NATGUG NEWS

Volume 7 Issue 1 July 1985

OFFICIAL JOURNAL OF THE

National TRS-80 & Genie Users Group.

INFORMATION ON THE GROUP

Membership of the Group is by subscription to the Newsletter, which is published monthly. Membership details are obtainable from the Group Secretary. Membership of the Group is open to anyone with an interest in the TRS-80 range of microcomputers, and compatible systems such as the Video Genie.

Details of the Group accounts, and the constitution of the Group, are obtainable from the Secretary.

Members requiring assistance with problems related to the TRS-80/Video Genie may call the Secretary. An attempt will be made to put them in touch with a member who can help with the problem.

Workshops are arranged from time to time in various parts of the country.

Sub-groups exist in many areas. A list is provided in the Newsletter from time to time.

The Group maintains two software libraries (Models I and II) which are free to members. Library lists are obtainable from the Secretary.

For confidentiality reasons, the membership list is not generally available, but members may ask the Secretary for a list of members in their area, and mailshots to all members may be arranged.

Back numbers of the Newsletter are available from the Secretary.

Please send all contributions for the Newsletter to the Editor.

Chairman and Newsletter Editor:-

Leon Heller,
8 Morris Walk,
Newport Pagnell,
Bucks. MK16 8QD.
Tel: (0908) 613004

Secretary and Newsletter Publisher:-

Brian Pain,
24 Oxford Street,
Stony Stratford,
Bucks. MK11 1JU.
Tel: (0908) 564271

CONTENTS

Swindon Workshop 25/26/27 October for details of the lectures etc. please ring Bob Sparling on 0793-740762

EDITORIAL

I'm sorry about the delay in getting this issue and the last one out. We'll try to catch up over the next couple of months, provided I get some contributions.

I was clearing out some old magazines recently and came across the issue of Practical Computing in which my original letter volunteering to form a TRS-80 Users Group was published, back in January 1979. I mentioned in the letter that I had just upgraded my original 4K machine to 16K, by installing the RAMs myself. I think the chips cost me just over £100!

I've just been beta-testing Lattice C for the QL. The availability of this compiler for the QL is interesting, in that it is 100% compatible with the MSDOS Lattice C implementation, so I could, if I wished, develop progams that could be compiled and run on the Model 1000 and 2000, and the IBM PC, come to that. Lattice C is very close to Unix/Xenix C, so programs developed on the QL could also be ported to the Model 16, or is it called the 6000 now? C will be the main development language on both the new Atari ST and Commodore Amiga machines, by the way.

QUANTA (formerly IQLUG, we've changed the name) will be having a stand at the PCW show in September. I'll be on the stand most of the time, and any NATGUG members will be very welcome.

C afficionados will be interested to hear that Richard Hudson, an aquaintance of mine, is forming a UK C Users' Group. I'll keep you posted on developments.

My Z8 system described a couple of issues ago now has the capability of programming 2764 and 27128 EPROMs, and with a few changes, 27256 and 27512 devices, although I haven't tried the latter two types (they are rather expensive). The addition of two 30p CMOS chips is all that is needed, and some software. It connects to any computer with an RS-232 port. A state-of-the-art EPROM programmer for about £20 isn't bad!

A new lap-held machine, the Model 200, is available in the States. Like the Model 100, the 200 appears to be made by Kyocera. It has more memory than the 100, and a flip-up 40 X 16 screen. It also has Multiplan in ROM, making it very useful for businessmen on the move. It's a bit expensive at $999, in my opinion, for the UK market.

I've ported my intelligent terminal program written in C onto the QL. I got it going with a couple of hours work, which emphasises the portability you get with the C language.

Leon Heller

MEMBERS' LETTERS

The other month a member told me that he had been hunting for the supplier of Zen and only discovered my name following an article on assemblers in 80-Micro, so to put the record straight here's a summary of whats gone into Zen over the last few years, where to get Zen from and what the cost is.

