

ICPTC

VOLUME 6
NUMBER 6
NOVEMBER
1984

INDEPENDENT COMMODORE
PRODUCTS USERS GROUP

HONORARY NATIONAL OFFICIALS

CHAIRMAN

Mick Ryan
164 Chesterfield Drive
Riverhead
Sevenoaks, Kent TN13 2EH
Tel: (0732) 453530

TREASURER

Jennifer Goldsmith

VICE-CHAIRMAN

Mike Todd
27 Nursery Gardens
Lodgefield
Welwyn Garden City
Herts AL7 1SF

MEMBERSHIP SECRETARY

Jack Cohen
30 Brancaster Rd
Newbury Park
Ilford, Essex IG2 7BP
Tel: 01-597 1229

EDITOR

Hugh de Glanville
45 Woodland Grove
Weybridge
Surrey KT13 9EQ

ASSISTANT EDITOR

Henry Velleman
52 Ventnor Drive
Totteridge N20 8BS

ASSISTANT EDITOR

Tom Cranstoun
109 Dalmeny Rd
Addiscombe
Croydon

DISCOUNTS OFFICER

John Bickerstaff
45 Brookcroft
Linton Glade
Croydon CR0 9NA
Tel: 01-651 5436

DISCOUNTS ASSISTANT

David Murray
8 Enticott Close
Whitstable
Kent CT5 3ET

DISCOUNTS ASSISTANT

(stockholder - disks, &c)
Brian Wise
17 Knighton Close
South Croydon
Surrey CR2 6DP

DISCOUNTS ASST. VIC-20

Derek Hoare
10 Snell Hatch
West Green
Crawley
Sussex RH11 7JB

TECHNICAL QUERIES VIC/64

Mike Todd
(see above)

NETWORKS/COMAL

Brian Grainger
73 Minehead Way
Stevenage
Herts SG51 2HZ
(Systel 438727925)

SOFTWARE ORGANISER

Alfred Minter
2 Whitefriars Way
Sandwich
Kent CT13 9AD
Tel: (0304) 617209

(Software Librarians - see inside back cover)



INDEPENDENT COMMODORE PRODUCTS USERS GROUP newsletter

Vol 6 No 6 NOVEMBER 1984

- 382 Editor's Notebook
- 383 Protection Copyright in Software
- 385 Inside Commodore DOS - Review
- 387 Deciphering Printer Manuals
- 390 Network News
- 394 Plus 4/C 16 Preliminary Memory Map
- 405 Superbase Corner
- 418 Christmas is Coming . . .
- 420 The AEA Micropatch - Review
- 426 Tee-Switch for Printers
- 428 Teach Yourself Programming 6 - Answers
- 429 The Great ICPUG Competition
- 432 Disk File - Sector 11
- 438 700 Series - Some Notes
- 441 Comal Corner
- 446 64 Library
- 448 An Improved Format Subroutine
- 452 C 64/Spectrum Emulator
- 453 National Committee for 1985
- 454 Regional Group Contacts
- 460 Odds & Ends
- 461 1541 Express - Review Update
- 462 Discount Corner
- 463 Readers Write . . .
- 464 Members' Sales and Wants

Europe's first independent magazine for CBM users

EDITOR'S NOTEBOOK

As everyone knows, we have been falling back a little on schedule of late, but we - that is I and my assistant editors and the Committee - decided to make a major effort to catch up and to bring out the November issue by the end of November, surprising you all.

What actually happened was that I myself was rather surprised, after 40 years of avoiding any major debacle on two wheels, to skid on a patch of oil on a roundabout on my trusty motorcycle on November 1st and do absolutely no good to one of my legs. This involved some time in hospital and rather restricted mobility thereafter, so the best-laid plans of editors received a setback.

However, everyone, and particularly Henry Velleman, has rallied round and we have produced a 'Christmas' issue for you under rather difficult conditions. If you find any ghastly errors, or feel it is not quite up to our usual standards, please bear with us; we just have not been able to check it as thoroughly as we would have wished. However, Mike Todd's amusing Great !CPUG Competition on p429 should make up for a lot of faults.

One vital thing that I have omitted to do, to the increasing annoyance of the Membership Secretary for at least two issues now, is to announce that copies of the valuable May 1983 issue are now available again from him (Jack Cohen). This was the issue that contained Sector 6 of the Disk File, and when it was out of print many people were disappointed. Now it is in stock again and those who want to complete their collection of, particularly, the Disk File, may apply to Jack.

I am not going to attempt to fill my page this month. I shall sign off with very best wishes for Christmas to all our readers, hopes for good computing and everything else for us all in 1985, and thanks to all those who have been so helpful with the Newsletter throughout the year. May any crashes you have be restricted to your computers.

HdeG

PROTECTION COPYRIGHT IN SOFTWARE

by Alistair Kelman

(This article originally appeared in ICPUG SE Newsletter October (No 22), whither it had come, with the author's permission, from the ICPUG pages of COMPUNET. We are indebted to everyone involved for permission to reproduce it here.)

In Britain and in most of Europe it is not necessary for an author to take particular formal steps to reserve copyright in any work he creates. A copyright arises the first time the work is written down and it seems probable that English courts will consider that writing to disk or tape will be considered 'writing' for the purposes of the law of copyright. Although there have been no decided cases in the United Kingdom on the subject computer software is considered to be a form of literary work for the purposes of the law of copyright and therefore when it is written down a copyright arises in it lasting for the life of the author plus fifty years.

The only right granted by copyright is a right to stop the work being copied. This means that if a person independently created an identical work to that of an author it would not infringe copyright in the author's work since it would not have been created by copying. However if the person has had access to the author's work and his work was written later than that of the author a court would look at the works and, if they were similar, would be likely to assume that copying had taken place. It is therefore important for evidential purposes for a software author to keep a note of the history of the development of his program and a record of its state at key stages. A suggested method of so doing is as follows:

- Keep a work diary into which notes giving dates and times when work was done are inserted.
- When the work is finished or when it has reached a critical state make a listing of the program, place it in an envelope, address the envelope to yourself; seal it very firmly with sealing wax and do one of two things:
 - EITHER send it by Registered Post to yourself and upon receipt keep the mailing slip and the unopened packet in a safe place until they are

needed in litigation

- OR get your bank to keep the envelope in their vault until it is needed in litigation.

By this method it will always be possible to prove that a program existed in a particular form on a particular day.

Although it is not needed for protection in the United Kingdom it is important to place a copyright notice in all copyright works. This is to reserve copyright in countries such as the United States which require such a notice. The notice is made up of three parts, all of which must be present. First should be the word "COPYRIGHT" or the abbreviation "COPR" or the letter "C" in a circle. Second should be the name of the copyright owner which for home users will be the name of the author. Third should be the year in which the program was written. These three parts make up the copyright notice and should be inserted in a REMark statement and also if possible in a form in which they will appear on screen at the start of the program and hidden in the program so that they will appear on screen when an unusual combination of keys is pressed.

Before a program is marketed if you are concerned about it being pirated in the United States it is important to register it with the United States Copyright Office which is part of the Library of Congress in Washington D.C. The US Copyright Office will send on request copies of Form TX which is the form needed to register computer software under American law. By sending a listing of the program together with a completed Form TX and a cheque for \$10 the copyright owner can claim very substantial damages for infringement when it takes place and most important of all gets an immediate right to sue for infringement in the United States. In practise, since American lawyers are allowed to work on a contingent fee basis, a software author who had registered his program could, if he found infringement taking place in the United States, get an American attorney to fund and fight a copyright infringement action in the United States for a percentage of the possible winnings at no additional cost to himself.

--oOo--

INSIDE COMMODORE DOS—REVIEW

by Mike Todd

This book describes itself as "the complete guide to the 1541 disk operating system" and I think that sums it up.

It is the same size as Raeto West's book with over 500 pages, but its price is rather hefty at nearly £20.00. It is written in three distinct sections; the first is the main body of text of two hundred or so pages, followed by another two hundred of detailed ROM and RAM lists. The final pages are devoted to additional program listings not given in the main text.

The book starts by introducing the reader to exactly what a disk operating system is. This describes the command channel and the commands that can be sent to it, followed by a thorough investigation of disk formatting, the layout of the disk itself and how sectors are laid out.

Chapter 4 is the start of the meat of the book and describes in great detail how the directory is made up, the structure of all the different file types - and much more besides. Then the techniques of direct access programming are described, through all the memory and block read/write commands, and this is further developed by examining some uses to which these can be put. The main one being to access the job queue and use it to read and write sections of the disk.

I suppose it would be possible (somewhat cruelly perhaps) to call this book a pirate's handbook since chapter 7 starts to delve into the subject of DOS protection. This is the technique used by software houses to protect against illicit copying of their programs.

However, it makes an ideal route into the bowels of the more advanced disk techniques and chapter 7 describes the Commodore data encoding scheme based around a technique known as Group Code Recording. This is the encoding of a standard 8-bit byte into a 10-bit word before writing it to the disk surface.

Protection techniques are wide ranging, and this chapter describes many of the deliberately induced errors which are used to "mark" a protected disk and which will be checked for by the running software.

These errors range from simple data block checksum errors, through to completely destroying individual tracks and the book includes complete BASIC listings of programs to create these types of errors.

The programs are described in some detail and, if you manage to

understand the techniques used, then you will have a thorough grasp of exactly how the drive operates, although I did find irritating the constant reference to errors by their numbers rather than type.

Chapter 8 is called "Getting out of trouble" and covers many of the problems both beginners and advanced programmers will meet, from accidentally scratching a file or "NEWing" a disk, to unclosed files.

Until now, we have only had tantalising glimpses of the depths of the Commodore DOS and chapter 9 starts to describe the routines within the ROM set, how the DOS is organised, and how its major routines work.

It also covers the recording process itself. The diagrams and explanations are easy to read and describe the process in great detail.

The memory maps are the most thorough that you will find anywhere and are partly based on the original Commodore source code, with many additional comments by the authors. Every byte of RAM and ROM is documented in detail, complete with labels, and should really be read in conjunction with a disassembly of the ROM contents, something the book does not supply.

The book is unashamedly aimed at the advanced disk user and will suit those just beginning to delve into the 1541, up to those who thought they knew it like the back of their hand.

It is full of clearly listed example programs, many of which are extremely valuable disk utilities, others are more specifically aimed at producing the errors described - and they work! Any machine code required is included as DATA statements within these programs, although assembler source listings are included for clarity.

Perhaps not surprisingly, not all protection techniques are described, and there are a couple of minor errors, but this is a book I wholeheartedly recommend, and I'm sure it will acquire much the same status as the Raeto West's book did for the PET.

The two authors, Dick Immers and Gerry Neufeld, are both very knowledgeable disk users and they acknowledge ICPUG as a major source of inspiration and insight. Their experience with the 1541 is extremely well presented and easy to read.

It is published in the USA by DATAMOST and distributed in the UK by Prentice-Hall at something a little short of £20.00 and a disk containing all the programs should be available.

If you believe this book is likely to be beyond you, Gerry Neufeld is soon to release a beginners' guide to the disk drive.

DECIPHERING PRINTER MANUALS

I have never been a lover of Commodore printers and along with many other Commodore users I have looked to the Japanese to fulfil my printer needs. Commodore are renowned for the pathetic quality of their manuals and those of us who have bought Japanese printers may have wondered if there was an international plot to destroy the brain cells of the microcomputer user by publishing instruction manuals in the form of anagrams and other devious word games.

Part of my assistance to the Editor consists of reading newsletters and magazines from other groups on the lookout for articles which may be of interest to the general ICPUG membership. A recent perusal of the Home Office Amateur Computing Club magazine for August '84 showed that it isn't only Commodore owners who are afflicted by the international plot described above; even Her Majesty's employees are affected! A very enlightening article called " How to Drive a Printer From BASIC " by Ronald Hewlett attempted to unravel some of the mysteries of the average printer manual. It is not appropriate to reproduce this article in full as it is written for micro users in general rather than Commodore users specifically. However I would like to quote rather freely as I'm sure many readers will find it truly informative and of help to them in making fuller use of the potential of their printers.

Printer manuals have a strange habit of giving examples such as:-
LPRINT "ABCD";CHR\$(&HE);"EFGH";CHR\$(&HD).

What on earth does this garbage mean? Mr Hewlett will now explain.

LPRINT - this is equivalent on Commodore computers to 'PRINT#4,' where a channel to the printer has already been OPENed using OPEN4,4 . This command diverts PRINTing from the screen to the printer.

'ABCD' - will be printed as ABCD

CHR\$(&HE) - if you typed this in as it is you would get a SYNTAX ERROR because BASIC only allows a number between 0 and 255 as the argument to CHR\$. What it is intended to mean is that '&H' shows that the following letter is in hexadecimal notation ie that the 'E' is actually Hex '0E' which is decimal 14. In standard Epson code (which virtually all current printers obey) CHR\$(14) commands the printer to print following

characters in double width.

"EFGH" - prints EFGH as double width characters.

CHR\$(&HD) - as explained above this is equivalent to Hex '0D' or decimal 13 and simply means Carriage Return. This forces the printer to actually printout the ABCDEFGH and then move the paper up one line and reset the print head to the lefthand edge of the paper ready to print further data. On Commodore computers it is not necessary to send CHR\$(13) at the end of a line as long as a semi-colon ';' is NOT used as the last character as it has been in the above example.

One mystery solved . . . onto the next:-

Many printer commands make use of ESCape codes to produce special effects. Time for another example . . .

ESC-(minus)+(n)D(for underline print mode set/reset) (n=0 or 1)

The explanation is as follows:-

ESC - this signifies an ESCape sequence and is represented by the ASCII code 27 decimal or Hex '1B'. To actually send ESC to the printer replace it by CHR\$(27).

-(minus) - this is attempting to show a minus sign '-' as the character to follow ESC. The minus sign is the equivalent of ASCII code 45 decimal or Hex '2D' and can be sent to the printer as CHR\$(45) or simply as "-". (ie a minus sign within double quote marks).

+(n)D etc - the 'n' within brackets is symbolising a number and the 'D' that it is decimal. The rest of the line is simply attempting to explain that 'n' should be replaced by '1' to turn underline on and by '0' to turn underline off!

In other words

```
PRINT#4,CHR$(27);CHR$(45);CHR$(1);"This will printout underlined"
```

will print all following text eg "This will . . . etc" underlined. By replacing the CHR\$(1) by CHR\$(0) the underlining is switched off.

The example programs given in the manuals are generally no help as they

often mix up several special features in the one example without explaining this or even making use of REMs to clarify what each line of the example is supposed to do. Armed with the information contained in this article you should now find that the list of control codes which are normally produced as an appendix in the manual become an essential quick reference guide to the intricacies of your printer.

Just to give one last piece of assistance the following are a few examples of some of the more useful control code sequences:-

```
PRINT#4,CHR$(15);"This prints condensed characters"  
PRINT#4,CHR$(18);"This cancels condensed printing"  
PRINT#4,CHR$(27);CHR$(69);"This gives emphasised printing and can be  
combined with other control sequences to produce, for example,  
emphasised double width characters"  
PRINT#4,CHR$(27);CHR$(70);"this turns off emphasised printing"
```

I hope this article will encourage you to re-read your printer manual in a new light and be able to make full use of all the clever features those Japanese gentlemen built-in. I'm sure they didn't really mean to confuse us all beyond belief!

HGV

--oOo--

TEACH YOUR GRANDMOTHER 8

Why write unnecessary code? Grandma recently showed me the following

```
10 IF AN$ = "Y" THEN OPEN1,4  
20 IF AN$ = "N" THEN OPEN1,3
```

In cases where you are testing for only one of two outcomes you don't need two "IF" tests. Assume the most likely result and only test for the alternative. The above example could be simplified to one line as follows:

```
10 CH=3: IF AN$ = "Y" THEN CH=4: OPEN1,CH
```

Grandpa

NETWORK NEWS

by Brian Grainger (Systel 438727925)

It is now the 20th November and have things changed since the last newsletter? Well you might not think so from the inactive ICPUG pages on Prestel but I can assure you that behind the scenes things are moving.

As you will see at the end of this piece a lot of you have joined in the fun on Prestel and even Compunet. Many thanks to all those who said how much they liked this piece in the newsletter.

As far as Prestel is concerned a lot has been going on. For a long time I have been unable to edit the ICPUG pages. It started when Prestel coordinated the computer related areas, i.e. Micronet, Clubspot and Viewfax. This, together with the fact that Clubspot became a main IP rather than a sub IP, meant that the ICPUG pages had to move. During the move no editing was allowed. Unfortunately, after the move Prestel decided to take an awful long time to issue Clubspot with the new IDs and passwords. Eventually they got them and were just about to send out the ICPUG ID when Prestel decided to change them all again! This was probably due to the fact that a hacker had managed to infiltrate Prestel and do naughty things like read the mailbox of HRH Prince Philip. That's what the folklore says anyway. This week I got a new password so I hope, by the time you are reading this, that ICPUG is back on Prestel. You may not be able to find us however!! The front page is currently located at 8108122 but a further move is expected to 8102132. You will always find us from the Clubspot index if we appear to have disappeared off the face of Prestel.

Last time I briefly mentioned the Timefame Letterbox, but by a mental aberration I called it Teleframe. I also mentioned that Timefame made a charge to upload pages which was also confusion on my part. Timefame is free and is great. If you want to find out what other Prestellers think about the latest news, or you want to put forward your own views, give Timefame a call. It can be found at page 81811 or index through from mailbox (P.7) by hitting 6,2,# in quick succession. Be prepared for your phone bills to go up though! I spend a LOT of time on this. I might mention that Chatline does not seem to be keeping the high standard of Letterbox and I have long since stopped looking at it.

Probably due to the fact that it is all automatic and nobody is sifting the messages for rubbish removal.

Micronet have finally produced a modem for the 64. It is made by OEL and costs £123.65 after an initial offer of £99.95. Micronet is beginning to look very professional now but they have gone and angered a lot of people, including ICPUG, by putting up Commodore 64 software in non CET format. For the non technofreaks amongst you CET was an unwritten standard for Telesoftware. It was used for Micronet PET programs as well as Telesoftware on Teletext. The beauty of it was, of course, that the software could be downloaded by any modem. I suppose Commodore started the problem by deciding to make Compunet in a way that was only accessible by the Commodore modem. What Micronet have done is to make their Commodore 64 software only available to those with Micronet modems! It is all very well these commercial wars but the loser is always us, the poor consumer. Each side tries to justify their position with, in the case of Micronet, incorrect information being displayed on the network. They suggested that the Commodore modem, and others, could not download untokenised software in CET format. If they had bothered to look at some of the PET software I mentioned in this column some issues previous they would have realised their error. Any way, despite the aggro, there are two points to make. ICPUG will not allow its 64 software to go up on Micronet until we can all download it, not just those with OEL modems. Secondly a program will be going up on Compunet soon which will allow the Commodore modem to download Micronet software!

The other big news about Prestel is that mailbox has gone national and, what is more, it is free. The folklore has it that Prestel deliberately floated rumours of mailbox charges to see what the reaction would be. The reaction was exothermic with the result that the service is without charge. I wonder if this had anything to do with Compunet suspending their charge of 10p a mailbox during November.

Many issues ago I mentioned there were some PET programs that not many people knew about on pages 70041200-70041204. Please note that these programs are slowly disappearing and they can no longer be guaranteed to work! I guess Micronet have got round to messing these pages up.

