

IPUG

NUMBER 6
VOLUME 3
NOV. 1981

INDEPENDENT PET
USERS
GROUP



INDEPENDENT

PET USERS GROUP

Vol. 3, No. 6

Newsletter

Nov. 1981

Europe's first independent magazine for PET users

Page	Contents
134	Editor's Notebook
135	MIKRO Bug
135	Day of the Week
136	Strictly for Beginners
137	Matters Arising
138	Three-Dimensional Axis Rotation
139	Round the Regions
141	BASIC 4.0 in the 8000 Series
142	The Annual General Meeting
144	Printer Shifting
144	Members' Sales and Wants
145	Program Testing
147	Games Upgrades
148	Photos
150	VIC Matters
154	Screen Input
156	Screen Output Routines
157	IPUG Projects
158	Review - HR-40 Graphics Board
160	Shop Window
161	CBM Column
162	Review - Arrow ROM
163	8000 - Series Printer Note
164	Program Snippets

The opinions expressed herein are those of the author and not necessarily those of IPUG or the editor. Items mentioned in "Shop Window" are culled from advertisers' material and IPUG do not necessarily endorse or recommend such items - *caveat emptor*

EDITOR'S NOTEBOOK

Recently Commodore published a list (CCN Sept '81) of known bugs in DOS and printer software. The details and fixes, where known, are too extensive to mention here, but summarising: 'there is no version of BASIC, DOS or printer software in any Commodore microprocessor equipment which is bug-free'. Don't be alarmed. In this respect Commodore are not alone and are in good company with other micro, mini and mainframe manufacturers and this is why, for every system a user group exists.

Software is an incredibly complex item and a time must come when a sizeable piece of software must be called 'complete and working' and enter the market where, instead of 1 or 2 people spending thousands of hours (= pounds) looking for bugs, there are thousands of people finding bugs (in 1 or 2 hours?). A program published recently in 6502 Micro Journal had a patch to fix an 'odd phenomenon' - the author had unwittingly stumbled upon a bug (discovered by Mike Todd) without knowing the cause or cure, but had the sense to document his ignorance in a REMark for future reference. No doubt someone will enlighten him.

Well what did you find lacking in the 'Compendium'? Was it perhaps an ASCII to DEC/HEX conversion table, or some useful collection of notes or data? In the pipe-line are two more IPUG publications on the lines of the Compendium. The first will be known as 'The IPUG ROM Gazetteer', since its substance will be the Compendium ROM List, revamped and greatly expanded to cover BASIC4 and the VIC. In addition the 'coloured pages' which caused problems for some people in respect of legibility, will be conventional. Let me know what extras you would like included and I will see what can be done. In terms of input, offers of useful, detailed and above all, accurate information will be most welcome. When will it be published? - when it's ready....

R.D.G.

MIKROBE

I knew that if I looked long enough I would find one. The MIKRO assembler chip contains a 'bug'. It occurs thus:

```

LDA ADDRESS      ;UNDEFINED LABEL REF
ADDRESS=$4F      ;DEFINE
JSR SUB          ;SUBSEQNT LABEL REF
END
SUB NOP          ;CORRECT ENTRY
BUG RTS         ;BUG ENTRY

```

The first pass finds the source code syntactically correct (which it is), but MIKRO does not generate a symbol table and, in the absence of errors, creates code on the second pass. However, in the example ADDRESS is not defined at first and consequently a call to 'SUB' actually goes to 'BUG', - or elsewhere. Subsequent absolute forward referencing labels are in error.

R. D. G.

--o0o--

WHAT DAY IS IT ?

```

100 J$(1)="SUNDAY":J$(2)="MONDAY"
160 J$(3)="TUESDAY":J$(4)="WEDNESDAY":J$(5)="THURSDAY"
220 J$(6)="FRIDAY":J$(7)="SATURDAY"
260 INPUT"<clr>DATE-THUS 24,12,1947";D,M,Y:IFY>1752THEN340
300 PRINT"YEAR MUST NOT BE PRIOR TO 1752":GOTO260
340 K=INT(0.6+(1/M)):L=Y-K:P=L/100
360 Z=INT(13*(M+12*K+1)/5)+INT((5*L)/4)-INT(P)+INT(P/4)+D-1
380 Z=Z-(7*INT(Z/7))+1
400 PRINT"<clr>THE DATE";D;M;Y;"WAS OR WILL BE A <6sp>";
   "<rvs>"J$(Z)
420 INPUT"<4dn>FOR NEXT DATE TYPE YES OR NO";L$
460 IFL$="YES"THENPRINT:GOTO260
480 IFL$="NO"THEN560
500 PRINT"DO IT PROPERLY!!":GOTO420
560 PRINT"GOODBYE"

```

--o0o--

STRICTLY FOR BEGINNERS

So you've got your PET and now you want to start using it. The Commodore Manual is some help, but not as easy to understand as I would have liked when I first read mine.

Let us start with the memory inside your PET. I do not mean now how much memory you have available (e.g. 8, 16, or 32K - kilobytes or characters by the thousand, 1024 actually) but how PET handles what you do to it on the keyboard.

Each memory position, or pigeon-hole, can be allocated a name. This name can be a letter, two letters or a letter and a number if the amount to be stored in it is a number. If you intend to store something which is not a number, then the name must be followed by a '\$' sign. This is so that the computer knows what to expect when it is told to look into storage spaces specified by you.

Let us try something practical to illustrate this: Type in the following short program; n.b. all programs must have line numbers. The PET starts executing the program at the lowest line number and continues to the end, which is usually the highest number.

```
10 PRINT"<clr> HELLO, WHAT IS YOUR NAME ?"
20 INPUTN$
30 PRINT"NICE TO MEET YOU, "N$" MY NAME IS PET."
40 PRINT"NOW LET US DO SOME ARITHMETIC, "N$"
50 INPUT"TYPE IN TWO NUMBERS, & I WILL MULTIPLY THEM";A,B
60 C=A*B:PRINT:PRINT"THE ANSWER IS";C
```

This tiny program shows how PET accepts letters, or alphanumeric characters in what are called 'string variables' (alphanumeric meaning letters and/or numbers either together or separately). Variable is the name given to the space in PET's memory which you have designated with a particular name. Variable because you can change the contents at any time. For instance, if you type in the above program and then type RUN, then when the PET asks you

to enter the two numbers, you type in say, 4 and 5, then PET will print the answer on the screen as 20. If you then type RUN again, (since the program ends after printing the answer) and this time type in say, 3 and 6, then the answer 18 will appear. What you have done is, first of all you placed a 4 in PET's memory into a space called (by you) A, then you placed 5 in compartment B, and PET itself placed the answer 20, in variable C.