Editor: This acts as a word-processor type of editor but restricted to single line editing. Commands include line-at-a-time and page-at-a-time single key forward and backward scrolling, block move, global search and replace, duplication of code, extraction of code from library files, read and write to disk in both ZEN and Edtasm format. The assembler accepts any mixture of upper and lower case for opcodes, operands, registers, etc but distinguishes between upper and lower for labels.

Monitor: A disassembler has been added to the existing dozen commands for Hex and Ascii display/modify, port I/O, byte and word search. The disassembler can accept 64 seperately identified data areas and outputs labelled source to the video, printer or internal text buffer.

Debugger: This has been equipped with single stepping of both ROM and RAM with full register display and disassembly of the current instruction. Any subroutines that do not require to be single-stepped can be set to be executed in real time thus allowing programs that contain disk I/O to be debugged with complete control.

Assembler: Early in 84 conditional assembly was added, a Basic program included to pre-process Macro statements and included was the ability to direct listing output to an Ascii disk file. Subsequently the following have been added:

BASE	Sets the Assembler's default number base.
DEFD or DD	For Define Data Bytes, eg DD 32," "=STRINGS(32," ")
DEFW or DW	Now takes multiple operands.
#	For .XOR. in arithmetic.
IFNOT	For conditional assembly.
IFNOTEQ	For conditional assembly.
PAGE	For new page in listing.
LONG	Allows full listing of data from DEFM, DEFD etc.
SHORT	Limits data listing to first four bytes.
GET filespec	Includes the code from another disk file.
CHAIN filespec	Chains another disk source file to the current one.
FILE filespec	Re-directs listing file from current to new filespec.

Finally if I could add that as well as the Model 1/3/4 package ZEN is also available for Z80 CP/M operating system and has been tested with both Montezuma and Tandy's CP/M Plus. Included with the CP/M version is a Basic program that converts 8080 code to Z80; some editing of the final file is necessary but 99% of the work is done.

The Model 1/3/4 package costs S44.50 and the CP/M one S44.50, but for any registered owner with one version wanting the other then the price is S24.50. Upgrades are available at S9.50 for owners of earlier releases. All from:
Laurie Shields Software, 151 Longedge Lane, Wingerworth,
Chesterfield. S42 6PR. Tel 0246 75568

When I'm not adding new things to Zen I seem to spend a lot of time finding out which of many copies is the latest version of whatever Basic program it was that I was working on before it had to be put to one side for another more important one. As an aid I now use the following program to identify difference. To use it save the programs under question in Ascii and then run "Compare/Bas". In case you've never save d in Ascii then the command syntax is:
SAVE "PROGNAME/BAS",A

```
100 ' Compare/Bas
110 CLEAR 10000: DEFINT A-Z
120 CLS
130 LINE INPUT"Filespec of 1st Ascii saved Basic program : ";F1$
140 LINE INPUT"Filespec of 2nd Ascii saved Basic program : ";F2$
150 OPEN"I",1,F1$: OPEN"I",2,F2$
160 GOSUB 1000: IF E1 THEN 500
170 GOSUB 2000: IF E2 THEN 510
180 IF B1S=B2S THEN 160
190 IF L1<L2 THEN LPRINT "(1) ";B1$: GOSUB 1000: GOTO 180
200 IF L1=L2 THEN LPRINT "(1) ";B1$: LPRINT"(2) ";B2$: GOTO 160
210 IF L1>L2 THEN LPRINT "(2) ";B2$: GOSUB 2000: GOTO 180
220 LPRINT"**  Problems  **":LPRINT "(1) "B1$: LPRINT"(2) ";B2$:GOTO160
500 LPRINT"End of ";F1$:GOTO520
510 LPRINT"End of ";F2$
520 CLOSE
530 END
1000 E1=EOF(1):IF E1 THEN RETURN
1010 LINEINPUT #1,B1$:L1=VAL(B1$):PRINT@384,CHRS(30);B1$
1020 RETURN
2000 E2=EOF(2):IF E2 THEN RETURN
2010 LINE INPUT #2,B2$:L2=VAL(B2$):PRINT@640,CHRS(30);B2$
2020 RETURN
```