I got a very interesting little book recently called 'The Micronet Handbook'. It is written by Barbara Hickford and is part of the

Personal Computer World series of books. So as to avoid favouritism I better mention that I got the book courtesy Commodore User! It is a very easy to read introduction to Micronet, Prestel in general, and Teletext. It describes each service, the costs involved, indexes to pages on each service and a glossary of the technical terms. I found it quite useful and I now know how to send a multi-page mailbox to somebody without forgetting what I said on a previous page! Unfortunately I haven't seen the book in the shops, but if you do have a look inside.

Having talked about Prestel at length I better say something about Compunet. The service has now started and some of you are even using it. I have had one comment that perhaps I was a little harsh on the service in the last newsletter but as I told the correspondent, until I can see Compunet again I will not alter my opinion. The general frustration of most users is that it is slower than Prestel. ICPUG have been negotiating with Compunet as to what service we will provide to ICPUG and Compunet members. At the time of writing we have agreed to Compunet/Commodore proposals subject to only a maximum of 30 per cent of ICPUG information going on general view to any Compunet user. We have not yet had the response from Compunet but we do not anticipate problems and expect to be setting up the database very soon. Software that has been provided free on Compunet has been the User to User software (U2U) as well as the viewdata software to access Prestel. A price of 5 pounds was fixed for the latter but, at the time of writing, has not yet been charged. The U2U software allows owners with 64s and Commodore modems to transfer disk files and messages directly over the phone. It does not work with cassette files but apart from that is very good. A problem was found with the initial version in that it did not work over a range greater than 100 miles. The version available by the time you are reading this should have overcome this problem. It did not seem to work with my preproduction 64 either, but not many of you are likely do have one of those. As indicated above the viewdata software is to be updated to allow downloading of Micronet programs and possibly some offline mailbox facility.

Well, that is it for this time. I shall finish off with the list of contacts made this period. It is getting longer each time. Welcome to Paul Crossley, John Moulton, Michael Arnold, Robin Tomlinson, Robin Leaver, John Hunter, Richard Wilson (a VIC man), Bill Donald, John Stock-Allchurch, T. Harrison-Smith, N. Yule, Mark Smythe, Ivan Drake,

Richard Floyd, Graham Carpenter, Paul Ostling, J. Tanner, John Whitley and Colin Spencer.

I'll be pleased to hear from new communicators but please bear with me. I cannot give instant turnaround these days. After some weekends it is not unusual to find 10, or more, mailboxes to answer!

--oOo--

TEACH YOUR GRANDMOTHER 9

You youngsters don't know how lucky you are to have compilers cheaply available to speed up your programs! I can still remember the days when you had to plan your program to speed things up a bit.

Just in case any of you haven't a handy compiler the following techniques will help to push your machine to its limit:

1 Identify the code which is most used. Put heavily used subroutines at the start of the program even if this means GOTOing around them.

2 Loops in the busiest code should be really compact. Forget about style and put as much as you can on each line. Explanatory REMs can go at the end of the program.

3 If you have a series of tests such as :-

```
IF ASC(MID$(CH$+SP$,7,1))=77 THEN TY=1
```

```
IF ASC(MID$(CH$+SP$,7,1))=76 THEN TY=2
```

```
IF ASC(MID$(CH$+SP$,7,1))=49 THEN TY=3
```

etc

it is very useful to resolve the complex formulae just once with $T=ASC(MID$(CH$+SP$,7,1))$ and then simply test T; ie IF T=77 THEN TY=1 etc

Grandpa

PLUS 4/C 16 PRELIMINARY MEMORY MAP

The following pages contain the first attempt at a memory map for both the C16 and + 4. It is intended for those who are taking their first delve inside the machines and is certainly not the final product. A more complete and comparative map is already in preparation. These notes should be read in conjunction with Mike Todd's article 'The C16 and Plus 4 - a preliminary look' in the September Newsletter.

ADDRESS NOTES

0000-0001 8 bit I/O port - used mainly for serial bus
0002 Stack search target byte (For)
0003-0004 RENUMBER - new starting line#
0005-0006 RENUMBER - line increment
0007-0029 As C64
002B-002C Start of BASIC text
002D-002E Start of variables - end of BASIC text
002F-0030 Start of arrays
0031-0032 Start of space for strings - end of arrays
0033-0034 Bottom of strings
0035-0036 String pointer
0037-0038 Upper limit of RAM available for BASIC
0039-003A Current line#
003B-003C TXTPTR for CHRGET
003D-003E Pointer to entry in GOSUB/FOR stack
003F-0040 DATA line#
0041-0072 As C64 (some locations also used by new commands)

0073-0074 AUTO line increment
0075 Graphics area allocated flag =FF if graphics allocated
0076 F-key number; PRINT USING workspace
0077-0078 MID\$= params; F-key length; MONITOR
0079-007B DS\$ descriptor
007C-007D GOSUB/FOR stack pointer
007E Temp store of colour/luminance/register#
007F Temp store of colour being set during COLOR
0080 Current VOL; current voice; temp during graphics routines
0081 Direct mode flag
0082 DISK - syntax byte; PRINT USING
0083 Graphics mode - B7=MCM B6=split screen B5=graphics

ADDRESS NOTES

0084	Current colour register#
0085	Colour# 2 - multicolour 1
0086	Colour# 1 - foreground/character colour (graphics mode)
0087	Screen width (=40 for hi-res; =20 for multicolour)
0088	# of lines on screen
0089-008B	PAINT flags
008C-008D	Pointer to screen RAM for hi-res routines
008E-008F	SSHAPE
008E-008F	Scaling factor for graphics routines
0090-0095	As C64
0096	Count of external ROMs being checked during RESET
0097	# files OPEN
0098	Default IN device
0099	Default OUT device
009A	MSFLAG to suppress messages (\$80 direct mode, \$00 in prog)
009B-009C	Address of byte currently being SAVED (IEEE)
009D-009E	SAVE end address; address of IEEE byte being LOADED
009F-00A2	MONITOR
00A3-00A5	TI clock
00A6	EOI received flag
00A7	Cassette write - character for output
00A8	Serial - character received
00A9-00AA	Pointer to colour memory during scroll/INS etc.
00AA	Counter during serial input
00AB	Length of filename
00AC	Logical file number
00AD	Secondary address
00AE	Device #
00AF-00B0	Start of filename
00B1	Count of # of error to be corrected during cassette read
00B2-00B3	SAVE start address
00B4-00B5	LOAD start address
00B6-00B7	Pointer to start of cassette buffer
00B8-00B9	Pointer to start of KERNAL vector table
00BA	Flag during CLOSE - prevents CLOSE of SA=15 if B7=0
00BA-00BB	Address of byte currently being LOADED from cassette
00BC-00BD	Ptr to output string (=address of byte after JSR)

ADDRESS NOTES

00BE-00BF	Address of byte to be read from external ROM
00C0-00C1	Start address of screen line during scroll etc.
00C2	RVS flag
00C3	Right hand of screen during screen input
00C4	Cursor line log
00C5	Cursor position log
00C6	Matrix value of key pressed
00C7	Flag input is from screen
00C8-00C9	Start address of current screen line (character)
00CA	Cursor position
00CB	Quotes flag b0=1 if in quotes mode
00CC	Temp store of cursor column during scroll
00CD	Current cursor line#
00CE	Character under cursor; character for output to screen
00CF	INS counter
00D0-00EB	unused (originally for speech synthesiser & applications)
00E9	CIRCLE - segment angle increment
00EA-00EB	Start address of current screen line (colour)
00EC-00ED	Pointer to keyboard decoding table
00EE	Keyboard image from scan
00EF	# chars in keyboard buffer
00F0	Halt display flag <>0 to stop screen display (set by CTL-S)
00F1-00F6	MONITOR usage
00F7	Pass number on cassette read b7=1 first time b7=0 on 2nd
00F8	Cassette block type ID
00F9	HSD mode (b7=o/p b6=i/p) & device#
00FA	Temp .X store during STOP key check
00FB	Current ROM bank#
00FC	RS232 X-ON flag = 17 to operate (=CTL-S)
00FD	RS232 X-OFF flag = 19 to operate (=CTL-Q)
00FE	Temp store for cursor line# during scroll
0100-010F	Number to ASCII building area
0110-0112	.A/.Y/.X save during CHROUT
0113-0122	Colour code conversion table (inc luminance)

ADDRESS NOTES

0200-0258 Input buffer

0259-025A Line# after STOP
025B-025C TXTPTR after STOP

025D-02AC DISK command workspace

02AD-02B0 Pixel cursor - 02AD/02AE=X 02AF/02B0=Y
02B1-02E3 Table of graphics co-ordinates and misc workspace
02E4 Page# of character generator - normally \$D0
02E6 Scaling on/off flag =0 if off <>0 if on
02E7 Flag to plot point double width (always 0)
02E8 BOX Flag to plot point double width (for PAINT option)
02E9 Bit to be set during plot point subroutine
02EA CHAR - # of chars in string

02EB TRACE - on=FF off=00

02EC DIRECTORY - counter
02ED-02EE DIRECTORY - # blocks in file entry (=line# from PRG)

02EF Used during co-ordinate scaling
02F0 Temp storage of pointer to graphics variable
02F1 Flag relative co-ordinates (B7=1 if + or - preceded co-ord)

02F2-02F3 FLPT to INT vector
02F4-02F5 INT to FLPT vector
02F6-02FD unused

0300-0301 =8686 Error vector
0302-0303 =8712 READY vector
0304-0305 =8956 Crunch routine
0306-0307 =8B6E LIST
0308-0309 =8BD6 Main execution loop
030A-030B =9417 Arithmetic element evaluation
030C-030D =896A Crunch new keyword (after escape token)
030E-030F =9B88 List new keyword (after escape token)
0310-0311 =8C8B Execute new keyword (after escape token)

ADDRESS NOTES

0312-0313	=CE42 Extra IRQ intercept point
0314-0315	=CE0E Main IRQ vector
0316-0317	=F44C BRK vector
0318-0319	=EF53 OPEN
031A-031B	=EE5D CLOSE
031C-031D	=ED18 CHKIN
031E-031F	=ED60 CHKOUT
0320-0321	=F0C CLRCHN
0322-0323	=EBE8 BASIN
0324-0325	=EC4B BASOUT
0326-0327	=F265 STOP
0328-0329	=EBD9 GETIN
032A-032B	=EF08 CLALL
032C-032D	=F44C (BRK vector - unused)
032E-032F	=F04A LOAD
0330-0331	=F1A4 SAVE
0333-03F2	Cassette buffer
03F3-03F4	Cassette write - # bytes to write (2's complement)
03F5-03F6	Cassette read - # bytes to read (2's complement)
03F7-0436	RS232 input buffer
0437-0472	Cassette read error log
0473-0485	CHRGET (0479=CHRGOT) subroutine
0494-04E6	LDA (xx),Y subroutines (see last issue)
04E7-04EA	PUDEF - fill symbols - space , . \$
04EB-04EC	INSTR - pointer to 1st string descriptor
04ED-04EE	INSTR - pointer to 2nd string descriptor
04EF	ERR - # of error that occurred
04F0-04F1	ERRL - line# where error occurred
04F2-04F3	TRAP - line# to execute on error (04F3=FF if off)
04F4	TRAP - state of TRAP prior to RESUME
04F5-04F6	ERR - pointer (from 025B/025C)
04F7	Stack pointer prior to command - used for error trap
04F8-04F9	DO - TXTPTR to start of loop
04FA-04FB	DO - Line# of start of loop
04FC-04FD	SOUND - LO timing bytes for voices 1/2
04FE-04FF	SOUND - HI timing bytes for voices 1/2

ADDRESS NOTES

0500-0502	USR vector
0503-0507	RND seed
0508	Reset flag = \$A5 after RAM pointers set
0509-0512	Logical file number table
0513-051C	Device# table
051D-0526	Secondary address table
0527-0530	Keyboard buffer
0531-0532	Bottom of RAM (normally \$1000)
0533-0534	Top of RAM (normally +4=\$FD00 C16=\$3FF6)
0535	Serial time out flag (set but never checked)
0536	unused
0537	Cassette write - set to 0 on OPEN but never accessed
0538	unused
0539	Cassette buffer pointer
053A	unused
053B	Current character colour - for text mode
053C	Current character flash mode (ON:b7=1 OFF:b7=0)
053D	unused
053E	Page of screen RAM - set but never accessed
053F	Max size of keyboard buffer (set to \$0A)
0540	Keyboard repeat flag
0541	Repeat speed
0542	Delay before repeat
0543	Shift/CBM/CTL key flags
0544	Screen flip countdown - used to prevent bounce
0545-0546	Address of key decoding routine (=DB7A)
0547	Screen flip lockout
0548	Auto scroll down flag
0549	Used in screen routines (store of cursor log)
054A	Set to \$0A on screen reset but never accessed
054B-055C	MONITOR
055D	# chars in current F-key
055E	Pointer to current F-key definition
055F-0566	F-key lengths (0-8)
0567-05E6	F-key definitions

ADDRESS NOTES

05E7-05EB	High speed disk - workspace
05EC-05FF	External ROM identities - 0=no ROM/1=init immediately
05F0-05F1	External ROM interrupt vector
05F2-05F4	.A/.X/.SR - saved during external ROM interrupt service
05F5-06EB	unused
06EC-07AF	GOSUB/FOR stack
07B0	Cassette read - =FF if reading to buffer
07B1	Cassette byte parity
07B2	Pointer to filename during transfer into buffer
07B3	Pointer to buffer during filename transfer
07B4	unused
07B5	Bit count during cassette read
07B6	Pass 1 & 2 error counters - points into error table
07B7	Pass 1 error count (set from 07B6 after 1st pass)
07B8-07BD	Cassette read timing constant (for FF02/FF03)
07BE	Temp stack pointer store during cassette read
07BF	Temp stack pointer store at start of building byte
07C0-07C1	Start address of cassette read
07C2-07C3	Number of bytes to be read (2's complement)
07C4	Error flag during cassette read pass 2
07C5	Counter for start of read synchronisation
07C6	Counter to test succesful synchronisation
07C7	Temp store for serial char during VERIFY
07C8-07C9	Cassette write timing constant (for FF02/FF03)
07CA-07CB	.Y/.X temp store during bit write
07CC	Flag used during byte building
07CD	RS232 byte for output
07CE	RS232 character waiting (b7=1 if waiting)
07CF	RS232 X-ON/X-OFF to be TX'd
07D0	RS232 flag to indicate X-ON/X-OFF to be transmitted
07D1	RS232 input buffer pointer - points to last char entered
07D2	RS232 input buffer pointer - points to last char read
07D3	RS232 count of number of characters in input buffer

ADDRESS NOTES

07D4	RS232 status
07D5	RS232 character read from FD00
07D6	RS232 X-ON/X-OFF flag - =FF tx held until X-ON received
07D7	RS232 flag input buffer not receiving (if X-OFF was sent)
07D8	RS232 flag RS232 is active
07D9-07E4 LDA (\$xx),Y where (\$xx) is held in \$07DF	
07E5	Screen window bottom - set by ESC-B
07E6	Screen window top - set by ESC-T
07E7	Screen window left - set by ESC-T
07E8	Screen window right - set by ESC-B
07E9	Screen scroll flag - set by ESC-M cleared by ESC-L
07EA	Auto insert flag - set by ESC-A cleared by ESC-C
07EB	Last character printed on screen (to identify ESC sequence)
07EC	Force scroll up - use during ESC-D (delete line)
07ED	Current 'cursor' colour
07EE-07F1	Screen wrap around table (bit mapped)
07F2-07F5	.A/.X/.Y/.SR used by SYS
07F6	Last key value pressed (for repeat)
07F7	List pause disable flag (disables CTL-S if <>0)
07F8	ROM/RAM flag for monitor (b7=0:read ROM b7=1:read RAM)
07F9	Colour code table flag (b7=1:codes from \$E143 b7=0:\$0113)
07FA	Raster control byte (= %0000 0100)
07FB	Page number of hi-res screen (= \$18)

Normal

0800-0BFF	TEXT screen colour RAM
0C00-0FFF	TEXT screen character RAM
1000-FCFF	BASIC RAM area (1000-3FFF on C16)

GRAPHICS mode

0800-0BFF	TEXT screen colour RAM (+4 and C16)
0C00-0FFF	TEXT screen character RAM (+4 and C16)
1000-17FF	UNUSED (used in C16 for BASIC RAM)
1800-1BFF	Hi-res screen (luminance?) (+4 and C16)
1C00-1FFF	Hi-res screen (colours?) (+4 and C16)
2000-3FFF	Hi-res screen bit mapping area (+4 and C16)
4000-FCFF	BASIC RAM area (+4 only)

ADDRESS NOTES

800A/8019 Warm/cold start routines
8155 Variety of LDA (xx),Y routines
818E Start of keywords table
8470 Start of error messages
8703 READY entry
8953 Crunch tokens
8A79 NEW
8AFF LIST
8BD3 Main statement execution loop
A954 Garbage collection
AEF7 PRINT USING
B42B TRAP
B729 KEY
BC20 ASCII names: FRED B TERRY R MIKE I
B8D1 Start of graphics commands
C8BC Start of disk commands
CD74 BASIC patches area
CD89 Encoded names: F BOWEN J COOPER B HERD T RYAN
CDAB Routine to print encoded names
CE00 Main IRQ service routine
CE60 Split screen
CECD SOUND interrupt handler
CEFD Update TI\$

D000-D7FF Character generator

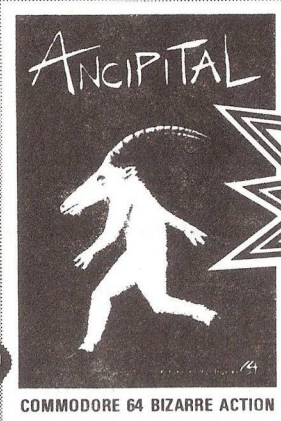
D802 Screen address pointers
D88B Clear screen
DC49 ASCII character to screen
DE48 ESC sequence handler
E01E Keyboard decoding
E153 Serial routines
E319 Cassette routines
EA5B RS232 routines
EB58 KERNAL error messages

ADDRESS NOTES

EC96/ECE6	Input/output from high speed disk
EDA9	Check if high speed disk interface is present
F30B	Initialise I/O
F352	KERNAL initialise routine
F445	MONITOR
FC1E	Hunt & log external ROMs - initialise if ID=1
FCB3	Main IRQ entry
FCF1	Execute code in external ROM
FCF4	Initialise all ROMs found
FCF7	Get a byte from an external ROM
FCFA	Handle external ROM interrupt routine
FCFD	Resume KERNAL interruptFD00 6551 for RS232
FD10	Parallel I/O (also FD20 & FD30)
FDD0	External ROM/cartridge select STA FDD0,X
FEC0/FEFF	High speed disk - device# 9/8
FR00	TED chip control registers
FR00/5	Timer 1 - 2 - 3
FR06	B5=bit map mode B4=screen off
FR07	B4=multi colour mode
FR08	Keyboard column input
FR09/A	Interrupt flag and enable
FR0B	Raster count
FR0C/D	Position of cursor
FR0E/11	Sound registers
FR12	Bit mapped memory address
FR13	Character generator address
FR14	Screen memory base
FR15/9	Screen colours
FR1C/D	Raster line for interrupt
FR3E	Select ROM
FR3F	Select RAM
FR49	New vectors for +4/C16
FR80	KERNAL version number
FR81	KERNAL vector table
FRF6	Initial RESET point
FRFC/F	RESET vector (FRF6) and IRQ vector (FCB3)

Mike Todd

SEND SAE FOR
LARGE ANCIPITAL
MAP



£7.50

COMMODORE 64 BIZARRE ACTION

NEW CBM 64 DISCS
REVENGE
ANCIPITAL
HOVER BOULIER
SHEEP IN SPACE
£9.50

ANOTHER GREAT ORIGINAL FROM LLAMASOFT



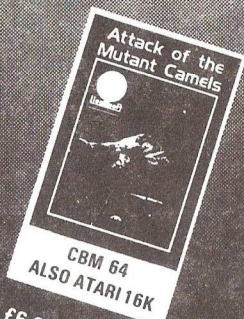
CBM 64

£7.50



VIC 20
ALSO CBM 64

£5.50



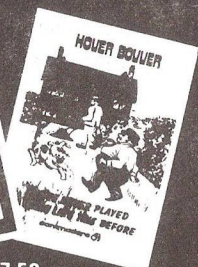
CBM 64
ALSO ATARI 16K

£6.00

£7.50



£7.50



CHALLENGING, WITTY, ABSORBING

LLAMASOFT

PLEASE SEND 50p P&P WITH EACH ORDER

49 MOUNT PLEASANT, TADLEY, BASINGSTOKE, HANTS.

SUPERBASE CORNER

edited by Hugh de Glanville

Well, here we are with our second issue of this new feature, possibly still slightly disorganized for reasons not unconnected with the Editor's physical state (see Notebook).