The second time of running the program, you altered A to 3 and B to 6. PET altered the contents of C to 18. So you see the contents were varied and that is why they are known as variables. Because we knew that your name was going to consist of letters only, we designated its variable as N\$. PET then knew that this was to be treated differently to A, B or C. How differently is too complex to explain at this stage, so I will leave it at that for now. What I would like from now on is your requests for information on anything you do not understand.

I found that when I got my PET I had all sorts of problems with the cassette in particular, and could find nobody to answer my queries. Having bought the machine at an out-of-town place, I was constantly telephoning the suppliers who tried to explain, but it was very difficult over the telephone. Joining I.P.U.G. and forming a local group was my solution. The experts came along and everything was revealed in due course. I think that someone to write to and a magazine to read must be a very good way to solve problems.

Ray Davies, 105, Normanton Road, Derby, DE1 2GG.

--oOo--

MATTERS ARISING

On p97 the impression may have been given that it is not possible to upgrade a 2040 disk drive to DOS2.1. This is, in fact, not so. Although there is a faint possibility that very, very early disk drives with DOS1.0

--oOo--

THREE-DIMENSIONAL AXIS ROTATION

By Stephen Binns

Any point in three dimensional space can be represented by a three element vector i.e. (x y z). In order to visualise the data it must be projected onto a plane e.g. x,y which is simply a matter of leaving out the z - co-ordinate. To obtain a perspective view rotate the data about the origin and this can be done by means of matrix multiplication. Applications are perspective diagrams, 3-D graphics, alt-azimuth systems for telescopes and satellite aerals, polar to cartesian conversion and in two dimensions corrections for paper alignment for digitizers.

Generally:

$$(x' \ y' \ z') = (x \ y \ z) * R$$
 where R is as follows:

x axis rotation by A radians

$$\begin{array}{ccc} 1 & 0 & 0 \\ 0 & \cos(A) & -\sin(A) \\ 0 & \sin(A) & \cos(A) \end{array}$$

y axis rotation by B radians

$$\begin{array}{ccc} \cos(B) & 0 & \sin(B) \\ 0 & 1 & 0 \\ -\sin(B) & 0 & \cos(B) \end{array}$$

z axis rotation by C radians

$$\begin{array}{ccc} \cos(C) & -\sin(C) & 0 \\ \sin(C) & \cos(C) & 0 \\ 0 & 0 & 1 \end{array}$$

```

1000 DIM D(N-1,2),R(2,2),V(2)
2000 REM SUBROUTINE TO ROTATE DATA IN SITU
2010 FORI=0TON-1:FORJ=0TO2:V(J)=0:FORK=0TO2
2020 V(J)=V(J)+D(I,K)*R(K,J):NEXT:D(I,J)=V(J):NEXT:NEXT
2030 FORJ=0TO2:D(I,J)=V(J):NEXT:NEXT:RETURN
  
```

--o0o--

Stage One Computers (Bournemouth) have a change of address: 300, Ashley Road, Upper Parkstone, Poole, Dorset. Tel: 735656.

--o0o--

ROUND THE REGIONS

Region B - Glos./Bristol Area: Mrs Janet Rich, 23, Sheppard Leaze, Wotton-under-Edge, Glos. Tel: Wotton 2498. Informal meetings last Friday of every month, except December, at the above address. Current project is trying to join up computers over short distances.

Region C - Hampshire Area: Ron Geere, 109, York Road, Farnborough, Hants, GU14 6NQ. Informal Meetings 3rd Wednesday of the month at the Civic hall, Fleet, 7.30 p.m.

Region D - South East Region: Wing Cdr. Mike Ryan, 164, Chesterfield Drive, Riverhead, Sevenoaks, Kent. Tel: (0732) 53530. Meetings every 3rd & 4th Thursday of every month at Charles Darwin School, Jail Lane, Biggin Hill, Kent at 7.30 p.m. - 19th Nov Commodore demonstration of their latest products. 26th Nov. & 10th Dec., open night. 17th Dec., demo./programming clinic on VIC by Harry Broomhall.

Region D - there is also a Canterbury group in this area, contact John Bickerstaff, tel: (0227) 9102 for further information.

Region G - Walter Green, 151, The Hatherley, Basildon, Essex, SS14 2QH. Walter has agreed to set up a group and we should be hearing from him shortly.

Region H - Norfolk/Suffolk area: We need a volunteer to help in this region.

Region I - Herts/Watford Group: Stephen Rabagliati, Inst. of Grocery Distribution, Letchmore Heath, Watford, Herts. Tel: (09276) 7141, Extn.39.

Region I - Herts/Stevenage Group: Brian Granger, Tel: Stevenage 727925. Meetings at Provident Hut, Assurance Building, Purwell Lane, Hitchin, Herts., on last Wednesday every month.

Region J - Wales. Contact F.J.Townsend, The Mill, Rhydowen, Llandyssul, Dyfed, SA44 4QD. Tel: (054) 555291.

Region K - Birmingham Area: J.A.McKain, P.P.I. Ltd., 177, Lozells Road, Birmingham. Tel: 021-554 0202. We have the venue, but need some help - any offers ?

Region K - South Midlands: M.J.Merriman, 12, York St., Stourport-on-Severn. Meetings on the last Thursday of every month at the above address, where one can receive help with business programming problems.

Region L - Derby Area: R.Davies, 105, Normanton Road, Derby. Tel: (0332) 41025. Meetings are the first Monday of every month at Davidson Richards Ltd., 14, Duffield Road, Derby.

Region N - Manchester Area: we have the venue, but need someone willing to help in Manchester.

Region N - Blackpool Area: David Jowett, 197, Victoria Road East, Thornton-Cleveleys, Blackpool. Tel: Cleveleys 869108. Meetings every 3rd Thursday at Arnold School, Blackpool.

Region N - Liverpool Area: Tony Bond, 27, Ince Road, Liverpool 23. Tel: 051-924 1505. Informal meetings every 2nd Thursday at the Merchant Taylor School for Boys in Crosby.

Region P - London: Barry Miles, Department of Business Studies, Polytechnic of North London, Holloway Road, London, N7. This is the PET group of the Amateur Computing Club [Ad. - Ed] which meets every other Tuesday at the Poly at 6 p.m.