Now for some problems from last month's newsletter (dated June but arrived in August)

Dosplus 4 & TRSdos 6.0 Memdisks. Bearing in mind that TRS6 is no more than a re-hash of LDOS one should not be surprised to find that if there's two ways of implementing something then you can guess who will adopt the slow, inefficient and illogical method. This is typified in their implementation of JCL handling. For reasons best known to themselves their JCL system only accepts line-input from disk and ignores single character input such as inkeyS. So the JCL file can be used to power up Scripsit but once there the JCL ignores all the keystrokes as they are requested by the machine code equivalent of inkeyS, and now we get the illogical part, except for the Break key, this is accepted and recognised as the abort JCL command so on exiting Scripsit then it gives job aborted. I'm afraid there's no solution except by patching Scripsit so that on exit it automatically executes a backup routine, but that would involve disassembling the whole of it to find a suitable patch area and thats no small job.

Marcus Rowland cannot have read Geoff's documentation thoroughly enough as I'm sure in there somewhere if a " PC=nn" command that allows control codes to be send to the printer without upsetting Scripsit's characters, lines and page counts. All that he needs to do is determine the Ascii values of the control codes and then use the " PC". Typically "*END" has the values 42 69 78 68 and so his contol line would be " PC=42,69,78,68", for anyone using my beefed up Scripsit it would be a bit more longwinded with " PC=42 PC=69 PC=78 PC=68"

The Alternate Source, 704 North Pennsylvannia Ave,
Lansing, MI 48906, USA, publish a book by Jack Decker on the Model 1

Rom routines fully explained and indexed, with a Model 3/4 supplement detailing the differences from the Model 1. An extremely useful book if you use Rom routines and costing about $25.

Also worth getting is a two-monthly or thereabouts publication by Jack Decker called Northern Bytes. Mostly machine code but containing a fair amount of general interest, the last one I received was 22 pages of closely typed information all of very useful and not obtainable elsewhere. (See Molimerx's last mailshot offering to photocopy and distribute copies of N.Byte's listing of 3/4 differences. I wonder if he got Jack's permission?). To enrol simply send your address and Access number to
Northern Bytes, c/o Jack Decker, 1804 West 18th Street,
Lot 155, Sault Ste Marie, Michigan 49783, USA.
Each issue costs $2.

Newdos 80 Version 2.0 does work on the Model 4 but naturally it has to be the Model 3 version not the Model 1. Unfortunately for Jon Silver the only legitimate way of getting a copy is to buy it and I believe Blandford Computers carry a stock. If he then gets Video4 from TAS he can have Newdos80 with an 80 x 24 video as well.

Laurie Shields

POKING CHARACTERS TO THE SCREEN ON MODEL 4

My first disappointment on converting from my Model 1 to a Model 4 was to find that I could not poke characters to the screen as I did on the Model 1.

Most of the programs I have written for use in my Hotel involve inputing Names and Addresses together with other relevant data in some form or other. My early attempts at programming used simple INPUT statements for this action. Naturally if it was realised that an error existed in a previous input line (ie the Name was spelt wrong when you were in the process of entering the address) you had to go back and retype entire entry again.

As my programming skills increased I was able to develop a type of "mini word processor" where the cursor could be moved to any entry line on the screen and characters inserted or overtyped in the previously typed line. This worked perfectly on the Model 1 and was used in many of my programms.

When I started to convert these programs to run on the Model 4 I was horrified to find that all the literature from TANDY for the Model 4 stated that you could not PEEK or POKE to the screen memory. It was only after I found an article in September 1984's 80 MICRO by Seth Monger on this very subject that I discovered that you can switch the Video memory into main memory - where you can PEEK & POKE to your hearts content - then switch it back to where it came from before you do anything else.