Recently both a reader and one of the computer journals made a very essential point about databases like Superbase, which perhaps puts what we are trying to do into perspective. It has to be realised that when you buy Superbase, you are not really just buying a data/file management system, you are buying a high-level language which has thrown in with it all the mechanics of managing data and files. You can at an elementary level just make mechanical use of its data management, or you can, as obviously many of us want to do, exploit it for really quite complex purposes.

The way this feature seems to be shaping is that we shall usually have some major section on programming from PS sources (Brian Leighfield and/or Tom Cranstoun); we shall have their help in answering vexing questions; we shall have readers' comments and queries; and for the moment the whole will be wrapped up in and accompanied by my comments and observations from my own fairly busy, if not very sophisticated, use of SB for business purposes. So this month, after my opening remarks, we have answers to one reader's several queries, and then, at rather more length than we expect to have usually, an explanation of how the standard SB Labels program works, as an introduction to the higher forms of programming. A particularly interesting section of this is where it goes into the somewhat perplexing "do" command: you are strongly encouraged to fire up your computer and work through the examples given there as you read the section, and not just to read through the code.

Incidentally, I was wrong in crediting Brian Leighfield with authorship of the SB Applications programs in the September 'Shop Window'. Any credit due for these goes to Roger Lamb, G Rook, and Dr Bruce Hunt.

The SB self-destruct function

Has anyone else run into the situation in which SB clearly instructs one to destroy one's own data disk?! This can arise if you are working

with a database in drive 1 of a double drive and you decide to format a new disk. Normally one would be doing this in drive 0, and SB tells you to put the blank disk in drive 0 and duly warns you that anything on the disk will be lost. What it actually does is to format the disk in the drive in which it is currently working, so if this happens to be drive 1, although it says 'put the blank disk in drive 0', it will then go ahead and format the disk in drive 1 - your database if you don't catch it in time by noticing, for example, that the wrong drive light has come on. Not a situation likely to crop up very often, but just the sort of 'funny' that I specialize in stumbling on.

Date formats

Is anyone perplexed by the fact that dates sometimes come up in US and sometimes in UK format? The default as the standard program loads is the US format, e.g. NOV3084, but as soon as you enter a date in the other format, all the dates flip to that form. If you would like the program always to use the UK format from the word go, you can simply insert a line in your start program saying 'date "00JAN00"' and then all your dates will be in UK form from the outset.

Timing operations

Talking of dates and times, although I have not seen it mentioned anywhere in the manual, ti\$ is available within Superbase, and it is very easy to add timing functions into programs. In fact some programmers seem to take an absolute delight in doing so, and I have a couple of programs that helpful friends have written for me that simply cover the screen with details of how many records they have processed in how many minutes and seconds, how long each cycle is taking, how long since what-have-you started - almost everything except how long before the next cup of coffee will arrive.

To take a very simple example, the following would time a 'sort'. I give the program because it illustrates the interconversion of str\$ and VAL involved in the operation, which may help other beginners.

```
10 st$=ti$: rem note program start time
20 file "x":sort all on [key][text1]
30 ft$=ti$: rem take finishing time
40 t=val(ft$)-val(st$):t$=str$(t): rem work out interval as a string
50 display @0@10,10 "Time taken = " left$(t$,2) " hours, and "
mid$(t$,3,2) " minutes":wait:menu
```

Count catching

A number of users wish that SB's counts were not so evanescent, apart from being usually one ahead of themselves. I have made close enquiry of the experts as to whether there is any way in which those fleeting figures up in the top right-hand corner of the screen could be 'fixed' in any useful way, but the answer is a definite No. It seems there really is no way SB's own internal counts can be dumped at the end of an operation. It is of course open to any user to slip a counter into his own program and display the result at the end, but we cannot get our hands on those elusive right-hand-corner figures, nohow.

Field numbers

For any program that makes frequent use of field names, or needs to store them in an array, it is convenient to have numeral field names. At the same time, explanatory text is more use on the screen display. The fact that SB reads its field designations like an oil sheikh (is there any significance in that, I wonder?), from right to left, and stops at the first word space it meets, makes it very easy to combine the best of the two worlds. Thus

```
key 1 [      ]
name 2 [          ]
addr 3 [          ]
```

gives us meaningful field names on screen, but as far as SB itself is concerned the field names are 1, 2, 3, &c. Note that although these are 'numbers' they are still string variables at this stage, and for numerical handling in a program they will have to be converted to numeric variables by the VAL function, and later converted back again, remembering to chop off BASIC's leading space for the minus sign. So if we have a field numbered 'k' (actually k\$), to turn it into a numeric variable

$$k = \text{val}(k\$)$$

and when we want to convert back again

$$k\$ = \text{mid}\$(\text{str}\$(k),2).$$

That is enough low-level comment from your editor. Let us now proceed to the genuine stuff from more erudite pens. And as we turn the page, this Column wishes all Superbasers fun over Christmas with the "do" function, and the best of programming in 1985.

More or less Superbase Corner's first query customer is Ifan Payne of Glamorgan who sent in two programs he had written for Superbase himself and asked several interesting questions. He finds Superbase an 'excellent but frustrating program; frustrating in that there are so many things that one knows could be done with it but so little guidance is offered. The first thing that could be done is to convince PS to create an index for its Manuals.'

Well, I am glad to say that Superbase 64 Manuals now have an index, and one is being prepared for the 96/700 Manuals. Space does not allow giving both Ifan's programs, but Tom C has used one of them, with slight modification which we are sure will be forgiven, to demonstrate the answer to one of the questions asked.

The questions are (1) How does one transfer 40-column SB64 files from 1541 to 8250 disks? (2) How does one get the 6400 printer to underline with Superbase? (3) How does one arrange automatic page numbering for 'reports', preferably within a standard page heading?

(1) 1541 to 8250 transfer is impossible directly, because the DOS are not compatible, but sequential files can be copied across with Jim Butterfield's "Copy all" program or a similar utility from the ICPUG library (for either 64 or 96 systems). As your 64 is being used with a DAMS interface, a 4040 will have to be used to read the 1541 disks, as the DAMS does not allow simultaneous access to the IEEE and serial buses. The whole procedure will be as follows.

All Superbase files except the Database itself are sequential files (so are Easyscript files for that matter). To transfer Database files you will need to:-

- 1: Take a hard copy of the STATUS of each database file.
- 2: Export all the database files.
3. Quit Superbase and load "Copy all"
4. Transfer the sequential files to a newly formatted disk on the 8250.

WARNING: do not try to copy the database.

Also do not copy the file definitions unless you want them for working a 64/8250 system as well. If you do this, onto the same disk, then rename them so that they do not interfere with your 8000 database.

5. Load up Superbase on the 8000 series machine and create a database.
6. Enter the original file names. The message "File does not exist: create it?" will be displayed in response to each name. Answer 'y' to this prompt and then, unless you did copy the file defs across and not

rename them (in which case you would be returned to the main menu), you will be invited to create new 80-column formats for your files.
7. Import the data files.

(If you are lazy, like the Editor, it is possible [but not really recommended by Tom] to take 64 file definitions across into an 8000 database, and then call up FORMAT and rearrange them on screen. The reverse is not possible.)

(2) To underline with the 6400 printer from Superbase, set the dipswitches to ASC and use the following routine:

```
10 u1$=chr$(27)+"E":u0$=chr$(27)+"R"  
20 print "Here is the";u1$;"Underline Example";u0$;"for your 6400"
```

(3) Tom's answer to how to include automatic page numbering in a report is given in the following modified version of Ifan's own program for automatic line numbering within a report.

```
11 rem based on a program from Ifan Paine  
20 pn=1:rem beginning page number  
30 ln=1:rem count of lines printed  
40 i=1.01:rem item counter  
50 ep=11:rem end of page line (+1)  
100 display @20,1@-"REPORT TITLE"&3,0;pn@-:rem xx,1 forces new page if  
    needed  
200 display @20,3"SUBTITLE"  
300 display @20,5"LIST HEADING"  
400 display @20,7:rem position print head away from titles  
600 select next:rem use select from "listname" if required  
700 display i"TEXT ITEMS":rem use meaningful field names/ variables  
    here  
800 i=i+0.01:rem increment item counter  
900 eof menu:rem use eof menu if using a list  
950 ln=ln+1:if ln=epthen pn=pn+1:ln=1:goto 100:rem inc line count  
1000 goto 600:rem repeat until finished.
```

Now on to our programming article, this month's pièce de resistance.

THE SUPERBASE LABELS PROGRAM

The labels program listed below can be found on every Superbase program disk. Therefore for anyone developing an interest in programming with Superbase it provides a valuable insight into use of the database. In this article we shall have a look at the labels program in order to see how it works and how it may be amended. In a following article the theme will be continued with a look at some further amendments and alternative methods of dealing with "The labels problem". The article is also, we hope, going to generate some feedback and we also hope that other Superbase users will contribute their thoughts on Superbase programming.

Mick Ryan has offered some very interesting label routines for listing and comment, and we plan to publish these later. The present article is really an attempt to ensure that we are all on a common footing by explaining how the program works. The article is has turned out rather longer than we hoped but we felt it necessary to produce sufficient material in the first main piece to provide a common basis for developing the tone of the users' corner. For the sake of space and sanity, a knowledge of Commodore Basic is assumed.

Finally, a message from one of the program authors. Tom Cranstoun has asked me to point out that the labels program was written in a hurry and is not meant to represent a definitive programming style. Well, this may be true for those competent at the heady levels that Tom operates on: for the rest of us who still are learning to program it represents an example of crisp technique. Over 50 000 users have it on their disks and a lot of them must be using it without any bother, or Precision Software Technical Support would be inundated with queries about it, whereas the vast majority of enquiries about the program are from those new users who do not seem to recognize that as it is a Superbase program, it only works within Superbase and cannot be run on its own.

First rules

The program assumes (1) that the user has made hard copy of (or committed to memory) the names of the fields in the file definition of database file he intends to print labels from;

(2) that the lengths of these fields, in the print size chosen, will fit into the label format;

(3) that if a select list is to be used, this has already been created and its name remembered.

If the above information has been entered correctly then the program will spin away and create your labels for you for very little effort. However it was only designed to be a basic, 'get-you-going', routine, and there are several ways in which one might want to amend it.

For instance, as it stands, the program requires an addressee's title, forename, and surname all to be available in one field, whereas it is often much more convenient from the database point of view to have the surname in a field by itself as key field, so that the database keeps itself automatically in surname alpha order, and records can be selected by surname. The title and forenames then have to be kept in other (text) fields. On this basis, the label program needs to be able to concatenate these two or three fields into one on the first line of the label.

The existing program provides for printing up to four labels across the page, for label widths of between 19 and 36 characters, and between 2 and 12 lines per label. These ranges will cater for most label operations. They can be extended it should be noted that care must be taken not to run out of memory.

The present article will attempt to provide you with sufficient information to make modifications in the above areas. In future articles we hope to be able to deal with the following amenities as well.

Some labels require a serial number to be printed on them (e.g. for manufactured goods), or a despatch number, or a date may be wanted on the label.

The present program prints one label for each record selected from the database file, selecting the records either in sequence from the first record in the file to the last, or from a previously created select list. Sometimes it is useful to print more than one label for each record.

Also as it stands it will not allow for the generation of half line feeds and other exotica necessary for those who have the misfortune to need labels printed with odd line increments.

THE PROGRAM

The first section of the program sets up the system parameters and names the string variables for screen control and error message display. To avoid unnecessary repetition of lengthy display statements, lines used more than once in the program are placed in variables to be called at appropriate points.

```

100 rem Label program, Precision Software, 1983
110 display:across:lmarg:rmarg:space:cont 1
150 err$=chr$(7)+"Incorrect Response, Retry"
160 cl$="display chr$(147)@1,2"+chr$(34)+" Superbase Label
    Program"+chr$(13)
170 cl$=cl$+" Precision Software Ltd. (c) 1983"
180 catalog:do "display@0"+mid$(cl$,18)+"

```

The command "catalog" on line 180 will display the database catalog on the screen to allow the user to select the file to print from in response to the prompt on the next line.

When run the program will display the catalog in exactly the same way as it is displayed when the "FILE" option is selected from Superbase menu 2, except that the prompt "Enter file name" does not appear/

Also on line 180 is the first use of the high-level command "do" in this program.

```

do "display@0"+mid$(cl$,18)+"

```

In order to understand the labels program we need to divert here for a moment and take a close look at the operation of the "do" command. If you already understand what the command does perhaps you can afford to ignore the next section.

Using the DO command

Load up Superbase and enter the following program:

```

10 a$="fred"
20 d$="display@1,6;chr$(34)+a$+chr$(34)
30 display@0@1,2;a$;@1,4;d$
40 do d$
50 wait

```

When run this program will demonstrate the action of the "do" command on the text strings, though this would be a very trivial way to use it, not worth bothering to do. One slightly more interesting example is the transposing of the contents of a string variable (e.g. a\$) which is itself a variable name (e.g. b\$) to the data string which has been assigned to b\$.

```

10 b$="fred"
20 a$="b$"
30 do "c$="+a$
40 display @1,6;b$;@1,8;a$;@1,10;c$*ju1

```

The value of this is that a field name can be held in memory in the form of a string variable.

```
10 a$="[name]"
```

Notice that the field name is put in quotes. The string variable therefore consists of the name of a field, not its contents. When one wishes to get at the data actually held in that field of a current record it is necessary to transpose the statement by a "do" command.

For the purposes of the next exercise, create a test file with a field in it called "name".

```
10 display @0@1,4"Enter the field name : ";@0:ask&16@25,4;a$
20 if left$(a$,1)>"[" or left$(a$,1)<"["then a$="["+a$
30 if right$(a$,1)>"]" or right$(a$,1)<"]"then a$=a$+"]"
40 display@0@1,6" a$=";a$
50 do"a$="+a$
60 display @0@1,8;a$:
```

Now you should have some inkling as to the use of the "do" command, and we can go back to the labels program.

The section of the program between lines 190 and 230 is concerned with setting global parameters for the operation.

```
190 ask &10"File To Print From";f$:file f$:select f:do cl$
```

This line asks for the name of the file (remember the catalog is currently being displayed). The file is then selected and initialised by selecting the first record.

```
200 display @1,6"Use a Select list, (y/n) : n"@0:ask &1@28,6;a$:do cl$
210 if a$="y"or a$="Y"then sl=1
220 if sl=1then display @0:ask @1,6"Enter Select List name :";sl$:
    do cl$
230 if sl=1then select from sl$
```

The numeric variable "sl" is used as a "flag" which indicates whether a select list will be used. If the option has been chosen (a\$=y) sl is set to 1. A new display (line 220) allows the placing of the select

list name into the variable assigned (sl\$) and the program selects the first record from the list.

We are now in a position to set up the print parameters for the label run.

```
300 display @1,6"How Many Labels (1-4) Across Page : 1"@0:
    ask &1@37,6;n:do cl$
310 if n>4or n<1then display @1,8@+err$@0:goto 300
320 display @1,6"Enter Label Width ( >19 chars.) : 19"@0:
    ask &2@35,6;w: do cl$
325 ls=w;if w>36then ls=36
330 if w<19or w*n>254then display @1,8@+err$@0:goto 320
340 display @1,6"Enter Start Column : 1"@:ask &2@22,6;s:do cl$
350 if s<1then display err$@0:goto 340
360 display @1,6"Enter No of Lines Between Labels : 9"@:ask &2@36,6;l:
    do cl$
370 if l<2or l>12then display @1,8@+err$@0:goto 360
```

The above section is simply concerned with getting the numeric data that governs the print format. By the end of execution of line 370 these system variables have been assigned:

n : the number of labels across the page

w : the number of spaces between the first character position of one label and the next on the same row

ls: this is a control variable to draw the label on the screen

s : the first column position to be printed

l : the number of lines from the top line of one row of labels to the top line of the next row.

Experienced users will recognise that if they want to hard-code these values into the system they can simply remove lines 310 to 378 and substitute one line of assignment statements for them, e.g.

```
300 n=1:w=35:s=40:l=12:rem print one label across, 35 characters wide
310 rem No of lines between labels=12. Start print at column 40
```

Providing you always use the same label roll and the same position on the printer platen, this can save much tedious repeat entering.

The arrays

```
300 dim l$(n,l),f$(l)
```

l\$(n,l) is used to hold the data to be printed. f\$(l) will contain the field names (complete with square brackets), or a symbol to indicate that nothing is to be printed on that line.

Lines 400-490 generate the screen display defining the field layout for the label print. Keyboard entry fills the array f\$(i) with field names; or asterisks to signify a blank line is to be printed.

```
400 display @1,8""+left$("                               ",ls)+ " "
420 for i=1to l:display @0@1,8+i"  *";@ls+2" "
430 display @0@1,6"Enter Field Name For Line";&2,0i
440 ask &14@4,8+i;f$(i)
450 if left$(f$(i),1)<>"["then f$(i)="["+f$(i)
460 if right$(f$(i),1)<>"]"then f$(i)=f$(i)+"]"
465 if f$(i)="*"then 470
466 check f$(i):display @0@1,6"Incorrect name, re-enter ":goto 440
470 display @0@1,8+i; " "+f$(i)
480 next:i$=left$("                               ",ls)
490 display @0@1,9+l""+u$+ " "
```

If one wishes to hard-code this section to avoid keyboard entry then these lines could be replaced by defining f\$(i) for each line and removing all the other lines between 400 and 500.

```
440 f$(1)=[name]
450 f$(2)=[add1]:f$(3)=[add2]:f$(4)=[town]
460 f$(6)=[county]:f$(7)=[code]
470 f$(8)="*":f$(9)="*"
480 rem no of lines between labels is set to 9
```

Of course the value entered as l should also be set to 9 in this example.

If one wishes to print more than one field on a line then this is where the program can be amended.

```
410 a$=chr$(34)+"&[app]" +chr$(34)+" "
```

```

420 a$=a$+chr$(34)+"&[fname]" +chr$(34)+" "
430 a$=a$+chr$(34)+"&[lname]" +chr$(34)
440 do "f$(1)=" +a$

```

The example above will allow for the concatenation of three fields with truncation applied to each field to remove trailing spaces.