Region P - West London. Geoff Squibb, 108, Teddington Park Road, Teddington, Middx. Tel: 01-977 2346.

Region S - Durham Area: Jim Cocallis, 20, Worcester Road, Newton Hall Estate, Durham. Tel: (0385) 67045. Meetings Lawson School, Burtley.

Region T - Scotland: Dr. Jim MacBrayne, 27, Paidmyre Crescent, Newton Mearns, Glasgow. Tel: 041-639 5686.

Region W - Yorkshire: Bob Wood, 13, Bowland Crescent, Ward Green, Barnsley, S. Yorks. Tel: (0246) 811585.

All enquiries and regional group material for publication should be sent to me as regional co-ordinator.

Eli Pamphlett.

--o0o--

BASIC 4.0 in the 8000 SERIES PET

I have not seen any mention of the differences between the ROM sets in the 4000-Series PETs and those in the 8000-Series machines. I have found that all the ROMs are the same except that at \$E000 to \$E7FF (i.e. the 2K ROM). This is numbered: 901447-29 in the 4000 Series and 901474-04 (or -03) in the 8000 Series. I understand that -04 is an upgrade fitted in more recent machines to correct a bug in the earlier -03 version.

The other BASIC 4.0 ROMs are numbered: 901465- with the suffixes: \$F000-\$FFFF -22 \$D000-\$DFFF -21 \$C000-\$CFFF -20 and \$B000-\$BFFF -23. This -23 is also an upgrade although I am not sure of the number of the earlier ROM.

David W. Jowett
197 Victoria Road East
Thornton
BLACKPOOL. FY5 3ST

--o0o--

PROMISES, PROMISES....

At the PCW show I netted Marck Pearlstone who assured me that all versions of 'The Last One' (p120) will be available by end of Nov/early Dec. Delivery of the Tandy II version is reported to have actually started.

R.D.G.

--o0o--

THE ANNUAL GENERAL MEETING

Our recent A.G.M. was not as well attended as we would have liked, but thank you to everyone who did make the effort to come. At the meeting Mick Ryan, our new chairman, was introduced and told the meeting about the improved rapport with Commodore. I.P.U.G. is now officially recognised. Various problems which had been reported by members have been sorted out and in addition, our stand at the PET Show was provided free of charge (dealers had to pay). Mick stressed the need for us to make the most of this goodwill for the benefit of the membership.

Regarding the regional groups, they are now well established; the two new groups in Hertfordshire have been well attended and we had an offer from someone to organise one in Essex. More details are given in the regional report.

Ron Geere stressed the need for more contributors to the magazine on a regular basis. He would like material in 'machine-readable form'. He was asked to explain in the magazine what this meant.

The following were elected to serve on the national committee:

Chairman, Mick Ryan, will deal mainly with policy problems, technical queries and be our liason official with Commodore.

Treasurer, Alan Birks, is responsible for the Group's accounts.

Eli Pamphlett, General Secretary and regional co-ordinator, will deal with any administrative problems and regional enquiries.

Editor, Ron Geere, has the responsibility to produce and edit the magazine.

Software Librarian, Bob Wood, will operate the software library service and give assistance to members with software problems.

Discount Schemes, David Annals, any discounts offered or required to 142, Windermere Road, Norbury, London, SW16 5HE.

Membership Secretary, Jack Cohen, all membership enquiries (formerly dealt with by June Dowson) 30, Brancaster Road, Newbury Park, Ilford, Essex. Tel: 01-597 1229.

VIC-20 Co-ordinator, Mike Todd, any VIC-20 matters, including material for publication. Address: 27, Nursery Gardens, Lodgefield, Welwyn Garden City, Herts.

Projects: Fred Offler (see p157).

General Dogsbody, Terry Pamphlett [I didn't write this, honest! - Ed] 7, Lower Green, Tewin, Welwyn, Herts. Tel: Welwyn 7325.

Publicity Officer - vacancy. Volunteer urgently required. Responsible for advertising the existence of I.P.U.G. and arranging advertising in the Newsletter.

It was decided that all regional organisers automatically become members of the Committee for a trial period of one year, and will receive the agenda and minutes accordingly.

Mike Todd will be in charge of the VIC-20 and in addition to subediting our VIC column, will also administer local groups which initially will run in conjunction with our PET Users Group. At this stage a name change was suggested since, constitutionally, we cater for all Commodore machines and peripherals. 'Independent Commodore Users Group' was suggested. However, a change of name was ruled out of order, and a referendum was agreed upon after a lengthy discussion. The referendum will go out at the same time as the subscription renewal notices.

The committee was pleased to report the need for only £ 1.00 rise in the annual subscription to £ 7.50 (UK). It has been suggested from regions that we have an all-in subscription which will also cover regional expenses. This is being investigated and comments would be welcome.

This is only a summary of the A.G.M., the complete minutes will be read at the next A.G.M.

Eli Pamphlett.

PRINTER SHIFTING

Commodore printers have the facility to print both in graphics mode and in text mode - just like the PET screen. POKE59468,12 puts the screen into the graphics mode (giving upper case and graphics characters) and this is how the printers will print when powered up. POKE59468,14 puts the screen into text mode (with lower and upper case) - but printing this to the printer can be a problem.

All the printer manuals point out that a cursor down character will shift into text mode (just like POKE59468,14) and a cursor up will restore the graphics mode. Unfortunately the effect of this switch only lasts for the current line and as soon as a new line is printed the printer reverts to its normal graphics mode.

Printers with ROM number -04 (and possibly subsequent printers) have an additional secondary address which will force the printer into text mode permanently (although cursor down now switches it into graphics mode). This is done by OPEN7,4,7:PRINT#7 - the only way to restore normal operations is to switch the printer off and on again.

To get a permanent text mode (which can be useful for listings) without the extra secondary address means intercepting the output and adding a cursor down following every carriage return, or before a line is printed.

M.R.T.

--o0o--

MEMBERS SALES & WANTS

Wanted: A secondhand Commodore cassette tape recorder in good condition, with connecting lead for use with 8032 computer. Please contact: Frank J. Pycroft, PI-COPY, 70, Parkview Road, London, SE9 3QW. Tel: 01-850 5975 (Ansafone).

--o0o--

PROGRAM TESTING

By Andy Rigby

The phases of producing a commercial computer program or set of programs can generally be split into:-

1. systems analysis 2. flowcharting 3. coding 4. debugging 5. testing 6. documentation 7. production. Although there is obviously some order in this list, the phases usually tend to merge together, sometimes with parts omitted, especially in the microcomputer world where things are on a smaller scale.