The following program is a demonstration of the Model 4 version of my input subroutine using the above technique. It involves POKEing a value of 134 into address 120 (port 132) to switch Video into Main Memory and a value of 135 to switch it back again. When debugging this program - if you press BREAK when Video switched to Main Memory, then type POKE 120,135 immediately afterwards - otherwise "funny things" happen. I find this subroutine ideal for my applications. Other members may be able to make use of it or adapt it to their own programs.

```
5 CLS:'           **** Model 4 Screen POKE Demo ****

6 CLEAR,-3073
8 GOTO 10
9 PG$="scnput13/bas":CLS:PRINT@400,"Saving Program":PRINT,PG$:SAVE PG$:
  PRINT,,"DONE":END:'
  NOTE:-
            CURSOR STAYS PUT WHEN SHIFTING TO RIGHT
            SHIFT LEFT DELETES AND MOVES UP TO CLOSE GAP

10 LC(1)=425:LN(1)=30:LC$(1)=" Name :- ":LT$(1)="ALPHA"
11 LC(2)=585:LN(2)=38:LC$(2)=" Address :- ":LT$(2)="ALPHA"
12 LC(3)=745:LN(3)=30:LC$(3)=" Town :- ":LT$(3)="ALPHA"
13 ' Put more print@ positions,  lengths  & labels here
14 LC(4)=905:LN(4)=2:LC$(4)=" Age :-":LT$(4)="NUM"
19 LC(5)=1260:'   this is position of instruction line
20 JJ=4:'   Set this to total No. of lines to Input
40 FOR J=1 TO JJ :LM(J)=63488!+LC(J)+LN(J):LS(J)=63488!+LC(J):NEXT
45 '
            ****   Enter data routine  ****

50 GOSUB 200:FOR J=1 TO JJ:GOSUB 100:NEXT:GOSUB 100:GOSUB 130
55 '
             ****  Rest of program here  ****

60 CLS:PRINT:PRINT"For use in rest of program:-":PRINT:PRINT:
   FOR J=1 TO JJ :PRINT"IP$("J") = "IP$(J):NEXT
70 PRINT:INPUT" press <ENTER> to edit the above strings
   or <Q> to Quit & Return to Disk Menu ";X$
72 IF X$="q"OR X$="Q" THEN RUN"direct0x/bas"
75 '
        ****  Routine to modify existing data ***

80 CLS:GOSUB 200:FOR J=1 TO JJ:PRINT@LC(J),IP$(J);:NEXT:
   FOR J=1 TO JJ:GOSUB 100:NEXT:GOSUB 130
81 '
            ***   Back to rest of program again  ***

82 CLS:PRINT"  Strings now changed to :- ":PRINT:PRINT
85 FOR J =1 TO JJ:PRINT" IP$("J")  = "IP$(J):NEXT
90 GOTO 70

100 REM   ::::::   INKEY INPUT ROUTINE :::::::
           :::::::::::::: For ::::::::::::::
             TRS-80 MODEL 4 (in MOD 4 MODE)
101 '          ***  With EDIT Facilities ****
102 '  ***   Modified from page 119  in 80 MICRO 5/1980  ***
     ** Plus Modifications from page 96 in 80 MICRO 10/1984 **