The final section of the program is concerned with selecting records and loading the array l\$(n,l) with the contents of the data fields according to the label layout specified in the array f\$(l), the entire contents of each label for one row being loaded into the array l\$(n,l) prior to printing. Then the row of labels is printed. The program checks to see if any more records are in the file or in the chosen select list. If more records are to be printed the program loops back to load the array with the next set of records.

```

500 display @0@1,6"Label Printing in Progress Please Wait"
510 tlen l:plen l:print @0;
520 r=n*(w+1):if r>80then rmargin r

```

These lines initialise the printer and set the parameters of paper and text length to equal the number of lines between labels. The right margin only needs to be set if the print width will be greater than 80 characters (if that is the current r margin setting).

```

530 lmargin s:for i=1to n:for j=1to l
540 if f$(j)="" then l$(i,j)="":goto 560
550 do "l$(i,j)=" +f$(j)
560 next:if sl=1then select from sl$:eol select l:select n:sl=0
570 if sl=0then select n
580 q=q+1:next

```

Lines 530 to 580 load the array l\$(n,l): the first record was selected earlier in the program. The first record's fields are the first items to be loaded within the inner loop (forj=1tol). If an empty line is to be printed then a null string is placed in the array (line 540). The next record is then selected from the list if sl=1 or from the file if sl=0. The outer loop then takes over to repeat the operation for the next label in the group. When j reaches n the array is full and the program continues to line 590.

The print routine.

```
590 for i=1to l:for j=1to n
600 x=(j-1)*(w+1)+1:print &w@x;l$(j,i);next:next
2000 eof print " ":display:across:lmarg:rmarg:space:menu
2010 lmarg:display @0@1,5&3,0q"Labels have been Processed":goto 530
```

Once again two nests of for-next loops are used, this time to print the array. The inner loop now executes the print command on the row, and the outer loop moves the print forwards to the array element for the next line. The print head simply prints the row on the next available line. Notice that the print commands are formatted along the row by using a sliding value for the column position x . The formula for determining the column position of the left margin of each label can be explained as follows.

$$x=(j-1)*(w-1)+1$$

Where w =the number of character positions from column 1 of first label in row to column 1 of next label in row, and j is an integer between 1 and n that at any time gives the label number across the page from 1 to n . As the first position to be used will be 0, then the first value of j is decremented. The value $(j-1)$ is then multiplied by the value $(w-1)$ which is the no of characters between labels less a decrement of 1 to balance the -1 used in the first value $(j-1)$.

Finally the value 1 is added to the equation to move the printer head on to the position it will print the first character of the next label on.

After printing the contents of the array l(n,l)$ the program emerges from the outer loop and checks for an end of file. Note that it does not need to check for an end of list, as if one is detected at line 560 then the program executes a select last followed by a select next to cause an end of file. If "End of file" is false then the program loops back to line 530 to load the array with the next set of records. If "End of file" is true then the print head is cleared, output reset to display, and margins and line feed defaults reset. Control is passed back to the menu system.

After reading all this I hope you are still awake: no doubt parts of the exposition are too simple for some and too complex for others, but I hope no one is actually more confused than when he started.

CHRISTMAS IS COMING . . .

What would you like to find in your Xmas stocking this year? An 8296D maybe, or just a printer or disk drive to give your system the finishing touch! To give you a hand with stocking fillers (for the kids!), Paul Jay, author of Commodore's "Faceache" and "Rolf Harris Picture Builder", has perused briefly a selection of games and a couple of graphics utilities for the 64. Take it away Paul . . .

Micro Power Ltd of Leeds have produced four new games;

SWOOP

This is a very good version of the Galaxion type game, where animated birds swoop down at your base and you have to shoot them down. If you like the old Space Invaders type of game, then this is a good variation.

FELIX IN THE FACTORY

Ladders and conveyor belts, a giant rat, and Gremlins. You have to run about collecting things. It's all been seen before, and has been done better elsewhere.

CYBERTRON MISSION

Electrified walls, 64 different rooms to explore, things to shoot and things to avoid. My previous comment applies.

GHOULS

This is a better one, though still not much original about it. You have to move a Pac-man type character to collect the "power jewels". A game of jumping onto platforms and avoiding the nasties.

Next three games from Quicksilva and again a mixed bunch;

STRONTIUM DOG AND AND THE DEATH GAUNTLET

This is just a terrible game. You move your man, Johnny Alpha, from left to right, avoiding a load of stuff moving from right to left! I was particularly annoyed with this game, as apart from being a computer nut, I collect comics, and this character is one of my favourites.

ANT ATTACK

Most of you will have seen this game on the Spectrum (wash my mouth out with soap and water) computer. It is probably the best game, as far as graphics go, available for that machine. It has been faithfully reproduced for the 64. The objective is to rescue the Damsel (or if you are a damsel, the helpless Prince), who is in a spot of bother with a lot of giant ants. The 3-D graphics are superb, though the controls take a bit of getting used to. Buy it!

SEE-SAW

This is the best game I have seen in a long time. For once, as far as I know, it is totally original in concept. You are confronted with a view of a castle, the objective is to scale the wall using a see-saw! However there are bricks being dropped over the wall which you have to slide along the see-saw so that momentum will carry you up. Nasties are trying to catch you and you have to use your brains to figure out the best positioning for the bricks. It's a hard game to describe, but a great one to play.

Finally a game from Alligata of Sheffield

ROCKET ROGER

Fly Roger through caverns collecting crystals. This game was demonstrated to me by the author, and he made it seem so easy, until I tried it for myself! The graphics are very good and the obstacles very difficult to get around. There is a huge playing area, but if you enjoy a challenge, then this is the best game of its type I have seen.

The utilities are both from Mirrorsoft and are recommended.

GAMES CREATOR

A great program for kids to make up their own games easily and without any knowledge of computing. There are obviously some limitations. Firstly you have only one background screen, then different skill levels are merely the same but faster, and only 5 aliens are possible. Also only sprites can be animated, although the the background can be scrolled to the left. The program contains a sprite editor, a graphics editor, a sound editor and a music editor. I would recommend this to anyone who wants to create their own quick games for fun, or to the serious user who wants a quick way to try his ideas in action.

GO SPRITE

This is, quite simply, the best sprite editor on the market. It is all joystick controlled (or a lightpen or the keys can be used if preferred) and uses on-screen icons to make life very easy. The animation facility lets you see your sprites in action; stepping through an animation sequence and moving about the screen at the same time. Up to 32 sprites can be defined at once, multi colour or normal. Sprites can be overlaid one on the other and DATA statements can be created to use the sprites in your own programs. If you are a games writer, or if you ever need to design sprites, then this is easily the best on the market.

THE A E A MICROPATCH—REVIEW

by Mike Todd

Once again I've had the opportunity of looking at an add-on that provides Radio Amateurs with the facility for sending and receiving teleprinted and morse code messages. This time, it's the AEA MICROPATCH of American origin, distributed in the UK by ICS Electronics Ltd.

The MICROPATCH is really two units within the one box - the hardware section consists of the tone detectors which feed logic-level signals to the C64's user port and the ROM, which connects to the cartridge port and has all the software to do the decoding. These two sections, although mounted on the same circuit board, are totally independent, with no direct connection between them.

The unit is about six inches square and an inch deep, and plugs into the cartridge port of the C64, with a connection to the user port by a separate lead (supplied). The unit requires 12V DC at 250mA (not provided), which plugs into the rear of the unit.

For reception, the audio output from the receiver plugs into a miniature jack socket at the rear (a suitable plug is provided), and there's a parallel of this available for feeding to an external loudspeaker. The sensitivity of this input left something to be desired and it was not possible to feed it with the recording output of a receiver - it required a full loudspeaker output.

My problem was that I didn't have an additional loudspeaker available and so it meant plugging headphones into the unit and operating these at fairly high volume to drive the unit successfully.

Connection to the transmitter is through a mic connector cable attached to the unit, which can be wired to a suitable connector for plugging into the transmitter.

Alternatively, the output is available is on two phono sockets - one providing the FSK (Frequency Shift Keying) output for modulating the transmitter, and a keying output for switching the transmitter.

Keying can be used on negative or positive systems simply by cutting the appropriate jumpers internally.

There are four LEDs on the top, one indicates power is connected, the other three are tuning indicators - all very useful, but difficult to view while in operation, as the whole unit extends behind the C64.

Even more difficult to get at were the selection switches designed to switch the hardware between CW (morse), the standard amateur frequency

shift of 170Hz, and a wide shift for RTTY (teletype).

Tone decoding is through a sophisticated dual-filtering system which performs significantly better under poor signal conditions than the cheaper phase-locked loop detectors found on other units.

The handbook talks about the North American tone standards and says the receiver should be set to lower sideband - it took a while for me to discover an amendment pointing out that the unit has been set up to the European standard of 1445Hz for mark and 1275Hz for space, and that the receiver should be set to receive in upper sideband mode.

Once I'd got all this right, tuning was quite straightforward, despite having to lean over the C64 to see the tuning LEDs - it is a shame that these are not visually represented on the screen.

The switch marked 'wide' provides for a frequency shift of 850Hz, but for those who want to receive different frequency shifts, the handbook explains how to re-adjust the unit.

While on the subject of the handbook, I must point out that it is by far the best I have encountered with these types of unit.

Some of the theory of operation is explained, and the manufacturers seem to be more than happy for you to dive inside and make adjustments (with the correct background and equipment, of course). The alignment procedure is given, circuit diagrams; parts list and board layouts are provided; and many other useful bits and pieces are included.

It also covers the theory behind RTTY and gives some tips on how (and where) to receive RTTY signals.

I found the hardware performed extremely well, although the ever-present jungle of wires was something of a nuisance - but that is difficult to avoid.

One of the problems of operating a computer and TV in the vicinity of a radio receiver is the amount of RF interference generated. This can sometimes block out all but the strongest signals.

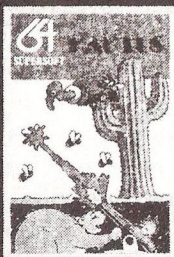
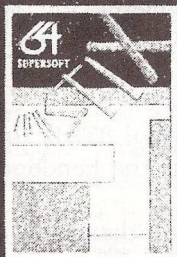
Careful attention to screening, earthing, and positioning solved this problem - that is, until I plugged in the lead that connected the MICROPATCH to the user port. At this point, the RF interference increased and I had to spend some considerable time adjusting my aerial installation.

Those with efficient aerial systems should have no problems at all, those with pieces of wet-string (like mine) may have some problems, but these are not insurmountable and in the process will result in a better receiver performance overall.

That having been done, I also found the RF interference extended to

CAN YOU BEAT OUR GREAT GAMES

We've got the biggest (and best) selection of games for the Commodore 64. We think they're unbeatable



MORE THAN GAMES...

Our FREE catalogue lists dozens of programs for your Commodore 64 to help you run your business, write programs, or just enjoy yourself. We've been producing top quality software since 1978. So if you want the best, just remember our name...

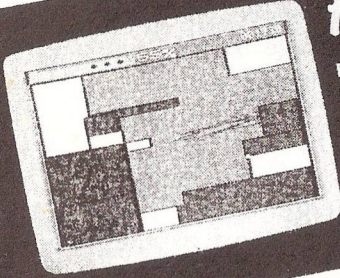
ARCADE GAMES (machine code)	
STIL	
CRAZY KONG	£8.95
WILD FIRE	£8.95
3-D FLOOPER	£8.95
MANGROVE	£8.95
TANK ATTACK	£8.95
	£8.95
ADVENTURE GAMES	
STREETS OF LONDON	£9.95
GOBLIN TOWERS	£9.95
COSMIC CARRIERS	£7.95
CRACKS OF FIRE	£7.95
FORESTLAND	£9.95
FANTASY GAMES	
HALLS OF DEATH	£9.95
LORD OF THE BALLOONS	£9.95
Post free	£6.95

SUPERSOFT

SUPERSOFT, Winchester House, Lanning Road, Wealdstone, Harrow, Middlesex, HA3 7SJ Telephone 01 861 1166

SUPERSOFT

the name to remember



for games

For only £8.95 you can buy a game that's exciting, soothing, and frustratingly addictive - all at the same time! STIX looks so different and sounds so different from all those other games that it will seem like being in another dimension when you sit down to play.



WILDFIRE £8.95



CRAZY KONG £8.95



HALLS OF DEATH £8.95

for business

Show your computer who's master with BUSICALC! Spreadsheet programs are used by large and small businesses to juggle with figures, prepare reports and so on. Some are very powerful indeed. The problem is that they're difficult to learn, and tricky to use - which is why we came up with the BUSICALC series.

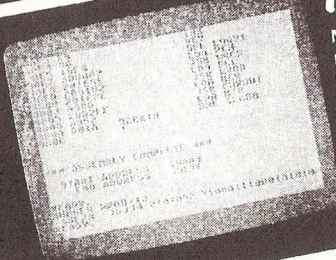
Whether you choose BUSICALC 1, BUSICALC 2, or BUSICALC 3 you'll get a program you can understand - and one that almost seems to understand you. Use it in the home, use it for teaching, use it at work - it'll save you time and money.

Item	Price	Quantity	Total
ITEM 1	10.00	10	100.00
ITEM 2	20.00	5	100.00
ITEM 3	30.00	3	90.00
ITEM 4	40.00	2	80.00
ITEM 5	50.00	1	50.00
GRAND TOTAL			320.00

for programmers

MIKRO is a full 6502 6510 ASSEMBLER with the power that professional programmers need, yet so simple to use that we recommend it to beginners! The MIKRO cartridge has many other facilities including editing commands and a machine language monitor, all for £57.50.

There's much more for the 64 in the SUPERSOFT catalogue. Ask your computer dealer for a copy, or phone 01-861 1166.



The Best 64 Software

SUPERSOFT, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex, HA3 7SJ Telephone: 01-861 1166

the TV set, creating patterning and tuning problems. These all immediately disappeared if I unplugged the user port connector. After a little bit of tuning around, however, I got a good picture and was ready to try out the software section of the MICROPATCH.

A SYS is used to start up the software - a little inconvenient as it could so easily be made to autostart. The screen then shows a menu with the options Morse, ASCII, RTTY, Command Menu, Options, or Back to BASIC.

Options provided are: unshift on space, morse fill, send RTTY idle characters, audio feedback from keyboard, auto carriage return, auto line feed, word wrap around, CW break in, and output mode. The Command menu provides for loading and saving QSO's or message buffers, working with the message buffers, and setting the real-time clock.

Selecting one of the three operating modes (CW, RTTY or ASCII) brings up a screen display, initially in two parts. The top section is a status area showing receive and transmit mode and speeds, the real-time clock, and amount of buffer space remaining. Below this, the rest of the screen is available for displaying received characters as they are placed in the receive buffer.

Of course, selection of RTTY/ASCII or CW will also have to be done with the switches on the unit itself, although I did manage to work much of the time with 170Hz shift selected, even when in CW mode.

Received text accumulates in a 29,000 character buffer and can be subsequently saved on disk or tape. At 75 baud it will take 50 minutes of solid reception to fill this up.

In transmit mode, an additional two areas are shown on the screen. After 12 lines of the received character display, there is a single line showing the 40 most recent characters transmitted, and the bottom 5 lines show the most recent characters entered into the transmit buffer. The transmit buffer can hold up to 1024 characters, but these are removed from the buffer as they are transmitted.

The function keys have an important part to play in the operation. They select receive and transmit mode, clear the receive or transmit buffers, allow the speed to be changed, invert mark/space for RTTY, and lock the speed in CW mode, and take a guess at the speed in RTTY mode. The receive buffer can be turned on or off, the printer selected, and a break-in buffer transmitted.

Control key sequences can transmit a predefined message, automatically return to receive at end of transmission, transmit file from disk or tape, or send the time.

The performance of the unit was good - the RTTY and ASCII modes worked well, provided that the tuning was accurate and the correct frequency shift and speed had been selected. On unknown transmissions, the speed guess feature was invaluable, but it shouldn't be invoked if there is no incoming signal, or the C64 will hang up.

Using the CW section was equally good, although it sometimes took time for the software to lock on to the incoming speed. Once locked, it was able to adjust automatically to minor changes in speed very quickly and major changes in speed after a reasonable time. Incidentally, there is no user-invokable speed guess on CW as it is in operation at all times; instead, the speed can be locked if, for instance, noise is making the software think that the speed is changing.

I was impressed with its performance. The software was easy to use and provided for almost every feature I could have wanted. Where no cables are provided, connectors are included to allow cables to be made up and, with the comprehensive handbook, this goes together to make a well thought-out package.

However, the hardware presentation was a little disappointing. Switches and LEDs were difficult to get at and the large unit hanging out of the back of the C64 was rather awkward to handle. My own preference would be for two separate units - a ROM cartridge (or even disk/tape-based software?) and a free-standing unit that could be positioned to give better access to the mode switches and LEDs.

Overall, I like the whole package and, at £129 (inc VAT + £1.50 p&p) it is a reasonably good buy. There is also a version available for the VIC (at the same price), but with more restricted screen and buffer space.

The units can be obtained from ICS Electronics, PO Box 2, Arundel, West Sussex, BN18 0NX, telephone (024 365) 590. I must thank Alan Clemmetsen (G3VZJ) for his patience in lending me a unit for rather longer than he anticipated.

--oOo--

TEE-SWITCH FOR PRINTERS

by Barry Biddles

(Reproduced by courtesy of ICPUG SE, from their October N/letter No 22)

Many of us, these days, possess two printers: a dot matrix printer for fast listings and drafts, and a daisywheel printer for writing to the bank manager for a further loan. Assuming that they are both centronic interfaced, it is possible to run them, one at a time, from only one interface. However, it can be very inconvenient to have to keep unplugging and reconnecting the interface, especially if the wiring is difficult to reach or hidden within the desk structure. A "tee-switch" is a box which goes between one interface and two printers, and allows either printer to be selected with a switch.

At first glance, this seems almost trivial, and one wonders what could be in a tee-switch to justify the three-figure price tag. To paraphrase Omar,

I often wonder what the dealer buys
One half so costly as the goods he sells.

Well, there are a lot of wires to worry about, and large multi-pole switches are not cheap. Then again, perhaps there are other features that I do not know about, never having met a commercial tee-switch. Probably though, the answer is simply that until recently the numbers involved have been small, and the sales mainly to business users. The purpose of this article is to show the practical electronics handyman how to build his own tee-switch for a tenner or less, depending on the state of the junk-box.

There are several different ways of using the centronics interface, so the first step is to establish what you have got. This is easily done by opening the interface output plug, noting which lines are connected, and what colour they are, then looking in the printer manual to see what they are. There are four types of connection, DATA OUT, CONTROL OUT, CONTROL BACK and POWER/EARTH. My own interface is a rather elderly Mill House Designs unit, and has eight DATA lines, pins 2-9, two CONTROL lines, and EARTH. Power for the interface comes from the printer via pin 18, at 5 volts.

It is not necessary to switch the DATA lines. These are outgoing lines, and can safely be taken to two printers over reasonably short lines.

Experiment showed that it was necessary to switch both of the CONTROL lines in my case, namely STROBE (pin 1) and ACKNLG (pin10). On another type of interface, with a more complex handshake, it may be that there are some control lines which do not need to be switched, and you could minimise the switching required by experimenting. Alternatively, and easier, just switch them all - there will not be very many.