Program testing is quite often given less priority than it needs. There are several reasons for this - firstly, the creative part has been done and testing a program can be repetitive and unrewarding. Secondly, by now the program is probably behind schedule and you are being badgered for an end product.

We should not confuse 'testing' with 'debugging'. Debugging is the 'weeding out' of errors in the syntax or flow of the program code which would generally prevent the thing from running at all under normal conditions. When debugging is completed, the program definitely solves some problem. Testing seeks to guarantee that it is the problem that was intended, and that further problems are not produced as a spin-off of the usage of the program.

Testing a program involves partly science and partly skill derived from experience. The science is in predicting what ought to happen when the program is run; the skill comes in knowing what to test for, how to test for it, and how far to test something before being confident of its reliability.

In generalised form, testing can be described as running a program with known test data that should lead to known results.

Methods Of Testing: Suppose we have written a program to calculate and print the roots of a quadratic equation, having accepted three numbers A, B and C from the keyboard.

We have run the program a few times during the debugging phase and obtained satisfactory results. We now want to try and assure ourselves that our program can handle any quadratic equation that is presented to it in the form of its three coefficients.

What we must aim for is to feed the program with enough widely varying test data to be reasonably sure that the correct results are produced in all cases. At the same time, we must draw the line somewhere and say 'I am confident of the program's reliability' without going to the extent of trying every possible combination of the three values.

Obvious first candidates for testing are large and small numbers; for example, if A, B and C are zero, what does the program do? If A, B and C are very large numbers, do we exceed the limits of our computers arithmetical capability?

Controls: Related to the concept of program testing is the topic of 'controls'. These are checks on the data at various points in the program. If they are good enough, we can see where errors are occurring and thus concentrate on that part of the program which is causing the problem in order to decide what needs to be corrected. One example might be to place temporary PRINT statements in our program code to show the contents of certain variables at regular points. If the final result is wrong, we can trace it back to the place in the program where the error first occurred.

It is easy to over-control just as it is easy to overtest. The art lies in knowing when to stop.

Summary: We can never assume that a program will always yield correct results just because we wrote it, or because it worked once, or because we've been using it for 6 months without any obvious signs of trouble. Experience teaches us to be suspicious of our own programs and to test our logic wherever possible.

Summary Guidelines For Testing:

1. Provide known, logical data
2. State the expected results
3. Know when to stop
4. If a result appears wrong, INVESTIGATE. Don't fob it off as 'probably a mistake in the test data' just because the rest of the testing has worked.

A good test should give the programmer positive indication of where errors might occur, so that all possible loopholes may be plugged.

--o0o--

GAMES UPGRADES

By R.D.Watts.

The following are conversions from BASIC2.0 to BASIC4.0:

INVADERS

POKE1298,85 POKE1303,228

POKE2642,85 POKE2643,228

POKE6566,85 POKE6567,228

CAR RACE

POKE 874,29 POKE 875,229

POKE 924,85 POKE 925,228

POKE5705,85 POKE5710,228

POKE5718,85 POKE5719,228

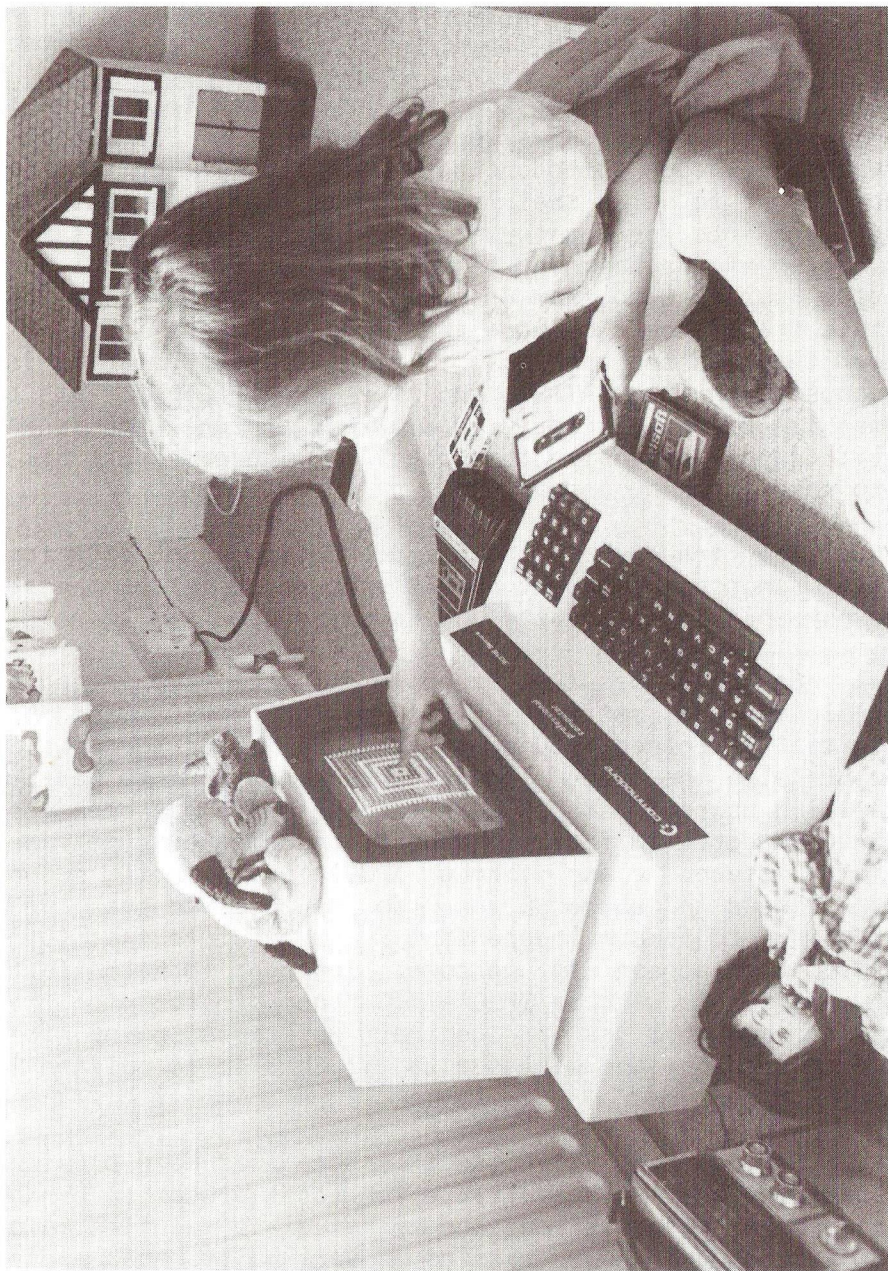
POKE5749,228

The programs must be saved using the monitor, from \$033A to \$2000. Note if any disk commands are executed, you will lose about the first 30 bytes of your program, so save the program as soon as the POKES are entered.