103 '               By :-  Ted Gladstone
                      031 661 4558
104 POKE 120,134:PRINT CHR$(15);:PRINT@LC(J),"";
105 X1=ROW(Y):X2=POS(Y2):Z1=63488!+80*X1+X2-1:LI(J)=0:Z=0
106 IF LT$(J)="ALPHA" THEN T2=127 ELSE IF LT$(J)="NUM" THEN T2=58
107 IF LT$(J)="ALPHA" THEN T1=31 ELSE IF LT$(J)="NUM" THEN T1=47
110 D=PEEK(Z1):POKE Z1,140:AL=A:SW=1
112 A$=INKEY$:IF A$="" THEN  POKE Z1,32:FOR I=1 TO 1:NEXT:POKE Z1,D:
    GOTO 110
114 A=ASC(A$):IF A=AL THEN SW=0:IF A = 13 AND J < JJ THEN POKE Z1,D:
    POKE 120,135:SW=1:RETURN
117 IF A = 13 THEN POKE Z1,D:POKE 120,135:RETURN
118 IF A > T1 AND A < T2 THEN POKE Z1,A:Z1 = Z1 + 1:LI(J)=LI(J)+1:
    IF LI(J)<LW(J) THEN LW(J)=LW(J) ELSE LW(J)=LW(J)+1:
    IF Z1 > LM(J) THEN Z1 = LM(J):POKE Z1,191
119 IF A = 25 AND LW(J)>LN(J)-1 THEN POKE Z1,D:GOTO 106
```

```
120 IF SW=1 AND  A = 25 OR A = 24  THEN  POKE Z1,D :LV(J)=LC(J)+LI(J)+63488!:
    R$="":R=VARPTR(R$):POKE R,LW(J)-LI(J):POKE R+1,LV(J)-INT(LV(J)/256)*256:
    POKE R+2,INT(LV(J)/256):IR$(J)=R$:SW=0:Z=0
121 IF A = 25  THEN Z=Z+1:LI(J)=LI(J):LW(J)=LW(J)+1:
    PRINT@Z1-63489!+Z," "+IR$(J);:GOTO 106
122 IF A=24 AND LI(J)<1 THEN POKE Z1,D:GOTO 106
123 IF A = 24  THEN Z1=Z1-1:LI(J)=LI(J)-1:LW(J)=LW(J)-1:
    PRINT@Z1-63489!+1,""+IR$(J)+" ";:GOTO 106
124 IF A = 9 THEN  POKE Z1,D:Z1=Z1+1:LI(J)=LI(J)+1:
    IF Z1 > LM(J) THEN Z1 = LM(J):POKE Z1,191
125 IF A = 8 THEN  POKE Z1,D:Z1=Z1-1:LI(J)=LI(J)-1:
    IF Z1<LS(J) THEN Z1=LS(J):LI(J)=0
126 IF A = 10 THEN  POKE Z1,D:Z1=63488!+LC(J+1):J=J+1:LI(J)=0:
    IF J > JJ THEN J = JJ :Z1=LS(J)
127 IF A = 11 THEN POKE Z1,D:Z1=63488!+LC(J-1):J=J-1:LI(J)=0:
    IF J<1 THEN J=1:Z1=LS(J)
128 GOTO 106
130 POKE 120,134:FOR J=1 TO JJ
132 POKE Z1,D: LV(J)=LC(J)+63488!:A=VARPTR(A$):POKE A,LN(J):
    POKE A+1,LV(J)-INT(LV(J)/256)*256:POKE A+2,INT(LV(J)/256):IP$(J)=A$
134 NEXT
136 POKE 120,135:RETURN

199 '              *** Screen    Layout   ***

200 FOR X% = 169 TO 1112 STEP 80:PRINT@X%,STRING$(58,191):NEXT
210 PRINT@263," Please answer the following :- ";
220 FOR J=1 TO JJ:PRINT @ LC(J)-(LEN(LC$(J))+3),LC$(J);:
    PRINT @ LC(J)-1,STRING$(LN(J)+1," ");:NEXT
230 PRINT@LC(J)-50,"  Press <ENTER> if OK else use arrow keys to fix  ";
235 PRINT:PRINT:PRINT TAB(15)"To Delete & Move up Press <SHIFT><LEFT ARROW>":
    PRINT TAB(15)"To Open Up for Insert Press <SHIFT><RIGHT ARROW>"
240 RETURN
```