There is a subtle danger connected with pin 18, the 5v supply. It is essential that you do NOT allow both printers simultaneously to supply 5 volts to the interface. This is because power supplies are low impedance devices, and a slight difference between the two voltages could cause large currents to flow. This problem is best overcome by switching the 5-volt line as well. In my case, I had a convenient two-pole switch, which was already fully booked switching STROBE and ACKNLG, so I looked for another answer. The solution was to take power only from the Epson printer. When using the Juki, the Epson has to be switched on simply to power the interface.

Tom Cranstoun recommends that the power connection to the interface is taken via a separate wire to the connector inside the PET, so that the interface is useable with any other printer that does not happen to be wired up on pin 18 (some of the first Epson MX80s were like this). However, I did not wish to add to the plethora of wires behind my machine.

Some of the printer manuals make a lot of fuss about earth returns, suggesting that each data or control line must have a twisted-pair earth return, distinguishing between logic earth and chassis earth etc. All that I can say to this is that in my own experience and that of one other hobbyist that I have spoken to, all this is unnecessary for short lines, say a metre in length. We both used lightweight 12-core screened cable from RS Components Ltd, using the screen as earth, and have not experienced any difficulties.

Simon Tranmer points out that similar methods can be applied to RS232 printers. Also, it does not have to be one computer connected to several printers. His own system consists of one FX80 connected to an IBM XT, a CBM 8032, a CBM 64 and an Apple 2e. He constructed it for a

fiver, saving well over £300 on the commercial equivalent. Incidentally, Simon confirms that for short cables you will probably get away with the simple cableform described above. He warns however that some printers can load the bus so much when switched OFF as to corrupt data transfer, so if you do have any trouble try switching everything ON before concluding that the wiring is deficient.

To summarise:

- 1 Identify the interface output connections, noting the colours.
- 2 Make up the cable for the second printer using lightweight multicore screened cable.
- 3 Solder the eight DATA lines, and the EARTH line, to the appropriate output pins within the interface, in parallel with the connections already there.
- 4 Disconnect all of the CONTROL lines, and connect them to one side of a multi-pole, two throw, break-before-make switch. Connect the new lines to the other side. Connect the poles to the appropriate points on the interface board.
- 5 Decide how to handle the 5-volt problem - see above.

TEACH YOURSELF PROGRAMMING 6—ANSWERS

by Brian Grainger

I asked you last time to create a function which will round its argument to two decimal places irrespective of how many it has to start with.

In BASIC the solution is:

```
DEF FNR(X)=INT(X*100+.5)/100
```

In COMAL the solution is:

```
FUNC ROUND(X) CLOSED  
RETURN INT(X*100+.5)/100  
ENDFUNC
```

Well, I said it was easy!

THE GREAT ICPUG COMPETITION

by Mike Todd (who else? - Ed.)

Well, it's here at last - a chance to win one (or more!) books.

The problem with a competition in a group like ICPUG is the range of abilities and interests of its members. So, I've had to put something together that all members will have a reasonable chance of winning. As a result, the competition is unusual in that the winners are not likely to be those with all the answers right, although it'll help.

Answer as many questions as you can on the following two pages, write them on a piece of paper (or photocopy the pages & write the answers on this) and send them to me, Mike Todd, at:

27 Nursery Gardens
Welwyn Garden City
HERTS AL7 1SF

and PLEASE put a letter C in the top left corner of the envelope. Entries should arrive no later than 12th January and all winners will be notified at the end of the month.

Which book(s) can you win? Well the selection is as follows:

- A - Inside Commodore DOS - for advanced disk programmers - includes disk of programs - all worth over £20.
- B - Dictionary of Computing (Wiley) - excellent reference which has nearly every computer term explained - hardback and worth £12.50
- C - Proper Basic (Wiley) - hard back general coverage book on aspects of BASIC programming - worth £12.50
- D - BASIC Adventures in Space 1: The Alien Planet - (Dr Watson) programming course for the VIC - inc cassette and worth £9.95
- E - BASIC Adventures in Space 1: The Alien Planet - (Dr Watson) programming course for the 64 - inc cassette and worth £9.95
- F - Beginners BASIC course for CBM 64 - (Dr Watson) - includes cassette with Honeyfold's HONEY-AID - worth about £10.00
- G - VIC-20 Assembly Language Course - (Dr Watson) - includes cassette and simple assembler - worth £12.00
- H - The Commodore 64 Handbook - (Century) - average sort of reference book for the 64 - worth £5.95
- I - VIC-20 programmers reference guide - from Commodore - £9-ish
- J - Key into your VIC-20 - (Foulsham) - Simple BASIC course - £5.95

Those are the prizes - but you'll have to indicate which ones you are interested in by putting a string of letters at the top of your entry indicating your order of preference on the above books.

I've tried to make the questions fairly universal, and many of the answers will be found in ICPUG newsletters over the last year or so.

If you write the answers on a separate sheet of paper, then please make sure you show which question you're answering by including its number, and don't worry if you can only answer a few, you still stand a chance of winning a prize - my (unbiased!) decision will be final - it's all really a bit of fun, so don't take it too seriously.

So to the questions:

If you came across the following lines, which machine (or language!) were they each written for and what do they do?

- 00 POKE 41229,80
- 01 POKE 808,239
- 02 POKE 781,5: SYS 60045
- * 03 ZONE 5
- 04 PRINT CHR\$(130);"ERROR"
- 05 PRINT#9,CHR\$(85);CHR\$(58)
- * 06 POKE 243,122: POKE 244,2
- * 07 DCLEAR U9
- * 08 POKE 59458,62
- 09 GET#2,AS: SYS61827
- 0A ON KEY"R",: CALL RESET
- * 0B SYS 64721
- * 0C SPRINK 1 TO 5,3
- 0D VDU 65
- 0E PRINT CHR\$(155)
- 0F POKE 642,16: POKE 644,30: POKE 648,30: SYS64824

Now, how well do you know your Commodore products? What products do the following product codes refer to (I've missed out prefix letters)?

- * 10 1525
- 11 2016
- 12 1910
- * 13 7501
- * 14 264
- * 15 1542
- * 16 802
- 17 1502
- * 18 1011A
- * 19 710
- * 1A 6581
- 1B 1101
- * 1C 1001
- * 1D 1531
- 1E 901227-02
- 1F 4030

That's 32 questions already! How are you doing? Some are quite tricky, but I don't think impossible. So, what about the following?

- * 20 Where is F Bowen?
- 21 I fumble jet dirt (anag.)
- * 22 What does GCR stand for?
- * 23 How many K of RAM in the 3032?
- * 24 His language is also a triartgle - who?
- 25 Jail racket, M? (anag.)
- 26 What was wonderful and cheap but became simple, at a greater price
- 27 (bytes of VIC ROM)+(bytes of VIC RAM)+(VIC processor)
- 28 Scab mission? (anag.)
- * 29 Who or what are Ted and Sid?
- * 2A When is 5*0.2 not 0.2*5?
- * 2B Which is the odd-man-out: 6509, 7501, 6502, 6802, 6510?
- * 2C Odd-one-out again: E843, 9112, E001, 0000?
- * 2D Which executes fastest: (1) A=A+1 or (2) A%=A%+1?
- * 2E Which of the two in 2D uses least memory?
- * 2F What is special about line number 351111?

The following two lines on a PET take a certain time to execute:

```
100 Q=50: B$=T1$
110 A=(2*Q)+0.417: A$=B$+"TEST"
```

Will each of the following modifications slow down, speed up or make no difference to the time these two lines take to execute?

- 30 Add line 10 X=50:Y=372.3
- 31 Change Q to Q% and A to A%
- 32 Delete line 100 and replace Q with 50 and B\$ with T1\$ in 110
- 33 Change line numbers to 10000 and 20000
- 34 Add line 10 C\$="TEST" and replace "TEST" in 110 with C\$
- 35 Put all on a single line 100
- 36 Run the two lines on a British 64
- 37 Remove brackets from 110
- 38 Add line 10 DIM A(100),Q\$(100,2)
- 39 Add line 10 X=2 and replace 2 in line 110 with X
- 3A Change B\$ in 110 to T1\$
- 3B Run the two lines on a 710
- 3C Change 0.417 to 417/1000
- 3D Change Q to QQ and B\$ to BB\$
- 3E Change line numbers to 1 and 2
- 3F Add spaces to make lines more legible

Finally - no, not quite! What, if any, are the problems that may occur in the following program lines:

```
40 0 GET T$: IF T$="" THEN
41 FOR A(4)=1 TO 100: N=N+1: PRINT 2*N: NEXT
42 T=T/100: IF T=0.87 THEN PRINT "ERROR": END
43 OPEN 2,8,2,"0:DATA,S,W": PRINT#2,NM$,AD$,PN$: CLOSE 2
44 D=INT(RND(-4)*6)+1
45 SAVE"0:PROGRAM,U",8
46 GOTO1000///THIS IS THE MAIN PROGRAM
47 PRINT 1.0000000000000000000000000000000000000000000000000000000
```

Now - finally, some general questions:

- 48 Where is Commodore's UK factory?
- 49 What is the name of the "head man" in the UK?
- 4A The PLUS4 is the production version of two other machines, what were they called?
- 4B About which micro did Jim Butterfield write his first book?
- 4C How many 64's have been produced in the UK (approx)?
- 4D What were the US and UK names of the 64 compatible machine that never appear on the market?
- 4E Uncle Jack decamped and bought ... who?
- 4F In what year did Commodore start business?

Well, that's it. Please don't forget to get your answers in as soon as possible and don't worry if you can't answer many. If you've got less than 10 right, then maybe you don't use Commodore machines or read ICPU! Between 10 and 30 ... not bad; 40-60 you're in with a chance of getting a prize; 60-80 you stand a better chance of getting a prize! If you get between 80 and 90 right then your name is probably Jim Butterfield! More than 90 ... you can set the questions next time!

Good Luck - answers next issue!

DISK FILE — SECTOR 11

by Mike Todd

MULTIPLE FILE COPYING

Did you know that the COPY command on the 4040 (and presumably other drives too) allows pattern matching? I certainly didn't until Nigel Peters mentioned that he'd discovered that it did.

For instance, if you want to copy all files beginning with the letters "PROG" from drive 0 to drive 1 so that they have the same names when they get there, then use the copy command as follows:

```
COPY D0,"PROG*" TO D1,"*".... in BASIC 4  
PRINT#15,"C1:*=0:PROG*" .... in other BASICs
```

This discovery allows, amongst other things, batches of files to be transferred - especially useful for those using Wordcraft files.

ILLEGAL RECORD LENGTHS

Brian Grainger mentioned in the September newsletter (p371) that a record length of 42 will not work. This is true. Neither will lengths of 58 or 63. These represent that ASCII characters * : and ? which the DOS fails to accept. I guess there's some odd bug within the command string parsing routines.

The values of 42 and 63 return 33,SYNTAX ERROR while 58 returns 30,SYNTAX ERROR.

RESETTING THE DRIVES

There are two very useful methods of resetting the Commodore disk drives. Both are sent down the command channel and the first forces a complete reset as though the drive had just been turned on. This command is:

```
PRINT#15,"U:" or  
PRINT#15,"UJ"
```

A second reset technique is slightly different, and generally resets the drive at the point AFTER the normal diagnostics have been

completed. This is done as follows:

```
PRINT#15,"U9" or  
PRINT#15,"U1"
```

I have read that these commands should not be used as they cause the drive and/or computer to crash. In my experience this is only because the commands were followed to soon by another disk access.

Inevitably, if the drive is in the process of resetting, it will not be in a position to handle any computer activity and will hang up. Therefore, a delay of at least a second is recommended, especially after "U:". A simple FOR loop from 1-1500 would suffice.

```
LOAD"*",8
```

When any Commodore drive is initialised, the LOAD"*" command will always load the first file on drive 0. Thereafter, LOAD"*" will load the last file accessed by the drive. Therefore, if a program has been using files it may be that a LOAD"*" will try to load the last file accessed - in other words a file type mismatch will occur.

In order to guarantee that the first file on the drive is loaded, the command can be change to LOAD"0:*" - the addition of the drive number makes the DOS perform a pattern match and this forces the first file in the directory to be loaded.

SPECIFYING DRIVE NUMBERS

I'm sometimes asked if the drive number should be explicitly stated when working with single drives. In theory there is no need since these drives automatically default to drive 0.

However, in keeping with my policy of never letting the drive set up its own parameters wherever possible, I would suggest that even with the 1541, drive 0 should be specified in all operations.

This principle should be applied to all aspects of disk operations, not just where the drive number is concerned. It is certainly not proven, but I believe that this precaution helps sort out one or two anomalies that have been found with single drives.

SPEEDING UP OF THE 1541

Though the serial connection is responsible for much of the 1541's lack

of speed, there are a couple of other factors that don't help.

One of these is the fact that, once the drive turns the motor on, it will be nearly 1.5 seconds before any reading or writing can take place. In addition, the slow speed of the head movement contributes to the overall delay in getting to the required track and starting the read or write operation.

All these timings are a direct function of a timer in one of the interface chips and it is possible to change the timing constant held in location \$1C06/\$1C07.

Ignoring the low order byte, the following command will change the timing constant to X:

```
PRINT#15,"M-W"; CHR$(7);CHR$(28);CHR$(1);CHR$(X)
```

The timer is initialised to \$3A00 (14848) which results in a timing pulse every 15mSeconds. Changing it to \$2000 (X=32) will make a significant difference in certain aspects of 1541 operation.

However, a couple of words of caution. Firstly, the same timer is used to determine how long it should be before the motor is turned off after the drive was last accessed. If set too short, it may be that the drive ends up being turned off too often, remembering that every time the drive motor is turned ON, there is a delay.

Also, the stepper motor which positions the head can only operate up to a certain speed and making it move too quickly will not only cause it to move incorrectly, but may also cause damage if it runs too long at the faster speed. A faster speed also means that the bumps will be heftier and likely to cause alignment problems quicker.

Therefore, this information is given mainly for experimentation although I have found that X=32 appears to be an acceptable value.

CARE OF DISKS

A recent MICRO-LIVE program on BBC demonstrated that disks are far more durable than we are led to believe. However, the tests performed by the programme were totally misleading and could well lull disk users into a false sense of security. They totally ignored the long term cumulative effects which can occur.

I would therefore suggest continued care!

4040/1541 COMPATIBILITY

Several times I have mentioned that disks written to on a 1541 drive should no longer be considered write compatible with the 4040. However, further investigations show that the problem is not as clear cut as it was originally thought.

One aspect of the problem is related to head alignment. If the head of the writing drive (which could be any drive, not just a 4040) is out of alignment, then writing a block on one track may well cause the partial erasure of the header block on another.

Header blocks are written once and for all when the disk is formatted and they never get re-written until it is formatted again. The blocks of data, on the other hand, are obviously re-written whenever a write occurs to that block.

Unlike normal magnetic tape recording, the disk head does not erase the old data before writing, it merely writes all over the top of it. This is perfectly adequate to obliterate data immediately under the head, but the possibility exists that old data at the edge may not get fully over-written, especially if there is a very slight misalignment. So, the head writes a wider track than necessary, and then erases the edges to reduce the track width - this is known as 'tunnel' erasure.

The problem arises when the drive doing the writing is badly out of alignment and this erasure encroaches on the header block of the next track and partially erases it.

This can even occur within the same drive over a period of time as the head goes out of alignment due to the bumps, as mentioned before.

The only real solution to this is to make sure the drive maintains its alignment and the second is to do regular back-ups onto freshly formatted disks.

The other cause of write incompatibility is the fact that the distance between headers and data blocks is different between the 4040 and 1541. So how does this cause problems?

This requires an understanding of how the 4040 and 1541 work with headers. Remember that characters are encoded as 10 bits, and these are written 10 at a time on the 4040 but only 8 at a time on the 1541.

The 4040 reads the 6 characters (60 bits) of the header block, waits 9 cycles (10 bits each) and then writes a sync. Because of slight timing delays, the actual gap is around 93 bits.

The 1541 actually has about 84 bits between the end of the header and the start of the sync sequence - slightly shorter than on the 4040.

Now, if we take a disk that has been written to on a 1541 and then write to it on a 4040, the 4040 will start to write its sync sequence after about 93 bits - 9 bits AFTER the point where the 1541 had written its sync sequence.

A sync is only detected after 10 consecutive 1's have been read, and these 9 left-overs would normally be no problem. However, if the bit preceding this left-over sequence also happens to be a 1, or the 4040 is running slightly faster than the 1541, the sync pulse could be written after 10 bits of the left-over 1's.

But of course, it's the END of the sequence of 1's (denoted by the first 0 following the sequence) which marks the point at which the drive will start to read data. Unfortunately, at the moment it starts to write the sync, the 4040 puts a spurious 0 on the disk - this is not a fault or a design error, just a fact of life.

This means that, if things aren't going perfectly right, the left-over 1541 sync sequence will be detected as such AND the start of the 4040's sync will be mis-interpreted as the start of the actual data.

The result is a DATA BLOCK NOT FOUND error.

So what, if any, are the solutions? Well, Commodore have introduced a modified 1541 ROM to increase the 1541 header gap by 8 bits.

Another is to modify the 4040 to write its sync pattern earlier. This can be done by altering location \$109D which contains the header gap length as follows:

```
OPEN 15,8,15
PRINT#15,"M-W"; CHR$(157); CHR$(16); CHR$(1); CHR$(8)
CLOSE 15
```

This should be performed only if it is necessary to write to a disk which has been written on a 1541. The 4040 should be reset before any further writing is performed.

On the 1541, this header gap length is fixed in ROM, but as I mentioned above, Commodore have modified it on later ROMs. To determine which of the 1541 ROMs you have, it is possible to examine the byte that determines the header gap length.

There are actually two relevant locations. One at \$F58D for use when

writing a block, and one at \$FCDO used during formatting. The value of the first of these can be checked using the following routine:

```
OPEN 15,8,15
PRINT#15,"M-R"; CHR$(141); CHR$(245); CHR$(1)
GET#15,A$
PRINT ASC(A$)
CLOSE 15
```

If the result is 8 then you have an early ROM and should not attempt to write to one of your 1541 disks on a 4040. However, if the value is 9 then there should be no problem.

It should be emphasised that the operating system in the 2031 drive is virtually identical to the 1541, but there have been no similar ROM revisions.

While on the subject of 1541 ROMs, three different versions of the E/F ROM have been identified to date. I intend to discuss these in a later DISK FILE, but for the time being, the ROMs so far identified are:

901229-02	Gap count = 8
901229-03	Gap count = 9
901229-05	Gap count = 9 & several bugs fixed

If anyone can identify any other ROM release for the 1541 I'd very much like to hear about it - for instance, where did ROM -04 get to?

--oOo--

700 COLUMN

by Mike Todd

TRANSFERRING PROGRAMS TO 8000 OR PET

In the last issue, Joe Griffin mentioned transferring programs from the 700 series onto an 8000.

The problem is that the load address for the 700 is actually location \$0003, which is in zero page, and the program will overwrite system variables on other machines - not too healthy an operation.

Even if the start address were ammended on the program file, the link addresses appear to cause problems when the 8000 (or other Commodore machine) attempts to re-link after loading.

Although Joe's program is one way of solving the problem, I found two other methods.

The first is very long winded and requires that the complete program be listed to a disk file as follows:

```
OPEN 2,8,2,"name,S,W": CMD2: LIST: CLOSE2)
```

and then merged back into the destination machine using one of the many merge techniques currently available.