--o0o--



1. Yes, I know it's an '*appl*', but watch out for PET/CBM versions of these software derivatives of 'Visicalc'.



2. We all have our toys

VIC MATTERS

By Mike Todd.

It's not so long ago that a record in the TOP-40 was echoing the question we were all asking - "Is VIC There?". Well, although Commodore have been slow in getting the VIC up front to match their advertising, at long last VICs are flowing into the country at a reasonable rate. This is being written at the beginning of October when just over a thousand VICs have arrived from Germany, and by the time you read this there should be more than 7000 VIC users in the UK! My own thoughts are that this is just as well, bearing in mind competition from ATARI and the BBC PROTON. This latter machine offers extremely good value for money and is only a little more expensive than the VIC - however, it has still to make a formal appearance.

The trade has had pre-production model VICs for some time, and the onslaught of VICs should be met by a wealth of experience in both soft- and hard-ware.

So where does IPUG stand in all this? We intend to co-ordinate the setting up of regional VIC user groups, initially centred around existing PET user groups. There will also be some financial compensation to assist in setting up regional VIC users groups which has been promised by Commodore. In fact, Commodore are being extremely helpful in this venture and (through IPUG) are offering support in all areas - although you can be assured that we will continue to remain independent - a feature that Commodore themselves appear to welcome. In order to help start this movement, I will be IPUG's VIC "co-ordinator" for the time being and act as a central reference point for VIC correspondence and will see to it that it is pointed in the right direction.

So to the VIC itself - one has been provided by Commodore and it is available for local groups to borrow. I have been trying it out and my immediate reaction is one of a well thought out machine, well built and apparently reliable but with two major failings. The first and probably most documented is the 22-character wide screen. There

appears no logical reason for this decision and it results in difficult to read listings and can cause problems in trying to lay out screen displays. I have no doubt that for games, this restriction isn't too bad, but for anything else I think it is a disaster and will certainly result in the VIC not taking the market lead as it ought. The other failing is the lack of RAM. Only 3.5k is available to start with although a 40 pound, 3k add on RAM pack will make up the difference. It is a shame that this RAM isn't included (or at least the facility provided for on-board expansion instead of requiring an add-on pack).

Anyway, let's have a look inside. The VIC is supplied with 16k of ROM which includes the RS232, serial interface and cassette driving software as well as screen, colour and keyboard routines. BASIC is the old BASIC2 (which will consequently have the old garbage collection problem) with one or two refinements - for instance, "null" INPUT (ie carriage return with no input provided) no longer aborts the program, instead execution continues with the next statement and the input variable is not affected. The SYS command is slightly different (and more useful), allowing the A,X,Y and status registers to be passed to and from machine code programs. Also, many hooks have been provided to allow interception of some of the interpreter's routines (e.g. the error handler has a hook which could be used as an error trap thus allowing some sort of "ON ERROR" command).

Many important operating system routines are available via a jump table at the end of the ROM. These will (should ?) remain in the same place whenever new ROMs are released and will make software portability very much easier. This concept has been used for many years on some operating systems (such as CP/M) and is referred to as a KERNEL operating system (note the spelling!). In addition, many of these routines also have vectors in RAM which means that programmers can intercept them to incorporate their own routines.

I don't want to go into details of the VIC chip at this stage (it is far too complex to explain in a short article) - but it provides for some very versatile screen handling.

For instance, you can move the "window" around on the screen, change its width and height as well as having the screen, colour and character generator move to where you want them. Some POKEs are listed at the end to do some of these things - there will no doubt be more in the future!

RAM in the VIC is a curious affair - an unexpanded VIC has RAM up to \$1FFF with a gap between \$0400 and \$0FFF. The 3k expansion module fills this gap. Further RAM extension is added from \$2000 at which time the operating system re-orientes itself and moves the screen RAM from \$1E00 to \$1000. There is an additional bank of screen RAM at \$9400 to hold the colour codes for each character position. It is intriguing to note that although this RAM is 4- and not 8-bits wide, it has a capacity of 1k - of which only half is used. A 40-column screen requires 1k ! Interesting....

Turning to add-on units - the following items are currently advertised: a lightpen at 25 pounds (all prices include VAT), joysticks at 13 pounds each. Fully implemented RS232 interface for 49 pounds, with a low cost version at 17.25 pounds. RAM expansion varies from 3k @ 39 pounds, 27k @ 184 pounds (with cheaper NMOS versions available soon) and 19k (including battery backup) @ 198 pounds. There's even a VIC toolkit at 25 pounds, and a ROM switch board to allow switching between different ROM sets (for games etc) at 39 pounds. If you want to use several add-ons, you will need a mother-board type expansion unit which should be available fairly soon at around 100 pounds. There should also be a standard IEEE interface in the future which will allow existing PET peripherals to be used. Hopefully, next time there should be news of the printer and the disk drive.

If you want to buy a VIC it will cost you 190 pounds but PETALECT [Now Datalect - Ed] offered IPUG members a discount which could be quite significant! For details, get in touch with David Annals, IPUG's discount co-ordinator.

Any comments, programs, tips etc can be sent to me - I will also try to handle any queries regarding the VIC or setting up local groups. The VIC promises to be a most exciting addition to the Commodore range and, although IPUG

are looking after it at the moment, there may well be a time when there will be a need for a separate national voice for VIC users - a voice that already exists through IPUG and one which Commodore are prepared to listen to.

