 .....J.J. Lloyd Instruments
48 .....Kingsley Computers
26 .....Landsoft
19 .....Level Ltd.
47 .....Mass Micros
18 .....Ortholog
5 .....Peach Data Services
35 .....Popular Computing Weekly
32 .....Qwerty Computer Services
8 .....Rabbit Software
30 .....Small Systems Engineering
56 .....Stack Computer Services Ltd.
30 .....T.A.L.
19 .....Tirith
54 .....Patrik Walshe
47 .....Wirt Microsystems

A NOTICE TO ALL

PET LOVERS

INTRODUCE YOUR PETS TO OURS
AND SOLVE YOUR INTERFACE PROBLEMS



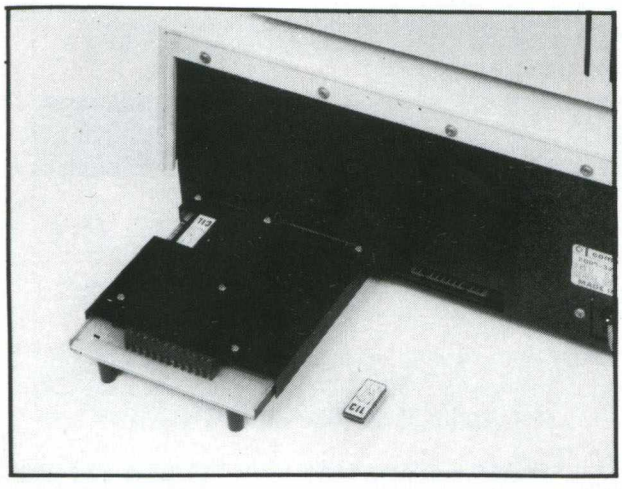
Only £195.00

ANALOGUE/DIGITAL I/O

THE PUPI

- **4 ANALOGUE INPUTS (12BIT)**
- **2 ANALOGUE OUTPUTS (12 BIT)**
- **4 RELAY OUTPUTS**
- **4 LOGIC INPUTS**

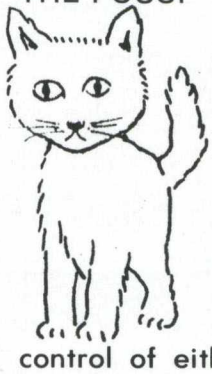
When connected to the "PET" User Port the PUPI gives you all the above features together with an operating system in EPROM, which interacts with Basic's variables, giving extremely simple operation. Inputs and outputs are ±10V and relays are rated at 10VA. Logic inputs can be used for microswitch sensing etc.



Only £195.00

HIGH SPEED A/D CONVERTER

THE PUSSI



- **4 ANALOGUE INPUTS (12 BIT)**
- **50 MICROSECOND CONVERSION**
- **STOP AND START TRIGGERS**
- **DATA ACQUISITION SOFTWARE**

Using an operating system in EPROM, the PUSSI provides a high speed A-D Converter with 4 multiplexed inputs, which is under control of either software, or remote start/stop triggers. A-D Conversion can be carried out from Basic, or Machine Code, with up to 1500 readings entered directly into memory at a software determined rate.

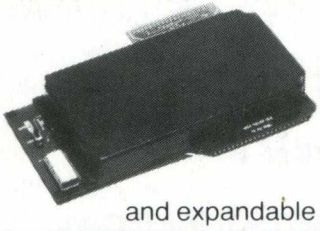
CIL
MICROSYSTEMS LTD

CIL MICROSYSTEMS LTD.
DECOY RD.,
WORTHING,
SUSSEX BN14 8ND.
TELEX: 87515 WISCO G ATTMIC
TEL: (0903) 210474

Write, phone or to
obtain further information circle number

VIC-20 ACCESSORIES FROM STACK

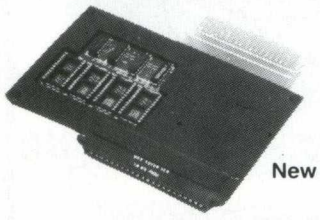
STACK STOREBOARD (memory expansion unit)



Power up your VIC-20 to a MASSIVE 32k COMPUTER!!

only **£49.00** (plus VAT) for 3k and expandable to 27k on the same board.

STACK ROM SWITCHBOARD



Use up to 4 ROMs at once! eg. games. ROMs. VICKIT. VICKIT II etc.

New Lower Price!! **£29.00** (plus VAT)

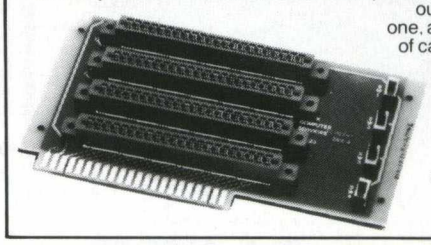
STACK 8k RAMPACK

Use this upgrade pack to increase memory size on Stack Storeboard by 8k a time.

NEW LOWER PRICE!! only £29.00 (plus VAT)

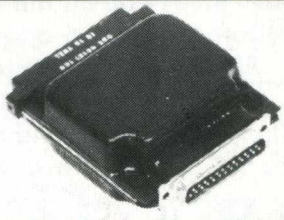
NEW STACK 4-SLOT MOTHERBOARD

Fits into the port at the rear of the Stack Storeboard or the Stack Low Cost 3k Memory and enables you to use up to 4 cartridges in addition to the extra memory on the Storeboard or 3k Memory. Each slot can be switched in or out, thus allowing the use of one, all four, or any combination of cartridges without having to remove them.



only **£24.99** (plus VAT)

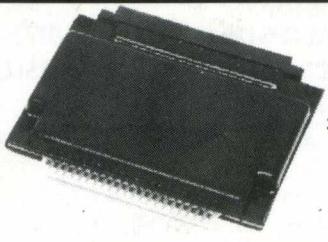
STACK LOW COST RS232 INTERFACE



Allows you to use a serial printer with your VIC-20.

£22.99 (plus VAT)

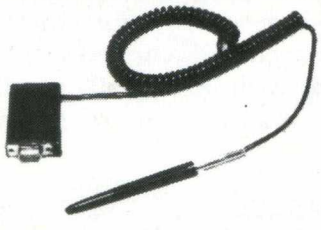
STACK LOW COST 3k MEMORY



The lowest costing memory addition gives you 6½k of user memory on your VIC-20. Also allows you to use those quality games which demand 3k of Hi-Res Graphics! Socket at rear allows you to stack up further accessories.

only **£25.99** (plus VAT)

STACK LIGHTPEN



Allows you to use VIC-20 without keyboard entry by simple programming. Sensor in pen sees the TV screen! Ideal for education. games. menu selection etc.

only **£25.00** (plus VAT)

STACK VICKIT SERIES

A series of ROMs which greatly simplifies programming and enhance the qualities of your VIC-20. Fits into Stack ROM SWITCHBOARD or Stack STOREBOARD.

VICKIT

Offers HELP to programmers....it also offers AUTO. DELETE. DUMP. FIND. OFF. RENUMBER. STEP. TRACE.

£25.00 (plus VAT)

Special Offer Price if Purchased with STOREBOARD only **£15.00** (plus VAT)

VICKIT II

A 4k ROM offering ALL THE FEATURES OF VICKIT plus.....TEXT. GRAPHICS. LINE. CLEAR. DRAW. PUT. FILL. SET. POINT

only **£29.00** (plus VAT)

Other exciting additions to the VICKIT series due soon!

Contact your local Commodore VIC dealer for details.

Stack Computer Services Limited, 290-298 Derby Road, Bootle, Merseyside. 051-933 5511. Telex: 627026.