An easier method is to make the 700 think its own BASIC starts at \$0401, load the program (which automatically relocates it) and then save the program again - this time it will have been saved from \$0401, just as it would on the other machines. This is what you do:

```
POKE 46,4: POKE 45,1:      - set start of BASIC pointer
BANK 1                    - select BASIC text bank
POKE 1024,0              - and make first byte =0
BANK 15                   - restore system bank
DLOAD"program"           - load the program, it relocates
DSAVE"newprogram"       - and save it
```

Restoring the 700 to normal is a matter of turning off and on again, or doing the same again but replacing the 4 & 1 in the first two POKES with 0 and 3.

THE 700 MEMORY MAPS

I don't intend to reproduce the memory map of the 700 series in the newsletter as it is perhaps of limited interest. In any case, machine code programming on the 700 is something of a chore as, at least for simple programs, there really is only 1k available from \$0400 to \$07FF.

Attempting to write machine code in any of the other banks is certainly possible, but requires many careful considerations. I have found that large programs are best written in BASIC, with machine code used only when absolutely essential.

Because of the banking operation, many zero page pointers are now three bytes long with the third byte containing the bank in which the pointer points. This doesn't appear to apply to the usual BASIC text and variable pointers as the system makes assumptions according to the machine.

Whilst on the subject of banking, the 700 comes in two varieties. The 256k version and the, commoner, 128k version (the 710). The banks are allocated as follows:

	<u>128k version</u>	<u>256k version</u>
BANK 0	not used	not used
BANK 1	BASIC text	BASIC text
BANK 2	All variables	Arrays
BANK 3	*	Simple variables & DOS
BANK 4	*	Strings
BANK 5-14	*	*
BANK 15	System memory	System memory

* indicates the bank is not present, but is available for expansion.

All POKEs and SYS go to BANK 15 unless the default bank has been changed using the BANK command.

I don't intend to cover much more about memory layout unless there is a demand - please let me know if you are using a 700 series machine and would like further details as this particular series has been notably neglected both by ICPUG and Commodore yet I consider it to be one of the best machines that Commodore have ever produced (apart from programming in machine code!). If nothing else, its keyboard is superb.

THE 700 SCREEN

The 700 screen editor is different to other Commodore machines and theoretically allows lines of up to 255 characters to be entered, although in practice lines appear to be limited to 160 character, even though the buffer is 256 characters long. It does this by automatically tagging a new wrap around line onto the previous line regardless of the number of wrap lines already linked.

However, it is possible to disable this feature and stop any wrap around occurring so that the screen behaves like the 8032 series and the line length is determined entirely by the current window width. This is done by a simple POKE:

```
POKE 923,64 - disable line wrap - like 8032
POKE 923,0 - enable it again
```

Note that some ESC sequences may disable the effect of this POKE.

CHR\$(131)

The character CHR\$(131) has something of a special use on the 700 series. It is effectively the CHR\$ equivalent of pressing SHIFT and the LOAD/RUN key and will invoke the same operation. In other words, PRINT CHR\$(131) in direct mode is exactly the same as SHIFT+RUN/STOP - that is DLOAD** followed by RUN. If DLOAD fails for any reason then RUN is encountered and the program in memory is executed.

If used in a program, the 700 remembers the character and only actions it when the program returns to direct mode through END, STOP or an error occurring. This has hidden dangers if an error occurs during program development when using the CHR\$(131) feature. As soon as the program stops, the command is executed and the program loaded will overwrite the current program in memory.

DCLEAR COMMAND

There appears to be an extra disk command not shown in the handbook of the 700. It is DCLEAR (with optional drive number) which appears to cause the drive(s) to be initialised.

That's all for this time. If you want more on the 700, drop me a line and let me know.

COMAL CORNER

by Brian Grainger

It has been quite some time since the last COMAL Corner and in that time many of you have taken the opportunity to get the latest version for the Commodore 64, i.e. 0.14. Many of you have written to say how much you like it and that it has freed you from the PEEKs and POKEs to get at the hires graphics and sprites.

The other major influence of 0.14 has been for you all to want the cartridge! In the notes I sent out with 0.14 I said the cartridge was expected in November. As with most Commodore dates this date has not been met, but the reason is rather unusual. The original intention was for the cartridge to be marketed through Commodore Denmark. However, demand in Germany for COMAL has been increasing and Commodore Europe felt that it should be marketed through their offices. Consequently, arrangements for shipping ROMs had to be changed. At the time of writing I believe the first ROM samples are on their way to Denmark for approval. What I do not know is when ROMs will arrive in quantity and what that quantity will be. I do know that demand in Denmark is high and 50 per cent of the initial batch was earmarked for the USA. At one time Commodore USA were leaving COMAL marketing to the US COMAL Users Group Ltd. but I understand that this is being reviewed again. The delay in the appearance of the cartridge has meant that I have not yet put together the COMAL News Sheet No.2 and the details I promised last July are not worth giving yet. It would only make you frustrated that you cannot get it!

Since last time I have written to the US COMAL Users Group Ltd. and they have kindly sent me the first four issues of their COMAL dedicated newsletter along with the disks of programs that go with the newsletter. I have permission to distribute their programs and publish their articles so when I have sorted things out a little bit there should be quite a bit of new stuff. As a taster, their latest newsletter was 56 pages and includes a fairly detailed memory map of COMAL version 0.14. More of that next time perhaps.

A number of you have been sending in some programs and I shall be collecting them together as an applications disk of COMAL programs. These will be distributed here in Europe and, of course, a copy will be

sent to the USA. Details will probably be available next time. Bert Millest has written some very good demonstration programs of various methods of sorting and these will form part of the disk.

One reader has pointed out an error in the notes that went with COMAL 0.14. The RND function for choosing a random integer value in a given range was incorrectly given as RND(n:m) where the range is n-m. This should have read RND(n,m). I do hope this hasn't caused too much head scratching.

Lots of you seem to be using COMAL with your various printers and plotters. Many of you seem to be having problems with facilities that require a secondary address. Let me clarify how to open a file to the printer with a secondary address other than the default.

The syntax of OPEN in its full form for printers:

```
OPEN FILE <num>,"",UNIT <dev>,<sa>,WRITE
```

Replace <num> with the file number but DO NOT USE 1 as COMAL uses that for itself.

Replace <dev> with the device number, usually 4 for the printer and 6 for the printer plotter.

Replace <sa> with the required secondary address.

File number 255 is rather special as this is the file number used when you SELECT"LP:"

One can print in lower case on the Commodore printers by making use of this.

First OPEN 255,"",UNIT 4,7,WRITE and then SELECT"LP:" will print in lower case. To return to upper case first CLOSE FILE 255 and SELECT"LP:" will print in upper case. I have not tried it but it may be possible to list to the printer plotter by using UNIT 6 in the print statement above. I have recently had a reader saying he cannot open a file to secondary address 2 to change the pen colour on the printer plotter as he would with BASIC. I have not solved this problem yet, because I do not have a printer plotter! If anybody knows how to change the pen colour using COMAL let me know.

One other problem with printers is that some of you are using the MX80

with suitable interface. In this case secondary address 128 MUST BE USED if the printer is to give sensible results.

How many of you noticed the mistake in my last 'Teach Yourself Programming' article, and I do not mean that it started on page 332 and not 322 as given in the contents! Line 30 on page 333 should have read: 0030 j=j div 60 rem tum jiffies into seconds

One more item of proof that COMAL is catching on is given in the Guidelines for Computing Studies by the Scottish Education Board. I quote from their Introduction to the programming course.

'It was felt that in this course one language should be used for all three levels - Foundation, General and Credit - and that language should be well structured. The use of procedures will be required from the outset to enable pupils to construct solutions to problems by building up short but powerful programs. These might involve the input, sorting and output of names, ages and other personal details. This immediately necessitates bi-directional parameter passing and the use of arrays as parameters. It is felt that COMAL best fits the requirements and, from 1986, IT WILL BE THE SOLE LANGUAGE USED.'

The capitals are mine but it is clear that the Scottish Education Authority have seen the light. Praise be the day the English authorities feel the same way. It is interesting to note that the so-called Structured BASIC of the BBC Micro does not meet the requirements above on two counts. Firstly, bi-directional parameter passing is not possible and, secondly, arrays cannot be used as parameters!

Let me finish this article by providing some procedures to draw circles, lines and arcs as well as some to undraw them! They come courtesy of Peter Smith and the listing is not indented to save newsletter editing.

```
9000 proc circle(x,y,r) closed
9010 an:=2
9020 setheading (270)
9030 moveto x,y+r
9040 for deg:=0 to 360 step an do
9050 forward ((r/57)*an)
9060 left (an)
```

```

9070 endfor deg
9080 endproc circle
9090 proc arc(x,y,r,sa,da) closed
9100 an:=2
9110 pi:=3.141593
9120 moveto x+(r*cos(sa/180*pi)),y+(r*sin(sa/180*pi))
9130 setheading (360-sa)
9140 for deg:=sa to sa+da step an do
9150 forward ((r/57)*an)
9160 left (an)
9170 endfor deg
9180 endproc arc
9190 proc ucircle(x,y,r) closed
9200 an:=2
9210 setheading (270)
9220 p:=peek(646)
9230 pencolor (peek(53281)-240)
9240 moveto x,y+r
9250 for deg:=0 to 360 step an do
9260 forward ((r/57)*an)
9270 left (an)
9280 endfor deg
9290 pencolor (p)
9300 endproc ucircle
9310 proc uarc(x,y,r,sa,da) closed
9320 an:=2
9330 p:=peek(646)
9340 pencolor (peek(53281)-240)
9350 pi:=3.141593
9360 moveto x+(r*cos(sa/180*pi)),y+(r*sin(sa/180*pi))
9370 setheading (360-sa)
9380 for deg:=sa to sa+da step an do
9390 forward ((r/57)*an)
9400 left (an)
9410 endfor deg
9420 pencolor (p)
9430 endproc uarc
9440 proc uplot(x,y) closed
9450 p:=peek(646)
9460 pencolor (peek(53281)-240)
9470 plot x,y

```

```

9480 pencolor (p)
9490 endproc uplot
9500 proc line(x1,y1,x2,y2) closed
9510 moveto x1,y1
9520 drawto x2,y2
9530 endproc line
9540 proc uline(x1,y1,x2,y2) closed
9550 p:=peek(646)
9560 pencolor (peek(53281)-240)
9570 moveto x1,y1
9580 drawto x2,y2
9590 pencolor (p)
9600 endproc uline

```

--oOo--

TEACH YOUR GRANDMOTHER 10

If you are printing a lot of information to the screen and wish to prevent the data scrolling off before the reader has a chance to absorb it then try this technique.

Keep a count of the number of lines that have been printed and test to see if you have a screen full. If you have then wait for a key to be pressed before continuing.

e.g.

```

10 FOR A=1 TO X
20 PRINT B$(A),C$(A):REM printout data
30 IF A/22 <> INT(A/22) GOTO60
40 PRINT"PRESS SPACE BAR TO CONTINUE"
50 GETA$: IF A$<>" " GOTO50:REM wait for space bar to be pressed
60 NEXT

```

This stops printing every 22 lines and waits until the space bar is tapped. You could just as easily use:

```

10 PRINT B$,C$
20 A=A+1
30 IF A/22 <> INT(A/22) etc,etc

```

Grandpa

64 LIBRARY

The following is a list of the contents of two new 64 library disks, numbers 7 and 8, which are now available to members.

Most notable of the programs from this list is Synth Sample, on disk 7. This was one of the many new things that Jim Butterfield brought with him to the PET Show this year and has to be heard to be believed.

As the number of 64 owners is still growing at a rapid rate, we are always interested in anything new that members may have written which may be added to the library. If you have anything that you think should be included please send it in to us (on disk or tape and with a brief description and any helpful instructions).

May I remind members to please observe the rules when applying for programs (see vol 6. no 3. p.212).

Dave Vorberg. 64 Software Organiser

Disk 7 - Graphics and Music.

LIST ME	Contains descriptions of the programs on disk.
SLIDESHOW	Loader program for Hires Screen Show.
RONNIE	A picture for the above program.
SLADY	" " "
TANK	" " "
BLITHER	" " "
UNCLE	" " "
GLOCKENFLUTE	" " "
A LITTLE PEACE	Song - A Little Peace.
BACH FUGUE	Tune - Quite long.
THE ENTERTAINER	Tune - The Entertainer.
FROSTY	Theme tune for Frosty.
RAINBOW	Song - With words and music.
SOUND EFFECTS	Create your own sound effects.
SYNTH SAMPLE	Plays different tunes chosen from a menu.
PRELUDE	Tune - Classical.
STARWARS MUSIC	Theme to Star Wars with graphics.
HAPPY BIRTHDAY	Song - With words and music.

Disk 8 - Utilities and Games.

LIST ME	Contains descriptions of the programs on disk.
NEWS EDITOR	Edits many word processor files.
80 COLUMN	Produces 80 column screen using hires.
READ ANY SEQ.	Allows sequential files to be listed to the screen.
FAST FORMAT	A quick disk formatter.
REL FILE PRG	Read, write relative files.
4MIN COPY INSTR	Instructions for 4-Minute copy.
4-MINUTE COPY	Fast disk copier.
LOAD ADDRESS	Finds the start address of programs on disk.
DISK MONITOR	Disk monitor.
40 ASM/MON.	Machine code for Disk monitor.
FALIGN CHECK	Checks alignment of disk drive.
UNSCRATCH	Unscratch a disk file.
SCRATCH	Scratch a disk file.
SORT DIRECTORY	Sorts disk directory into alphabetical order.
QUICK DISK	Fast disk program loader.
FAMILY ROOTS	Trace your family roots program.
DICE SPRITE	Displays dice sprite.
WORD PROCESSOR	A simple word processor for the 64.
SUPERTREK	Star Trek game with sound and colour.
TIPS	Tips on the 64 for the beginner.
CAVES	An adventure.
ASTRO PANIC	A shoot 'em up space game.
BOOT MATHS	Test your maths skills - with sound and graphics.
SPACE NIM	The game of NIM.
SOUND HELPER	Learn how to make use of the SID Chip.
BLACKJACK	A card game.
TWIN BAGELS	Bagels for two.
BATTLESHIPS	The game of battleships.
PENTOMINOS	The game of pentominos.
HIRES MOTION	Hires picture with the illusion of movement.
CLIFFY	Questions answered correctly make man jump a cliff
MINOTON	Track down the Minoton.

--oOo--

IMPROVED FORMAT SUBROUTINE!

Grandpa's letterbox was inundated with mail from members claiming to have the 'all-singing, all-dancing' formatting routine after 'Teach Your Grandmother 7' in the July issue. The following article is by Mike Hart whose routine came closest to fulfilling the claim.

Many routines have been published in the past to provide a way of 'formatting' numerical data so that it is rounded to the specified number of decimals and to ensure that the decimal points line up when it is printed in a column. Many of these routines are very long and tortuous and may slow the system down considerably if there are a lot of numbers to process. I therefore decided to write a routine (in BASIC) which would be as short and as economical as possible, which would approximate to the speed of machine code routines and would format fully even 'difficult' numbers such as exponentials.

In particular, the routine needed to:

- round both positive and negative numbers correctly avoiding the errors that are occasionally introduced when the CBM arithmetic routines process certain numbers - try to round 812.676144 by using the $\text{INT}(X*1000+0.5)/1000$ approach!
- process numbers less than ± 0.01 that would otherwise be expressed in exponential mode.
- put in leading zeros for values between +1 and -1 (e.g. to ensure that .7 is expressed as 0.7).
- add a fractional part of trailing zeroes to ensure consistency so that when working to 3 decimal places 2.3 will be expressed as 2.300 and 2 as 2.000, for example.

The routine presented here is effectively contained in three lines (i.e. lines 3-5) and assumes that whatever number one wishes to process has been copied into the variable Z. The other variables associated with the formatting subroutine all start with Z so that the programmer can avoid contaminating the rest of the program. Line 1 sets up certain default values but these may be changed in the course of the program if desired. The demonstration (line 10 onwards) is set up with three decimal places (Z3), a rounding factor of 1000 (Z4) and a 'field-length' of 9 (Z2). The string of padding blanks (Z1\$) (see note 1) can always be made longer if desired, and obviously the GOTO at the end of line 1 points to the normal start of the program. Notice particularly that Z4, the rounding factor, is specified EXACTLY - if you attempt a

short-cut such as $Z4 = 10 \uparrow Z3$ then the result may be internally stored in a slightly inaccurate form and this may introduce errors later on. This is due to the fact that exponentiation involves manipulating the logarithm of a number and some loss of accuracy is possible. A 'balancing factor' (Z6) is included to compensate for occasional failures to round exactly. The internal construction of the program is as follows:

LINE 3 Makes a rounded string of the number multiplied by the rounding factor. Notice that this works just as well for negative as for positive numbers. The 'balancing factor' (Z6) is necessary because the CBM interpreter does not Force a round before performing INT and one has to correct this deficiency. The balancing factor is the smallest that trial-and-error has demonstrated to be effective for both positive and negative numbers. If you wish to demonstrate the presence of the INT bug for yourself then try the following;

```
PRINT 123.4555*1000 + .5, INT(123.4555*1000 + .5)
```

Both should give 123456 but the INT gives 123455 due to the bug. The presence of the 'balancing factor' enables numbers such as 123.4555 and -123.4555 to round correctly to 123.456 and -123.456 respectively. If you do not mind the occasional inaccuracy caused by failure to round up, then you can cut out the reference to Z6 in line 1 and the whole of the term 'Z6*SGN(Z)' in line 3. This also has the advantage of speeding up the whole sub-routine by some 10% but personally I would rather sacrifice a little bit of speed for complete accuracy. (Incidentally, the PRINT USING routine in the COMMAND-O chip will fail to round a negative number such as -123.4555 to 3 decimal places correctly!)

LINE 4 is only called into play for numbers (positive or negative) that are less than 1 and require a leading zero to be inserted. The effect of Line 4 is to turn, for example, -.123 into -0.123 or 7E-03 into 0.007. Numbers that would normally be expressed in scientific notation get turned into 'normal' numbers by this line but a similar technique is not used for very large numbers, which generally constitute less of a problem. (Numbers larger than +/-999999.9995 are handled incorrectly by this routine and if you think your program may produce numbers out of this range then you should test for them before calling the routine and deal with them independently).

LINE 5. This line is one of the most critical in the whole subroutine. If we assume that a Z of 123.4555 has been converted to the string Z\$

of 123456 (in line 3) then this line inserts the decimal point in the correct place and pads to the left with blank spaces before RETURNing. It is obviously necessary that integers avoid this line altogether and that is why they are taken care of by the conditional statement at the end of line 3.

How fast is the routine?

Given that care has been taken to ensure that the routine is as accurate as possible, how does it compare with machine code routines for speed? In order to make meaningful comparisons, I undertook some trials that compared my BASIC PRINT USING routine with (a) the COMMAND-O chip PRINT USING routine in a BASIC4 4032 PET and also with (b) the PRINT USING routine given by Raeto West in 'Programming the PET/CBM'. The results are summarised in the table below:

Averages of 5 trials using 1000 numbers

PROGRAM	MACHINE	AV. TIME	No PER SEC
BASIC PRINT USING	C-64	0.0540	18.5
RAE WEST PRINT USING	C-64	0.0411	24.3
BASIC PRINT USING	4032 PET	0.0518	19.3
RAE WEST PRINT USING	4032 PET	0.0375	26.7
COMMAND-O PRINT USING	4032 PET	0.0472	21.2

The BASIC PRINT USING stands up pretty well to the machine code opposition! In the case of the Rae West routine, the routine works by truncation and does not round at all (you have to do this before the number is sent to the subroutine) nor will it attempt to process numbers expressed in exponential format (such as 1E-03) which severely limits its usefulness. As we have seen, the COMMAND-O PRINT USING also has some deficiencies and will also make a mess of some exponential numbers e.g. 1E-03 which is 0.001 emerges as -03.000 to 3 decimal places. All in all, the BASIC PRINT USING comes out well in the accuracy stakes and also formats and prints at some 19 numbers to the second which, as you can see, starts to get pretty close to the speed of the machine code routines.