Finally, to some of the POKEs I promised - firstly, the VIC has three keys which you may want to inhibit - there is the STOP key, the character "flip" keys (CBM + shift) and the RESTORE keys. These can all be disabled as follows:

SCREEN FLIP:

To DISABLE - PRINT CHR\$(8) To ENABLE - PRINT CHR\$(9)

STOP key: (also stops the TI/TI\$ clock)

To DISABLE - POKE 788,194 To ENABLE - POKE 788,171

RESTORE key:

To DISABLE - POKE 37150,2 To ENABLE - POKE 37150,130

VIC chip registers can be altered as follows - remember that the RESTORE + RUN keys when pressed together will set the display back to normal) - if you make the screen too big you will run out of RAM and get some weird effects at the end of screen. This can be cured, and won't occur on expanded VICs:

POKE 36864,X Moves the screen window
(where X<128) horizontally (X starts at 12)

POKE 36865,Y Moves the screen window vertically
 (Y starts at 38)

POKE 36866, W OR 128 Sets the number of characters on a line
 (if the right hand edge runs off screen
 you will get some corruption - try
 moving the window to the left)

POKE 36867,H*20R128 Sets the number of character lines on
 the screen.

SCREEN INPUT

PET programmers will be well used to the INPUT# and PRINT# commands, and some will have used the screen as an output file (using device number 3), or the keyboard (device number 0). This can be a useful debugging aid when keyboard or screen can be temporarily substituted for disk, printer or cassette during debugging in order to simulate communication with these devices, it only being necessary to change one parameter in the appropriate OPEN command.

However, although the keyboard can only be accessed as an input file, the screen can be accessed as output OR input ! The former more common, but the latter can have its uses.

Assuming that you have executed OPEN3,3 then INPUT#3,A\$ will set A\$ to whatever is on the current line on the screen - WITHOUT pressing RETURN ! This allows lines to be input under programmer control without the user getting involved.

Unfortunately there are problems using this technique: e.g. the line is read subject to normal INPUT rules - i.e. a comma separates input items so that if one variable is specified then the line will be read up to the first delimiter (e.g. comma). Also only the first 39 characters (or 79 on long lines) are read and the returned string has trailing spaces included up to the 39/79th position with all leading spaces being ignored.

To position the cursor on the required line, use a <home> followed by the required number of <csrdn> otherwise problems arise when the INPUT#3 encounters a wrap-around line.

Having determined the problems, we must now try and find a use for this facility. One possible (albeit rather trivial) application is the conversion of screen characters in their POKE form (i.e. with "A"=1 etc.) back to ASCII (with "A"=65). This can be simply achieved by the following:

```
PRINT "<clr>*<home>";
POKE32769,CH : INPUT#3,A$ : CH$=MID$(A$,2,1)
This routine puts the character value into screen location
32769 (the second character on the screen), reads the top
line and extracts the required character from the string.
The reason for putting a "dummy" character at the start of
the line is to force INPUT#3 to read the character even if
it is a space - otherwise, since leading spaces are
suppressed, spaces wouldn't be converted.
```

A simple screen dump program is also possible which reads the screen into an array or to a printer. As an example, the following program allows the user to write on the screen and then by hitting RETURN the entire screen contents are transferred into the array SC\$. A shifted space at the start of each line ensures that leading spaces are read. If the user overtypes this with an ordinary space then leading spaces on the line will be suppressed! Also remember that the last character on each line is ignored.

```
10 DIM SC$(25) :REM set up array
20 OPEN2,0 : OPEN3,3 :REM open kbd & screen files
30 : REM *** set C$ for line positioning
35 C$="<home>" : FORI=1 TO 25 : C$=C$+"<csrdsn>" : NEXT I
40 PRINT "<clr>"; :REM clear screen
50 : REM *** put SHIFTED spaces at start of each line
55 FORI=1 TO 24 : PRINT" " : NEXT I : PRINT" ";
60 INPUT#2,A$ :REM flash cursor & allow typing
70 FORI=1 TO 25 :REM continues here after RETURN
72 :PRINT LEFT$(C$,I); :REM position cursor onto line
74 :INPUT#3,A$ :REM read line
76 :SC$(I)=LEFT$(A$,39) :REM put line into array
78 NEXTI
```

Line 76 transfers the first 39 characters to avoid the transfer of entire wrap around lines since the continuation line will be read anyway! However, this results in some odd things happening if the initial shifted space of a wrap around line is overtyped with an ordinary space. These are just two simple examples and I'm sure that someone will find a more useful application in the future - if you do I'd be pleased to hear about it.

SCREEN OUTPUT ROUTINES

By Mike Todd

If you are trying to print a character onto the screen from a machine code program, you have a choice of four routines with which to do it! Since you will see all four used at different times and by different programs it is not surprising that there is an element of confusion!

The four routines are (in BASIC2) at \$FFD2, \$F232, \$E3EA and \$E34C. The first two are in fact identical except that the \$FFD2 routine is always in the same place regardless of which BASIC you are using.

Let's first look at the \$E34C routine. This takes the ASCII character in the accumulator and prints it on the screen, handling line pointers etc. as it does so. However, it cannot deal with the ASCII control characters.

These are looked after by a routine at \$E3EA which, as well as handling normal printing character (which it routes to the \$E34C routine), performs the necessary routines to handle the control characters.

These routines are exclusively for handling output to the screen - the \$FFD2/\$F232 routine directs the ASCII character to the most recent device set up by the routine at \$FFC9. This could be any IEEE device, the cassette unit or the screen. If it is the screen (which it will be if you call \$FFD2/\$F232 without setting up an external device) it simply executes the \$E3EA routine!

Thus you will see that all the routines are related to each other and I hope that you will now be able to choose the routine most appropriate to your purpose.

--o0o--

THE MEMBERSHIP

Estimates put the I.P.U.G. membership at about 40% business, 40% academic and 20% hobby/enthusiasts.

--o0o--

IPUG PROJECTS

Data will be available in the near future for the following projects:

The Interface Interface.

Here the objective is to lessen the vulnerability of the user port and cassette port edge connectors to damage by frequent plug interchange. The 'II' is left permanently connected to both ports. Its functions include:

- a) duplicate user port and cassette port edge connectors.
- b) Canon sockets which duplicate a) above.
- c) 5-volt output which may be switched from PET internal to battery external.
- d) a Read/Write monitor.
- e) a CB2 sound amplifier.

Video Box - supplying a composite video signal to an external monitor.

Sound box - to provide, by switch selection, either CB2 sound or digital-to-analogue converted signals.

Mains switching box. There will be three variations:

- i) relay operated.
- ii) Triac operated.
- iii) opto-diode operated.

UHF modulator allowing use of your domestic TV as a monitor.

Light pen.

Disk Error Bleeper.

To enrol, please send one stamped S.A.E. to Fred Offler and he will contact you as soon as the data sheets become available. No service charge has yet been agreed, but it is thought that it will be a once only payment of £ 1.00 for I.P.U.G. (£ 10.00 for non-members) - but don't send any money until requested. Enquiries or orders should be sent to Fred Offler, 37, Brooklyn Road, South Norwood, London, SE25 4NH. Cheques and Postal Orders should be crossed and endorsed "A/C Payee".