The technique looks as if it could be useful for other purposes as port 132 also controls which memory bank is active - perhaps we could have another basic program running at the same time in the other bank. I do not have the time or experience to experiment with this at the moment but it looks interesting - perhaps some of our more knowledgeable members could comment.

Ted Gladstone.
Mansion House Hotel.
EDINBURGH
031 661 4558

HOPE FOR CONVERSION

In reply to Mr Rowland's article in last month's NATGUG NEWS I would like to give my experiences on converting from my faithful old TRS-80 Model 1 to a Model 4 in the hope that it may give any other members contemplating such a move, a little more reassurance than that which was obtained from Mr Rowland's article.

About a year ago I too bought a secondhand Model 4, my main objective in doing so was that I wanted an 80 column screen, all the bits in one box, and the ability to use all the existing software that I already had for the Model 1. This, I am pleased to report, I have achieved with very few exceptions. This is mainly due to the fact that I have used LDOS exclusively on the Model 1 and by using LDOS in the Model 3 mode, all my programs

run with no modification or conversion whatsoever (except for slight changes in commands to the printer). Even in the Model 4 mode I can read and write to all my Model 1 data disks. TRSDOS 6 is LDOS anyway so all the commands are familiar - although "SYSTEM" instead of "CMD" took a bit of finding - serve me right for not bothering to read the manual. (I may hasten to add that I am not interested in games programs so cannot comment on their compatability - all the above have been business programs and programs I have written myself in Basic.)

As to the use of external drives with the Model 4 I, naturally, wanted to use my old drives from the Model 1 and I too was advised by TANDY that these drives would not work on the Model 4. I was about to give up when I started experimenting and found that all that was required was to alter the stepping rate for the external drives with the "SYSTEM" command (Model 1 drives step at 40 ms whereas the Model 4 expects drives to step at 6 ms - therefore configure drives 2 and 3 to step at 40 ms).

ie :- SYSTEM(DRIVE=2,STEP=3)

This command works in Model 3 and Model 4 mode even if it's not in the Model 4 TRSDOS Manual. There is no need to alter any dip switches or anything within the external drives (contrary to the installation instructions for setting switches for disk drive number) or change the cable - merely plug into the Model 4 - configure the system as above - and hey presto you have 4 disk drives all active.

The problem Mr Rowland had with the printer cable is more baffling. I used an Epsom MX-100 with my Model 1 and this works perfectly, simply by plugging the same cable into the Model 4. It's also been hooked up to a DMP 120, a MX-80 and a TANDY Daisy wheel all belonging to friends who normally use them on their Model 1, so I guess it's Mr Rowland's cable that's at fault. The only incompatability, as stated above, is that certain line feed commands used on the Model 1 give unexpected results when used with the Model 3 but it is a simple matter to change these when running in the Model 3 mode.

The screen does give a very slight waver when the disk drives start up but, at the moment, nothing serious and barely noticeable. Up till now i've never had any I/O errors and it has never hung up - unlike the Model 1 that was forever hanging up until I fitted gold plugs to the expansion interface cable.

One final point, and one that I hope is exclusive to my machine, is that I have discovered that there is a fault in the printer driver routine within the ROM. This manifests itself by displaying "SYNTAX ERROR" or rebooting whenever an LPRINT or LLIST command is given when using the TRSDOS operating system in Model 3 mode or even when using it as a cassette based machine with no disk operating system. This only came to light recently when I tried to configure one of my LDOS disks in a certain way and I discovered all programs on that disk that contained any LPRINT statements caused a reboot. Up until that time the only program that would not work properly on the Model 3 was VIDEO4 - which should have given me an 80 column screen in the Model 3 mode. I put this down to incompatability with LDOS but at the recent Edinburgh Workshop I was told that it should work OK. Lawrie Shields and I burnt a lot of midnight oil after the Workshop and he eventually traced it down to a fault in the "C" ROM or possibly a wrong version of the "C" ROM in my machine. After many telephone calls to TANDY and elsewhere trying to find someone who could supply me with new ROMs - all to no avail - I have in the meantime given up trying to fix it. I don't want to

be without my machine while TANDY take it away to sort it out - they want to charge me double anyway because i've been inside to fit extra memory. It works perfectly with all the programs I am currently using (LDOS having its own printer driver routines) and when I eventually transfer all my programs to the Model 4 mode, these ROMS will be obsolete anyway.