The use of the PRINT USING routine

To incorporate this PRINT USING routine into your own programs it is best to type it in exactly as shown from lines 1-6 (although lines 2 and 6 are only to assist readability), including the use of the low line numbering. The reason for this is that line 1 defines variables at the very start of the program and when these values are called, the

internal routines do not have to search through other variables in order to find them. For the same reason, constants have been defined as variables, as this too speeds up the entire subroutine. These techniques are applicable to other BASIC programs as well where speed is important. If you know that you are not going to require integers then you can cut out the whole of the conditional statement contained at the end of line 3; this too will speed processing slightly.

To call the sub-routine merely copy whatever variable you wish to format into Z and then call the subroutine with GOSUB 3. You may consider that this is a slightly messy way to do it - why not use a user-defined function instead? As you might have guessed, the user-defined function takes quite a bit more time to process and therefore I chose the 'copy' method. The fully formatted number is returned as Z\$ for you to PRINT to screen, printer or disk as you wish.

If you wish to alter the number of decimal places in the course of the program then you need to alter the parameters of Z3 and Z4. To effect the change, make Z3 the number of decimal places required and Z4 the relevant rounding factor. For example, to round to two decimal places make Z3 = 2 and Z4 = 100 before the subroutine call. These values remain in effect until you change them again.

If any readers have come across a BASIC PRINT USING routine that formats to the same level of accuracy and is faster, then I would be most interested to hear of it.

```

1 Z$="":Z=0:Z3=3:Z2=9:Z4=1000:Z5=5:Z6=0.0000307:Z1$="":GOTO10
2 :
3 Z$=STR$(INT(Z*Z4+Z5+Z6*SGN(Z))):IFZ3=0THEN
  Z$=RIGHT$(Z1$+Z$,Z2):RETURN
4 IF ABS(Z)<1 THEN Z$=LEFT$(Z$,1)+RIGHT$("000"+MID$(Z$,2),Z3+1)
5 Z$=RIGHT$(Z1$+LEFT$(Z$,LEN(Z$)-Z3)+"."+RIGHT$(Z$,Z3),Z2):RETURN
6 :
10 X=RND(0):N=16:C=1
20 FOR J=1TON:X=EXP(RND(1)*14-5)*C:Z=X:IF J<10 THEN PRINT " ";
30 PRINTJ,X;SPC(18-LEN(STR$(X))):A=TI:GOSUB3:PRINTZ$;:
  B=B+(TI-A):PRINT
40 C=C:NEXT:PRINT
50 AV=B/(N*60):PRINT "AVERAGE=";AV;" SECS"
60 END
70:
100 REM Z = ORIGINAL NUMBER TO FORMAT
110 REM Z$ = OUTPUT STRING

```

```

120 REM Z2 = FIELD LENGTH
130 REM Z3 = NO. OF DECIMAL PLACES
140 REM Z4 = ROUNDING FACTOR
150 REM Z5 = HALF ADJUST (0.5)
160 REM Z6 = BALANCING FACTOR
170 REM Z7 = STRING OF PADDING BLANKS (see note 1)
180:
190 REM 3 DECIMAL PLACES BY DEFAULT
200:
210 REM FOR DIFFERENT FORMATS MAKE:-
220:
230 REM Z3 - DECIMAL PLACES REQUIRED
240 REM Z4 - RELEVANT ROUNDING FACTOR
250 REM E.G.
260 REM Z4 = 1 FOR INTEGERS
270 REM Z4 = 10 FOR 1 DECIMAL PLACE
280 REM Z4 = 100 FOR 2 DECIMAL PLACES
290 REM Z4 = 1000 FOR 3 DECIMAL PLACES
300:
310 REM E.G. FOR 2 DECIMAL PLACES
320 REM Z3=2:Z4=100:Z=X:GOSUB3:PRINTZ$

```

Note 1.

Ken Rose also wrote in on the subject of formatting and makes the valid suggestion of using shifted spaces as padding. The BASIC interpreter treats shifted spaces differently from normal spaces and will therefore not remove 'leading blanks' used as padding when using INPUT to read formatted data from screen or disk.

HGV

--oOo--

C 64-SPECTRUM EMULATOR

In the July Newsletter, under 'Shop Window', an incredible product was mentioned which supposedly allowed Spectrum (that's twice in one issue! - Ed) programs to run on the 64. Surprise, surprise but this product has failed to materialise and Video Vault, who intended to market it, have undertaken to refund monies sent to them. They are still optimistic that it will appear and are keeping a file of interested persons. We are indebted to Mr DR Corbett, a member from Maidstone, for this information.

NATIONAL COMMITTEE FOR 1985

David Annal ACC REPRESENTATIVE AND 64 LIBRARIAN SOUTH
Tim Arnot REGIONAL ORGANISER & REGIONS SOUTH
John Bickerstaff DISCOUNT OFFICER
Alan Birks 4040 SOFTWARE LIBRARIAN NORTH
Harry Broomhall COMMUNICATIONS
David Broughton SECRETARY
Jack Cohen MEMBERSHIP SECRETARY
John Collins CBM LIAISON
Tom Cranstoun ASSISTANT EDITOR SUPERBASE
Hugh de Glanville NEWSLETTER EDITOR
Terry Devereux ARTISTIC ADVISOR
Rod Eva ASSISTANT MICRONET EDITOR
Ron Geere ASST EDITOR, TECH ADVSR & ASST REGNL ORGNSR. ALSO FARNBOROUGH REGM
Jennifer Goldsmith TREASURER
Brian Grainger COMPUNET/PRESTEL EDITOR & COMAL REP. ALSO HITCHIN REGION
Joe Griffin 4040 & 8050 LIBRARIAN, (4040 SOUTH)
David Harrow PUBLICITY AND ADVERTISING REVENUE
Derek Hoare VIC DISCOUNT OFFICER
Jeff Jones ASST NATIONAL REGIONAL ORGANISER(NORTH), ALSO MERSEYSIDE REGION
David Jowett 64 LIBRARIAN (DISK AND CO-ORDINATION), ALSO BLACKPOOL REGION
Alistair Kelman LEGAL CONSULTANT
Joe Kennedy EXHIBITIONS ORGANISER
Will Light PRESS OFFICER, ALSO COVENTRY REGION
Alfred Minter SOFTWARE LIBRARY ORGANISER
David Murray ASSISTANT DISCOUNT OFFICER.(Postal queries only).
Fred Offler BASIC & MC COURSE ORGANISER
Stephen Rabagliati ACC PRESTEL LIAISON, ALSO WATFORD REGION
Peter Reynolds ASSISTANT EDITOR & COMPENDIUM EDITOR
Mick Ryan CHAIRMAN, ALSO SOUTH EAST REGION
Mike Todd VICE-CHAIRMAN AND TECHNICAL ADVICE CO-ORDINATOR
Henry Velleman DEPUTY EDITOR
Dave Vorburg 64 SOFTWARE LIBRARY ORGANISER
Raeto West STRATEGIC PLANNING & SOFTWARE ADVISOR
Brian Wise VIC LIBRARIAN AND DISKETTES ASSISTANT DISCOUNT OFFICER

REGIONAL GROUP CONTACTS

Berkshire , Bracknell

Mrs Lewin , (Organiser.) , 9 Windlesham Road , BRACKNELL , Berkshire RG12 1JX
Tel: Bracknell 0344-423115 (home)
Meets: Based on local Church.

Berkshire , Slough

Mr B Jones , (Organiser) , 53 Beechwood Avenue , WOODLEY , READING , Berkshire RG5 3DF
Tel: Reading 0734-661494 (home) Slough 0753-34585 Ex 81 (work)
Meets: Slough College of Higher Education, Wellington Street, Slough. 2nd Thursday. 1930

Buckinghamshire , Aylesbury

Mr K H Knight , (Organiser) , 22 Mount Street , AYLESBURY , Buckinghamshire HP20 2SE
Tel: Aylesbury 0296-5181 (home)
Meets: Mandeville Centre, Aylesbury Every Friday 1930-2130

Cambridgeshire , Peterborough

Mr A P J Scott , (Secretary) , 18 Coventry Close , Werrington PETERBOROUGH , Cambridgeshire PE4 6NN
Tel: Peterborough 0733-74819 (home) Peterborough 0733-53161 Ex 381 (work)
Meets: Rm A24, Bretton Woods School, Peterborough Every other Wednesday 1900-2200

Cheshire , Macclesfield

Mr M Mayers , (Organiser) , 46 Green Street , MACCLESFIELD , Cheshire SK11 1JQ
Tel: Macclesfield 0625-32852 (home)
Meets: Fermain Club, Oxford Road, Macclesfield Every other Monday 2000-2230hrs

Cheshire , Stockport

Mr P J Hogan , 3 Greenhalgh Street , Heaton Norris , STOCKPORT SK4 1SR
Tel: n/a
Meets: * (Ask Contact)

Cleveland , Cleveland

Mr L P Butler , (Organiser) , 14 Wolsingham Drive , THORNABY Stockton on Tees , Cleveland TS17 9DW
Tel: Stockton on Tees 0642-762526 (home)
Meets: * (Ask Contact)

Clwyd , Connah's Quay

Mr J Poole , (Organiser) , 6 Ridgeway Close , CONNAH'S QUAY , Clwyd CH5 4LZ
Tel: Deeside 0244-815277 (home)
Meets: RAF Sealand + 6 Ridgeway Close, Connah's Quay Wed + 2nd & 4th Tuesday 1630-1930

Clwyd , Rhyl

Mr F Jones , (Organiser.) , 77 Millbank Road , RHYL , Clwyd
Tel: Rhyl 0745-54820 (home)
Meets: * (Ask Contact)

Clyde Region , Glasgow North

Mr A Morgan , (Organiser) , 19 Craigmarnloch Avenue , Balmoral Park TORRANCE , Glasgow, Clyde G64 4AY
Tel: n/a
Meets: * (Ask Contact)

County Antrim , Belfast

Mr D Weddell , (Organiser.) , 9 Upper Cavehill Road , BELFAST , BT15 5EZ
Tel: Belfast 0232-711580 (home)
Meets: Last Wednesday.

County Antrim , Northern Ireland

Mr D Bolton , (Organiser.), 19 Carrickburn Road , CARRICKFERGUS , County Antrim BT38 7ND
Tel: Carrickfergus 09603-63788 (home) Carrickfergus 09603-69129 (work)
Meets: * (Ask Contact)

Cumbria , Kendal

Mr D R Walker , (Contact.), 7 Kent Park Avenue , KENDAL , Cumbria LA9 5JT
Tel: Kendal 0539-22863 (home)
Meets: * (Ask Contact)

Derbyshire , Derby

Mr R N Davies FBCD, DCLP , 105 Normanton Road , DERBY , Derbyshire DE1 2GG
Tel: Derby 0332-514016 (home) Derby 0332-41025 (work) Systel No: 332514016
Meets: Derby Professional Colour, Osmaston Works, Osmaston Rd, Derby. 2nd & 4th Tuesday 1900

Devonshire , Torquay

Mr M Stibbe , (Organiser.), The Lawn , Lower Woodfield Road , TORQUAY
Tel: n/a
Meets: * (Ask Contact)

Dorset , Poole

Mr D M Shave , (Organiser.), 97 Canford Cliffs Road , POOLE , Dorset BH13 7EP
Tel: Bournemouth 0202-700428 (home)
Meets: * (Ask Contact)

Durham , Durham

Mr J Cocallis , (Organiser), 20 Worcester Road , Newton Hall Estate DURHAM , Durham
Tel: Durham 0385-67045 (home) Systel No: 219990467
Meets: 32 Old Elvet, Durham Every Thursday 1900-

Dyfed , Burry Port/Llanelli

Mr F Townsend , (Contact.), Maes-yr-Afon Pinged , BURRY PORT , Dyfed SA16 0JE
Tel: Kidwelly 0554-891065 (home)
Meets: * (Ask Contact)

Dyfed , Talybont

Mr S Kniveton , (Organiser.), Penpompren Hall , TALYBONT , Dyfed
Tel: Talybont 097086-303 (home)
Meets: * (Ask Contact)

Eire , Dublin

Mr J D Malone , (Organiser.), Software Village , 2 Leeson Park , DUBLIN 6
Tel: Dublin 0001-963488 (work)
Meets: 2 Leeson Park, Dublin

Eire , Dublin South

Mr T MacDonald , (Organiser.), 2 Sycamore Road , Mount Merrion , County DUBLIN
Tel: n/a
Meets: * (Ask Contact)

Essex , Basildon

Mr W W Green , (Organiser), 151 The Hatherley , BASILDON , Essex SS14 2QH
Tel: Basildon 0268-22430 (home)
Meets: * (Ask Contact)

Essex , Cattle Point

Mr M A Cooper , (Chairman), 12 Newpark Road , South Benfleet , Essex SS7 5UT
Tel: South Benfleet 03745-52885 (home) South Benfleet 03745-52885 (work)
Meets: Cattle Point Commodore Comp Club, Richmond Hall, High St, South Benfleet Every other Monday 1900-2200

Essex , Chelmsford

Mr J G Ross , (Organiser), 38 Quilp Drive , CHELMSFORD , Essex
Tel: Chelmsford 0245-441948 (home) 01-637-9111 Ex 2118 (work)
Meets: * (Ask Contact)

Essex , Colchester

Mr D Day , (Organiser), 11 Park Vale Close , CASTLE HEDINGHAM , Essex CO9 3DS
Tel: Heddingham 0787-60148 (home)
Meets: * (Ask Contact)

Essex , Ford Dagenham

Mr R C Cooper , (Secretary.), 117 Belmont Road , ERITH , Kent DAB 1LF
Tel: Erith (03224)984-37448 (home) 01-592-3000/2189/2898 (work)
Meets: Thames Foundry Bldg,Thames Avn,Ford Motor Co.,Dagenham. 6 weekly on Monday. 1400

Gloucestershire , Cam

Mrs J Rich , (Organiser.), Rose Cottage , 20 Old Court, Springhill , CAM QL11 SPF
Tel: Dursley 0453-47708 (home)
Meets: Rose Cottage, 20 Old Court, Springhill, Cam. Last Friday.

Gloucestershire , Cheltenham

Mr E Syrat , (Secretary.), 14 Branch Hill Rise , Charlton Kings CHELTENHAM , Gloucestershire GL53 9HW
Tel: Gloucester 0242-528678 (home)
Meets: Cheltenham Ladies College, Archway Entrance, St George's St Last Thursday. 1930

Hampshire , Basingstoke

Mr M Bell , (Organiser), 250 Gershwin Road , BASINGSTOKE , Hampshire RG22 4HL
Tel: n/a
Meets: * (Ask Contact)

Hampshire , Farnborough

Mr R Geere , (Organiser.), 109 York Road , FARNBOROUGH , Hampshire GU14 6NQ
Tel: Farnborough 0252-542921 (home) Camberley 0276-63377 Ex 4281 (work) Systel No: 252542921 rdg
Meets: 235 High Street, Aldershot. 3rd Wednesday

Hampshire , Gosport

Mr B Cox , (Organiser.), 10 Staplers Reach , Rowner, GOSPORT , Hampshire PO13 0EY
Tel: Fareham 0329-280530 (home)
Meets: Gosport Community Association,Bury House, Bury Rd,Gosport 1900

Hampshire , Petersfield

Mr A J Cooke LRSC , (Organiser,Reg Prestel Ed), 7 Russell Way , PETERSFIELD , Hampshire GU31 4LD
Tel: Petersfield 0730-62840 (home) Hazelmere 0428-55569 (work) Systel No: ajc
Meets: Meets at Bedales School, Petersfield. 2nd Tuesdays. 1930

Hampshire , Portsmouth

Mr N J Ryan , (Organiser.), 55 Portchester Road , North End PORTSMOUTH , Hampshire PO2 7HZ
Tel: Portsmouth 0705-699871 (home) Portsmouth 0705-827681 (work) Systel No: 732453530 njr
Meets: Portsmouth Polytechnic

Hereford&Worcester , Stourport

Mr M J Merriman , (Organiser.), 12 York Street , STOURPORT-ON-SEVERN , Worcestershire
Tel: n/a
Meets: 12 York Street,Stourport-on-Severn. Last Thursday.

Hertfordshire , Hitchin

Mr B D Grainger , (Organiser.), 73 Minehead Way , STEVENAGE , Hertfordshire SG1 2HZ
Tel: Stevenage 0438-727925 (home) Systel No: 438727925 bdg
Meets: Provident Mutual Assurance, Willian Road, Hitchin. Last Wednesday. 1930-2230

Hertfordshire , Watford

Mr S Rabagliati , (Organiser.), Suite 2, Tarmay House , 146-150 St Albans Road , WATFORD
Tel: Hemel Hempstead 0442-44743 (home) Watford 0923-50161 (work) Systel No: 01993223
Meets: Co-op Hall, St Albans Road, opposite junction with Lowestoft Road. 2nd Monday. 1930

Iceland , Iceland

Mr G Gislason , (Organiser), Syntax , Bleiksarhlid 4 , 735 Eskifjordur
Tel: n/a
Meets: * (Ask Contact)

Isle Of Man , Isle of Man

Mr J W F Livsey , (Organiser), Datacentre Ltd, 4 Athol St , DOUGLAS , Isle of Man
Tel: 0624-25538 (work)
Meets: * (Ask Contact)

Kent , Canterbury

Mr R Moseley , (Secretary), Rosemount , Lord Romney Hill, Weaving , MAIDSTONE
Tel: Maidstone (0622) 96-37643 (home) Maidstone (0622) 96-37643 (work)
Meets: Physics Lab, Kent University. 1st Wednesday. 1945

Kent , South East

Mr D Pettet , (Membership Secretary.), 69 Charterhouse Road , ORPINGTON , Kent BR6 9EN
Tel: Orpington (0689) 94-23505 (home) 01-828-8070 Ex 2303 (work) Systel No: 068923505
Meets: Charles Darwin School, Jail Lane, Biggin Hill. 2nd, 3rd, & 4th Thursdays. 1930-2130

Lanarkshire , Bellshill

Mr W G McLuckie , (Organiser.), 15 Roman Drive , BELLSHILL, Nr Motherwell , Lanarkshire ML4 2JQ
Tel: n/a
Meets: * (Ask Contact)

Lancashire , Blackpool

Mr D W Jowett , (Organiser.), 197 Victoria Road East , THORNTON , Blackpool FY5 3ST
Tel: Blackpool 0253-869108 (home) Blackpool 0253-56635 (work)
Meets: New Venue to be arranged.