REVIEW

HR-40 HIGH-RESOLUTION GRAPHICS BOARD Supersoft £ 149 +VAT

Screens of 40-column PETs are an array of 40x25 character cells - each of which is 8x8 dots. The pattern of dots for each character is stored in ROM and so is not readily accessible. However, with a screen of 320x200 individual dots it should be possible to access high resolution graphics if someone could develop the necessary hardware.

Micro Technology Unlimited in the USA did just that a long time ago and produced a high resolution board for PETs allowing full access to the 320x200 dot screen at a cost of around 300 pounds. Now, Britain has stolen the lead on the USA - SUPERSOFT has released a 300x200 high resolution board for 40-column machines (an 80-column version may be released later) at a price of only 150 pounds.

The board is smaller than the MTU board (about 8" x 3") and has 2 ROMs, 8k RAM and a few other ICs. There are also 2 empty ROM sockets to hold any existing ROMs that you may have in the UD3 or UD4 slots. The reason being that the 8k of RAM is addressed at \$9000-\$AFFF which could conflict with existing ROMs - a simple POKE switches in your own ROMs provided that you've put them on the board. In fact you never lose anything by installing the board - a very important advantage.

Although I was totally exhausted when I installed the board (having been up for 24 hours, worked a night shift and spent half a day at the PCW show !), I reckon that the 7 minutes it took me to get it working is not bad going. There is no soldering required, all connections are made using 3 small clips which clip onto pins of existing ICs, a few connectors which push onto pins on the expansion port, a small (dead!) IC which rides "piggy-back" on one of the on-board ICs, a flying lead which replaces the \$E000 ROM and the board itself plugs in to replace the character generator ROM. All this is clearly explained in the instructions.

The on-board software is mapped in the empty space after the I/O chips (\$E900-\$EFFF) - therefore there is no competition for other ROMs you may have. Simple POKES allow the selection of hi-res screen, normal PET screen, both, or neither. You can also use this RAM whenever you don't want it as hi-res. On-board software allows single points to be set or cleared, lines to be drawn or cleared, areas to be filled or cleared as well as clearing or inverting the screen - these commands are not "wedged" into the CHRGET routine, but use a common SYS command followed by the hi-res commands and parameters. For instance to set a line from top left (x=0,y=0) to the middle of the screen (x=160,y=100) you use:

```
SYS HR,SL,0,0,160,100 (where HR is set to 59650)
```

Later versions of the board have had the line drawing algorithm changed resulting in faster line plotting. Text can be plotted if you have an addressable character generator available. This can be done by putting the character generator (removed when installing the board) into the \$B000 slot. This could be moved into RAM and allow you to create your own character set - it only requires another POKE to tell the board where the table is. This does NOT affect normal PET characters since that character generator is as before (except that it is now one of the ROMs on the hi-res board).

Uses of such a board are numerous - one which might appeal to business users is the ability to draw forms (including company logos) on the hi-res screen and (by mixing the two screens) get user input from the normal screen. One drawback has been brought to my attention, and that is that anyone with a board already sitting on the main board (e.g. NETKIT users) may experience difficulty in gaining access to the necessary connection points.

I enjoyed using the board and with the prospect of a range of utility software being developed it promises to be one of the best value add-ons for the PET that I have seen.

M. R. T.

SHOP WINDOW

Furniture specifically designed to house computer, printer and storage devices is available from a number of sources. A large PET desk is made by Amplicon Microsystems Ltd., 143, Ditchling Road, Brighton, East Sussex, BN1 6JA. Contact Howard Pilgrim, tel: (0273) 562163. A range of office equipment for microcomputers is marketed by Westra Office Equipment, tel: 01-843 1122. Note, however, that a complete system console is available from Fred Offler, either ready-made, or as a constructional product. Contact Fred for pricing details (ready-made costs £100).

A copyholder dedicated for use on the PET enables easier viewing of material being typed. Known as the "Easiview Copyholder" it has two A4 size panels to hold the text. It is fully adjustable and costs £24.00 + VAT from Concordia Automation Components Ltd., 6, Central Road, Worcester Park, Surrey, KT4 8HZ. Tel: 01-337 4541.

Making strange bed-partners, an EPROM programmer is sold by the above company, in addition to a monitor utility known as MOUSY. The utility is available on disk or cassette at £168. The programmer is also priced at £168.

A PAL colour encoder is available which will accept RGB and sync signals and convert them to a composite PAL colour video signal. This is then modulated on to a UHF carrier at channel 36. The circuit includes a sound input and is useable for Prestel/Teletext systems. Further details and pricing from RGB/PAL Encoder Dept., Acorn Computers Ltd., 4A, Market Hill, Cambridge, CB2 3NJ.

A complete Winchester disk subsystem which can be configured to any operating system that accommodates IEEE-488 protocol is supplied by Sintrom of Reading, Berks. Capacity is 5M-byte and includes 22-bit error detection and 11-bit correction. The 5.25" disk system is known as MSC9800. Price £2950. Tel: (0734) 85464.

A machine-code 'word processor' costing under £40, suitable for old and upgrade ROMs, cassette or disk (including 80-column version program) is advertised by Simple Software Ltd., 15, Havelock Road, Brighton, Sussex, BN1 6GL. Tel: (0273) 504879.

Finally, special services are offered for insurance of microcomputer systems by R.J.Deer Insurance Services, 14, York Place, Clifton, Bristol, BS8 1AH. Tel: (0272) 38117.

R.D.G.

--o0o--

CBM COLUMN

The latest CBM price list (September) indicates a rationalising of the computer range. Models 4016, 4032 being both 40-column models with the 8032 appearance (12" screen), oust the 3000 series, but BASIC 2.0 is available as an ancillary. The small system printer is the 4022 which ambles along at 65 characters/second.

The 8010 modem makes its debut at £225 (v.a.t. extra) as do a number of system configurations (complete with cables!), along with the 2031 single disk drive (£ 395), the 8422 22M-byte Winchester at £3495 (VAT extra) and 8096 96K computer at £1095.

The VIC has taken off like a rocket and looks set to make up for the sales losses resulting from its delayed quantity production (see VIC column) even though the Texas Instruments have got in first with the TI 99/4 in Boots, Argus and Rumbelows stores. (You want a ZX what ?? silly).