Don't let the above problem put you off the Model 4 - nobody I spoke to at TANDY etc had ever heard of it before - it must be unique to my machine, otherwise I am delighted with it.

ps. I'm still keeping the Model 1 as a spare - just in case.

Ted Gladstone
Mansion House Hotel.
Edinburgh
031 661 4558

"ALLWRITE" and All That.

Both Dave Washford in the January News and John Miller in the May issue discussed "Allwrite" and indicated that there was some special U.S. legislation which prevented Prosoft from marketing this product in the UK. I also had this impression after my enquiries to Chuck Tesler had elicited a rather terse note saying that legislation prevented the sale of the product and I have even heard members saying that the export is banned by the US Government in case this excellent word processer falls into Russian hands !

The implications to TRS-80 users of having all the American supplies of software disappear overnight was shattering and I turned for advice to *"80 Alert"* the readers agony column run by the last of the TRS-80 magazines. The following is a verbatim quote from their reply.

> "There is no new US legislation preventing Prosoft from selling software internationally. Mr. Tesler chooses not to do so in many instances (there is no overseas shipping charge in the Allwrite ad) due to the fact that US software is not protected in most foreign countries including the United Kingdom. Mr. Tesler has proof of foreign companies selling copies of his programs and legally, there is nothing that he can do about it. As far as he is therefore concerned, as long as foreign laws protect the pirate, he will decide what he will and will not sell in these countries."

It appears, therefore, that the decision is very much a personal one on the part of Chuck Tesler. Whilst this restriction is very much operative for "Allwrite" which, to date, has not been available from any source other than Prosoft, it is not really true for other products marketed by Prosoft. I have recently purchased "Dotwriter 4.0" from an American discount house and would presume that fonts are also available from this source although I have not yet had written confirmation of this fact. Perhaps "Allwrite" will eventually find it's way into the distributive network and we will all be able to benefit from this, by far the best word processor available for the TRS-80 range.

"Allwrite" has been advertised with the Radio Shack catalogue number of 90-0242 for the Model 4 version. Is it possible that Tandy (UK) may actually market this product in the near future ?

Ken Arntsen,
3 Home Pastures,
Hatfield Heath,
Bishops Stortford,
CM22 7ES.

FILE ENCRYPTION IN C

The following program (for the C/80 compiler) encrypts a file by exclusive-ORing each byte with successive bytes from a key string. The file may be decrypted by running the program on the encrypted file using the same key, because of the symmetrical nature of the exclusive-OR operation. It won't stop the US National Security Agency or GCHQ looking at your files, or even a determined hacker, but it might be useful for business purposes.

```
/* crypt - encrypt and decrypt a text file */

£include <fprintf.h>

£define EOF -1

main(){

        int c, *in_chan, *out_chan, i, keylen;
        char source[12], dest[12], key[64];

        cls();
/* get source file name and open file */
        printf("Source file: ");
        gets(source);
        in_chan = fopen(source, "rb");
        if (in_chan == 0)
                error("File open error");
/* get destination file name and open file */
        printf("Destination file: ");
        gets(dest);
        out_chan = fopen(dest, "wb");
        if (out_chan == 0)
                error("File open error");
/* get key */
        printf("Key: ");
        gets(key);
        keylen = strlen(key);
/* get characters, encrypt them, and output them */
        for (i = 0; (c = getc(in_chan)) != EOF; i++)
                putc(c ^ *(key + (i % keylen)), out_chan);
/* close files */
        fclose(in_chan);
        fclose(out_chan);
}

error(s)
char *s;{

        puts(s);
        exit();
}
```