Lincolnshire , Boston

Mr R Ashton , (Chairman), 17 Coningsby Close , BOSTON , Lincolnshire PE21 8HP
Tel: Boston 0205-66115 (home) Boston 0205-61122 (work)
Meets: Railway Staff Association Club, Boston Railway Station 1st Tuesday 1930-2200

Lincolnshire , RAF Scampton

Mr C Smith , (Organiser.), 5 Cornwall Close , LINCOLN , Lincolnshire LN1 2UH
Tel: Lincoln 0522-730070 (home) Lincoln 0522-730421 Ex 313 (work)
Meets: Education Centre, RAF Scampton. Weekly, Monday and Tuesday. 1900

London , Acton

Mr K V Meehan , (Organiser), 236 Chesterton Court , ACTON , London W3
Tel: 01-993-5660 (home) Systel No: kum
Meets: Work

London , Battersea

Mr S Bransgrove , (Organiser.), 31 Mysore Road , BATTERSEA , London SW11 5RY
Tel: n/a
Meets: * (Ask Contact)

London , Maida Vale

Mr M M Franks , (Organiser.), 196 Castellain Mansions , Castellain Road Maida Vale , LONDON W9 1HB
Tel: 01-286-0650 (home) 01-493-5141 (work) Systel No: 012860650
Meets: 196 Castellain Mansions, Maida Vale Thursday 9Aug84 1930-

London , MOD Branch, RAFPCA

Wg Cdr C R Deeley RAF , (Chairman), 50 Bedmond Road , Leverstock Green , HEMEL HEMPSTEAD HP3 8LL
Tel: Hemel Hempstead 0442-55110 (home) 01-218-0988/0546 (work)
Meets: Northumberland House, Northumberland Avenue, LONDON Mid-week, monthly 1245-1345

London , Stanmore

Mr G M Harris , (Organiser), 27 Silverstone Way , STANMORE , Middlesex HA7 4HS
Tel: 01-954-4548 (home)
Meets: Bernay's Hall, Stanmore Broadway Every 3rd Thursday 1900-2200 hrs

London , Teddington

Mr G Squibb , (Contact.), 108 Teddington Park Road , TEDDINGTON , London
Tel: 01-977-2346 (home)
Meets: * (Ask Contact)

London , Wembley

Mr B Vyas , (Contact), 277 Ealing Road , WEMBLEY , London HA9 1EU
Tel: 01-998-3712 (home)
Meets: * (Ask Contact)

London NW9 , RAF Hendon

Sgt M Kendell , (Organiser), Education Centre , RAF HENDON Aerodrome Road , LONDON NW9 5JB
Tel: 01-202-6577 Ex 248 (work)
Meets: Education Centre, RAF Hendon Thursday 1900

Merseyside , Liverpool

Mr J Stout , (Organiser.), 6 College Avenue , FORMBY , Merseyside L37 3JJ
Tel: Formby 07048-74266 (home) Southport 0704-31388 (work)
Meets: Merchant Taylors' School for Boys, Great Crosby. 2nd Thursday. 1900-2300

Merseyside , Merseyside

Mr J P Jones , (Chairman.), 41 Virginia Avenue , LYDIATE , Merseyside L31 2NN
Tel: Liverpool 051-526-4813 (home) Liverpool 051-334-4090 Ex 222 (work)
Meets: Cardinal Allens Grammar School, Honeys Green Lane, W Derby. Alternate Wednesdays. 1930-2200

Middlesex , Greenford

Mr R D Green , (Secretary), 18 Legion Road , GREENFORD , Middlesex UB6 8XG
Tel: 01-578-2302 (home) 01-750-7262 (work)
Meets: Room 7, Greenford Community Centre, Oldfield Lane, Greenford Every Thursday 1900-2200 hrs

New Zealand , New Plymouth

Dr D R Black , (Organiser), Box 5087 , Western New Plymouth , Taranaki
Tel: n/a
Meets: * (Ask Contact)

Norfolk , Cromer

Mr J Blair , (Contact point.), 7 Beach Road , CROMER , Norfolk
Tel: Cromer 0263-512849 (home)
Meets: VIC20 contact point only.

Norfolk , Wretton

Mr P Petts , (Organiser), Bramley Hale , WRETTON Kings Lynn , Norfolk PE33 9QS
Tel: Stoke Ferry 0366-500692 (home) Stoke Ferry 0366-500846 (work) Systel No: 366500846
Meets: Regional contact.

North Yorkshire , Scarborough

Mr K J Prince , (Organiser), 59 Chantry Road , EAST AYTON Scarborough , Yorkshire YO13 9ER
Tel: Scarborough 0723-863737 (home)
Meets: Gibson's Hotel, Queen's Parade, Scarborough, North Yorkshire 3rd Tuesday 1930-2200

Northamptonshire , Corby
Mr P Ashby , (Organiser), 215 Lincoln Way , CORBY , Northamptonshire
Tel: Corby 05363-4442 (home)
Meets: * (Ask Contact)

Northumberland , Prudhoe-on-Tyne
Mr J R Christer , (Organiser), 16 Redwell Rd, Oaklands Estate , PRUDHOE-ON-TYNE , Northumberland NE42 5AP
Tel: Prudhoe 0661-35507 (home)
Meets: Kimberley Clark Ltd, Sports & Social Club, Prudhoe Mill

Nottinghamshire , Mapperley
Mr M Graves , (Organiser.), 8 Digby Hall Drive , Gunthorpe Road CEDLING , Nottinghamshire NG4 4JT
Tel: n/a
Meets: Arnold & Carlton College, Digby Avenue, Mapperley. Every Friday.

Nottinghamshire , Nottingham
Mr E D Harvey , (Organiser.), 68 Roseleigh Avenue , NOTTINGHAM , Nottinghamshire NG3 6FH
Tel: Nottingham 0602-608491 (home)
Meets: Castlegate Congregational Federation Building, Nottingham. 4th Monday. 1930-2230

Oxfordshire , Banbury
Mr J Temple , (Organiser.), Kibanda Rose Bank , BLOXHAM , Oxfordshire
Tel: n/a
Meets: North Oxon Technical College, Banbury

Oxfordshire , Witney
Mr & Mrs C Ross-Smith , (Organiser), Enghaven, School Road , FINSTOCK , Oxfordshire OX7 3BN
Tel: Ramsden 099386-697 (home)
Meets: * (Ask Contact)

Somerset , Somerset
Mr P Montague , (Organiser.), 17 Deane Drive Galmington , TAUNTON , Somerset TA1 5PF
Tel: Taunton 0823-75630 (home) Bristol 0272-777691 Ex 215 (work)
Meets: 12 Laxton Close, Taunton.

South Yorkshire , Sheffield
Mr R Wood , (Organiser.), 13 Bowland Crescent , Ward Green BARNSELEY , South Yorkshire SYU 5JS
Tel: Barnsley 0226-85084 (home) Chesterfield 0246-811585 (work)
Meets: University Building, Mechanical Eng., Mappin Street, Sheffield 1st Wednesday 1915-2100

Strathclyde , Ayr
Mr J Smith , (Organiser and Reg Co-ord), 19 Brewlands Road , SYMINGTON Kilmarnock , Strathclyde KA1 5RW
Tel: Kilmarnock 0563-830407 (home) Systel No: 563830407
Meets: Symington Primary School, Symington, Kilmarnock 1st & 3rd Wednesdays. 1900-2100

Strathclyde , Glasgow
Dr J MacBrayne , (Contact.), 27 Paidmyre Crescent , NEWTON MEARNS , Glasgow
Tel: Glasgow 041-639-5686 (home)
Meets: Regional contact, helps and directs to John Smith RN102.

Strathclyde , Glasgow
Mr A J Quin , Dept of Environment Studies , Glasgow Coll of Bldg & Printg , 60 N Hanover St G1 2BP
Tel: Glasgow 041-332-9969 (work)
Meets: * (Ask Contact)

Strathclyde , Paisley
Mr C Heath , (Organiser), 41 Thrushcraig Crescent , PAISLEY , Strathclyde PA2 6PP
Tel: n/a
Meets: The South End Action Centre, Stock Street, Paisley Alternate Mondays 2015-2230 hrs

Suffolk , Bury St Edmunds

Mr A Morris , (Organiser.), 30 Kelso Road , BURY ST. EDMUNDS , Suffolk IP33 2EL
Tel: Bury St Edmunds 0284-61870 (home) Bury St Edmunds 0284-69661 (work) Systel No: 28469661
Meets: The Coach House, Long Brackland, Bury St Edmunds. 2nd Wednesday 1930-

Suffolk , Mildenhall

Master Sergeant F Owens , (Organiser), Commodore Co-ordinator , Box 3241 RAF Mildenhall , Suffolk
Tel: Erriswell 063881-3481 (home) Systel No: 28469661
Meets: USAF Recreation Centre, RAF Mildenhall. 3rd Sunday. 1400

Warwickshire , Coventry

Mr W H Light , (Chairman.), 22 Ivybridge Road , Styvechale COVENTRY , West Midlands CV3 5PH
Tel: Coventry 0203-413511 (home) Coventry 0203-462634 (School) (work) Systel No: 203413511
Meets: Stoke Park School & Community College. 4th Wednesday. (Not Aug or Dec) 1900-2200. £4

West Germany , Berlin

Major E F Carrel , (Organiser.), Berlin ICPUG , 29 Signal Regt , BERLIN BFPO 45
Tel: Watford 0923-81209 (home) 010-49-30-309-4476 (work)
Meets: * (Ask Contact)

West Germany , Minden

Mr B Dempster , (Vice Chairman.), Vice Chairman , 664 Squadron AAC LAD , BFPO 29
Tel: n/a
Meets: * (Ask Contact)

West Midlands , West Midlands

Mr J Bowman , (Organiser.), 6 The Oval , ALBRIGHTON , West Midlands
Tel: No telephone (home) Albrighton 090722-2393 Ex203/517 (work)
Meets: Old Town Hall, Red Lion St, Wolverhampton. 1st & 3rd Thursdays. 1900

ODDS AND ENDS

by Brian Grainger

1) Raeto West, he of the PET bible fame, has written another book. Called 'Programming the VIC' it will no doubt become the VIC bible! Cost is 15.90 from Biblios Ltd., Star Road, Partridge Green, Horsham, W. Sussex RH13 8LD (see yellow insert - Ed.).

Review next time maybe.

2) The 64 bible by Raeto West is expected around March next year.

3) It may have been coincidence but I'm sure my MPS802 printer stopped printing when the paper inadvertently ran out when using Easyscript. If it does have a paper sensing mechanism I am impressed. It always worries me that if the paper runs out my print head gets bashed on the roller.

4) After all the rude comments about the MPS802 manual last time I am beginning to wonder if I have got a new edition. I find it readable, mentions all the facilities I know about (except that mentioned in 3)) and unlike Bert Millest's manual it does give the Unenhance control code as chr\$(15). It is copyrighted 1984, but no other clue to print date is given.

5) The Jim Butterfield disk, mentioned in September, not only includes some synthesiser music but also a program to play music from the keyboard. It includes some interesting slide effects as well.

--oOo--

1541 EXPRESS - REVIEW UPDATE

Following the review of the 1541 Express in the last issue of the newsletter we are very pleased to say that RAM Electronics have contacted us with news of two important updates. Firstly a change has been made to the operating system so that use of SHIFTEd RUN/STOP now loads the first program from disk using the format LOAD"*",8,1 (the ',1' does not apparently appear on the screen!). This is a very worthwhile change negating one of my major criticisms.

Secondly, and of equal interest, RAM are offering members a 10% discount on this item (we are trying to persuade them to offer a discount on their whole range, so keep an eye on Discount Corner). The discounted price is £45.85 including VAT and postage and packing. Cheques or Postal Orders should be made payable to RAM Electronics (Fleet) Ltd and sent with your name, address AND full ICPUg membership number to:

Richard Brown
RAM Electronics (Fleet) Ltd
106 Fleet Road
Fleet
Hants GU13 8PA

Telephone enquiries:- 02514 5858

HGV

DISCOUNT CORNER

Previous 1984 Discount Corners appeared on pages 62, 135, 204, 295 and 376. The article on page 121 gave general advice.

WRITTEN ENQUIRIES.

David Murray would welcome your written enquiries. Please send these to 8, Enticott Close, Tankerton, Whitstable, Kent CT5 3ET with a 13p or a 17p stamp, since David uses window envelopes.

TASKSET

This company will allow members 100 discount from any of their advertised products such as Gyropod, Pipeline, Jammin, Cosmic Convoy and Boso's Night Out all for the CBM64. The normal selling prices for tapes are £6.90 and £9.99 for disks. However, Cadcam Warrior will sell at £9.95 for tapes and £12.95 for disks. Please place your orders direct with Taskset Ltd, 13 High Street, Bridlington, YO16 4PR. Please quote your membership number when ordering and if you have any queries please contact Mr A.M. Nutter, telephone 0262 73798.

RAM ELECTRONICS.

See Henry Velleman's addendum about the 1541 Express (10% discount for members on direct ordering, post free).

DATALECT COMPUTER SERVICES LTD

This company will allow 15% discount from their standard repair charges (as listed below) for Commodore products. Their charges are based on the delivery and collection costs being paid for by members and that the cost of parts used and VAT should be added to the prices listed. STANDARD LIST PRICES are VIC20 £20.00; C64 £25.00; C2N Cassette deck £15.00; 1541 disk drive £35.00; all other disk drives £45.00; all other microcomputers and printers branded CBM (except 6400) £40.00. Please call Paul Dunn, Service Manager, before visiting; his 'phone number is Woking (04862) 25995 and address 12 Woking Business Park, Albert Drive, Woking, Surrey GU12 5JY.

Finally, please accept our wishes for a Happy Christmas and a Good Computing (CBM) New Year. From Brian, David, Derek and John the Discount Men.

--oOo--

READERS WRITE . . .

SAVE AND REPLACE BUG

Roger Priestly, 5 Park Cottages, Mayles Lane, Wickham, Hants writes:

The controversy over @-REPLACE rages on, and we are still no nearer a definite answer. All that can be said with certainty is that there are bugs in Commodore DOS. The existence and location of one of these bugs has been known for years, and it is inexcusable that nothing has been done about it, particularly when the solution is obvious and simple. I refer, of course, to the problem with SCRATCH and unclosed files. SCRATCH works by following the chain of sector pointers on the disc, freeing each used sector in the BAM. The problem with an unclosed file is that this chain is not correctly terminated, and can lead anywhere on the disc. Blocks will be freed until a correct termination is found, and this may well include part of an active file. Subsequent file creation will then overwrite the "free" part of this file, and chaos looms - SCRATCHing files corrupted in this way merely makes matters worse.

As I said earlier, the solution is simple. SCRATCH should test for the "unclosed file" condition, and either (a) refuse to touch them at all, or (b) simply delete the Directory entry, leaving the blocks allocated in the BAM. Option (b) would appear to create fewer problems. Either option will leave the disc untidy, and VERIFY is still needed, but at least valuable files will not be scrambled. PLEASE, Mr. Commodore, DO something. There is a simple test for screwed discs, using a dual drive. If the files of the suspect disc are COPYed to a newly formatted disc, and BLOCKS FREE on the copy is less than on the original, file problems exist. The converse is not necessarily true, unfortunately.

TURBO LOAD CASSETTES

J Kemp of 81 Sunderland Ave, Welling, Kent writes:

I have a CBM64 used in conjunction with a C2N cassette deck with which I was able to LOAD and SAVE programs with no problems at all until I bought a "TURBO LOAD" game. Then I was unable to LOAD every time, sometimes the Header would appear on my screen and once the game loaded.

I immediately suspected the cassette itself and had it changed, with no difference at all. Commodore suggested a faulty cassette player which luckily was still under warranty. This was changed and cleared the problem.

After speaking to other users of C2N cassette players it appears that the heads in the player can go out of alignment. They can be adjusted, although I would suggest a dealer to carry out the work as I understand the adjustments are critical and the cross headed screwdriver required is very small.

DO IT YOURSELF RS232 FOR THE 64

B Cutts of 12 Cedar Close, Meopham, Kent writes:

. . . it is possible to produce an RS232 adaptor for the C64 user port for about 5.00. It involves using only two chips (MC1488 and MC1489 line driver/receiver IC's) plus a couple of transistors and a few resistors and capacitors. Power for the circuit is obtained from the computer and the unit gives out proper +12v and -12v RS232 levels as well as converting down to +5v for the computer.

A suitable version of this circuit appeared in the August 1984 edition of PRACTICAL ELECTRONICS, and they also produced a PCB for the project.

For those who are interested, back numbers can be obtained for £1.00 from :- POST SALES DEPT (PRACTICAL ELECTRONICS), IPC MAGAZINES LTD, LAVINGTON HOUSE, 25 LAVINGTON ST, LONDON SE1 0PF (or try your local library - Ed).

--oOo--

MEMBERS' SALES AND WANTS

FOR SALE

Audiogenic MAGPIE 64 database (cartridge) complete package as new. Very Powerful as reviewed ICPUG newsletter March 1984. £59; JM Muir tel. 0292 74237/268410 evenings.

Wordpro 4 word processing package for 8032 PET with 8050 disk drives. Includes extensive manual, program disk and ROM. Current retail price £395+VAT, yours for only £100; B Cutts tel. 0474 812523.

Automatic and programable CB radio scanner. CB radio together with interface and program. Built for VIC 20 but could be altered. Features include simplex transmission. £65 (ring for further details); also cassette version of Simplicalc for VIC 20. £2.50; J Sykes tel. Eastbourne 32403.

CBM64 cassettes:- Bumping Buggies £4.50; Hustler £3.50; Packacuda £4.00; Acardia £4.50; Revise "O" + CSE Physics £6.50; all original and new, only 5 months old; M Amin tel. 021-326 7383.

Following items little used and in excellent condition:- Dual disk drive 4040. £350; CBM64 computer and 1701 colour monitor. £250 or monitor available separately for £125; DW Bennett tel. 0425 618253 (evenings)/04252 72511 (office).

Phillips TP200 monochrome (green screen) monitor, very good condition. £60 ono; L Faulkner tel. 0533 704676.

4040 Dual Disk Drive complete with IEEE to IEEE cable £400; R Patey tel. 0403 64027(home)/0403 812738(work).

WANTED

Program to convert Silicon Office text files to Wordcraft; J Sykes tel. Eastbourne 32403.

SOFTWARE LIBRARIANS

8050*
Joe Griffin
Clovelly
Lynwick Street
Rudgwick
Horsham
W Sussex RH12 3DJ

VIC-20
Brian Wise
17 Knighton Close
South Croydon
Surrey
CR2 6DP

*also responsible for 64/VIC programs on 8050 disks

CHIEF LIBRARIANS (and DISK copies) 64 LIBRARY

Northern UK

David Jowett
Windmill Electronics
335 Red Bank Road
Bispham, Blackpool FY2 0HJ

Southern UK

David Annal
142 Windermere Rd
London
SW16 5HE

CASSETTE COPIES

Sumames beginning with:

A B C

Fred Owens
PO Box 3241
RAF Mildenhall
Suffolk

D E F

William McLuckie
15 Roman Drive
Bellshill
Lanarkshire
Scotland ML4 2JQ

G H I

R Burns
48 Grasmere Rd
Royton
Oldham
Lancs OL2 6SR

J K L

Mike Hingston
1 Stafford Rd
Eccles
Manchester M30 9HN

M N O

Barry Whitelaw
25 Buckerills
Basildon
Essex

P Q R

Nigel Corry
4 Horsted Way
Rochester
Kent ME1 2XY

S T U

Raymond Medford
7 Thom Grove
Hale
Altrincham
Cheshire WA15 9AW

V W X Y Z

Bill Hull
Rocklea
Pilling Lane
Preesall
Blackpool, Lancs