The PET accounts for over two-thirds of micro-computing systems in use by local authority departments according to a recent report. The micro is a result of red-tape and committee wrangling over centralised systems causing local management to 'go it alone'.

The Commodore Information Services address is P.O.Box 109, Baker Street, High Wycombe, Bucks. Tel: Slough 79292.

R.D.G.

--o0o--

REVIEW

ARROW ROM

Supersoft £ 30

When I managed to get to the front of the crowd round the Supersoft stand at the PET Show I discovered that they had a new program in EPROM called ARROW which, among other facilities, permits cassette operation at 3600 baud. A summary of its features is so similar to that of Carl Moser's PET RABBIT that I thought ARROW must be the same program. However, Milton Bathurst who wrote ARROW, had come from Belgium to be on the Supersoft stand. After his demonstration of its capabilities I was in no doubt that ARROW is a new and excellent program. The commands, like the program name, are self-evident and simple:

← S"FILE NAME" or, for machine code:

← S"FILE NAME",hex start addr,hex end addr

← L file name optional

← V file name optional

← A Append - in the same manner as the TOOLKIT command A Load or verify error is shown by ?? The points to note, which are mentioned in the instructions, are:

(a) Only the first cassette can be used with ARROW

(b) The file name, which is optional, must not exceed 15 characters

(c) The header tone is of only 3 seconds duration but many tapes have a 5 second leader. The other commands are:

← R enables repeat function (indicated by a high-speed flashing cursor)

← O disables repeating (needed if CBM LOAD or SAVE is used)

← F flips character set (easier than POKE 59468,14)

← Pd (d=1 to 9) times fast forward tape advance for up to 9 programs of 32K

← H hexadecimal calculator and hex/dec conversion

← G graph plotting with quarter blocks. The 2K program is available in any one of 6 locations in BASIC 2.0 (new or upgrade ROM) and any of four locations in BASIC 4.0 Apart from the advantage of the high speed save and load facility, ARROW will also:

(a) Save screen memory and load it back to the screen

(b) Save ROM and load it back into RAM fitted in the same location (e.g. Greenwich Instruments 'Instant ROM' or in my case, a home-made version with two 6116s). These areas of

memory cannot be saved directly in the standard PET tape format although they, and the first cassette buffer can be saved to CBM disk. The first cassette buffer can be saved with ARROW but it is corrupted with the file name which is now stored from \$A2A0. So ARROW cannot be used for making back-up copies of such programs as 3D STARTREK and COSMIADS - You will have to continue with the rather lengthy method needed with the standard PET tape format. I have been using ARROW regularly since the PET Show and have found it entirely reliable even with programs comprising a mixture of BASIC and machine code (which RABBIT is unable to save). After fairly extensive use I have only a few minor criticisms: (a) If, when one starts searching for a program in mid-tape, the header is over the playback head then the program name is frequently displayed as a mess of graphic characters. Neither this, nor a subsequent program is FOUND. One has to start again. (b) The indication of a verify or load error is two question marks but OK still appears on the next line. (c) I find the ←H function a trifle clumsy but I have not used it seriously as I have got into the habit of using the !N function in MIKRO. To sum up, if you use a cassette unit numbered above about 6000 (this is a guess - see my note in IPUG Vol 3 No 1, Jan 81, p30) then the purchase of an ARROW ROM will convert a high proportion of the waiting time into computing time. If you bought ARROW at the Show or soon after then ask Peter Calver for a copy of the current instruction leaflet. This later version provides a better description of the use of the graph plotting functions - and mentions the 15 character file name restriction.

David W. Jowett.

--o0o--

8000-SERIES PRINTER NOTE

The 8026 (= Olympia ES-100) is a simple non-intelligent printer with keyboard. The 8027 has no keyboard, but the space saved by not having any decode logic is used to provide an intelligent interface to the computer which handles many facilities and makes the 8027 comparable to Qumes and Diablos, except for speed and price.

Information from Pete Dowson.

--o0o--

PROGRAM SNIPPETS

The following routines are offered as ideas for incorporation into ones own programs.

```

1000 REM PRIME NUMBER BY GEOFF LAWRENCE
1020 I=0:T=2:INPUT"TEST NUMBER-ZERO TO STOP";N
1040 IFN=0THENEND
1060 IFN<4THEN1140
1080 J=INT(N/T):K=J*T:IFN=KTHEN1160
1100 I=I+1:L=T*T:IFL>NTHEN1140
1120 T=(I*2)+1:GOTO1080
1140 PRINT"<clr><rvs>";N;"<off>IS A PRIME.<3dn>":GOTO1020
1160 PRINT"<clr><rvs>";N;"<off>IS NOT A PRIME."
1180 PRINT"<rvs>";T;"<off> IS THE SMALLEST FACTOR.<3dn>":
GOTO1020

```

No prompt is given, just enter the binary value:

```

100 REM BINARY TO DECIMAL CONVERSION BY GEOFF LAWRENCE
120 R$="00000000":V=0:INPUT"<clr>BINARY";A$
130 L=LEN(A$):IFL<8THENA$=LEFT$(R$,8-L)+A$:L=8
140 FORW=LTO1STEP-1
150 IFMID$(A$,L+1-W,1)="1"THENV=V+2^(W-1)
160 NEXT:PRINT"<8sp>=";V
170 GETR$:IFR$=""THEN170
180 GOTO120

```

Plot of the frequency of occurrence of the numbers 1-9

```

1000 REM NUMBER PLOT BY GEOFF LAWRENCE
1020 DIMD(9):LS=33574:PRINT"<clr>"
1040 GETX:IFX=0 THEN 1040
1060 D(X)=D(X)+1
1080 LC=INT(D(X)-8*INT(D(X)/8))
1100 L=INT((D(X)-1)/8)
1120 IFLC=0THENDC=160
1140 IFLC=1THENDC=100
1160 IFLC=2THENDC=111
1180 IFLC=3THENDC=121
1200 IFLC=4THENDC= 98
1220 IFLC=5THENDC=248
1240 IFLC=6THENDC=247
1260 IFLC=7THENDC=227
1280 POKE(LS-(40*L)+X),DC:GOTO1040

```

To check for end of screen, put $I=LS-(40*L)+X$:
 IFI<32808THEN END (or wherever).



Printed, published and distributed by Carlton Press, 10 Bernard Road, Gorleston,
Gt. Yarmouth, Norfolk NR31 6EG, England, UK. Tel: Gt. Yarmouth (STD 0493) 61331