Leon Heller

CPM - A NEW USERS TUTORIAL PART 4

In part 4 of my tutorial is I will discuss the transient commands DUMP, LOAD, ASM and DDT. I must emphasize at this point that my knowledge of these commands is very limited as I am not deeply into Machine Code, however, I will do my best. I am also going to take them in a slightly illogical sequence as I shall deal with DUMP and LOAD before ASM, those of you already in the know will be aware that LOAD is used after a program has been assembled with ASM.

DUMP.COM

DUMP is very self explanatory, and simply dumps the contents of a file to the screen in Hexadecimal format. The command is simple to use and takes the form:

A>DUMP CHESS.COM
A>DUMP B:CHECKERS.COM

A listing of the file in HEX is sent to the screen, if a printout is required, then you should press CTRL P before executing DUMP. Here is an example of the printout from DUMP, in fact it is the first few lines of the DUMP.COM program

```
0000 21 00 00 39 22 15 02 31 57 02 CD C1 01 FE FF C2
0010 1B 01 11 F3 01 CD 9C 01 C3 51 01 3E 80 32 03 02
0020 21 00 00 E5 CD A2 01 E1 DA 51 01 47 7D E6 0F C2
```

That's all there is to DUMP. There is scope here for the machine code programmers amongst you to modify this so that it is a little more helpful by displaying/printing the name of the file being dumped etc.

LOAD.COM

LOAD is a program which takes a program in Hexadecimal format and converts it into a .COM file. Initially I saw no reason for it, until I discovered that the ASM.COM program assembles an .ASM file and turns it into a HEX file. Therefore to assemble a file and make it into an executable .COM file requires firstly and .ASM file to be assembled, which creates a .HEX file, which is in turn LOADed to create a .COM file. It is run by simply typing
A>LOAD INVADERS
A>LOAD B:STARTREK
The first example LOADs a program called INVADERS.HEX from drive A and creates a program called INVADERS.COM also on drive A. The second example LOADs a program called STARTREK.HEX from drive B and creates a file called STARTREK.COM on drive A. Note that the extension .HEX is the default for this command. The only programs that can be LOADed are those that execute in the TPA (Transient Program Area) ie at 0100H. Again a very simple little command, I don't know why I didn't discuss it earlier!!!

ASM.COM

This is the Assembler which comes standard with CPM. Note that it is not like EDTASM, which EDiTs and ASseMbles programs. ASM will assemble any standard INTEL 8080 format (whatever that is). When it is executed a two files are created. Firstly a HEX file which is ready to be LOADed and secondly a PRN file which contains the listing of the ASM file together with any error messages and also the Machine Code values. Confused? Well if you don't understand Machine Code then you will be, if you do understand M/C then it's my fault as I don't really know enough about M/C. All will I hope become clear so don't give up. Press on.

To demonstrate the use of ASM I will create a small file to be assembled, god knows what this file will do when executed, but that's not what I am trying to show you.

```
; USELESS.ASM
; A USELESS DEMO PROGRAM
        ORG     100H
FIRST:  EQU     0005H
```

```
BLANK:  EQU     32
        CALL    FIRST
        LXI     H,80H
        INX     H
        CPI     60H
        JMP     FIRST
        END
```

The first thing to note is that the assembler pseudo-ops, (I thought they only existed in Private Eye), are not the usual Z80 ones seen in EDTASM etc. So don't think you can transfer all your wonderful Model 3 and 4 stuff across; in fact the different codes is only the first of your problems. These codes are 8080, remember what I said about Intel etc above, however all is not despair there is a Z80 assembler called ZASM on the market and it is much the same as ASM. However, I digress, what I am going to do is talk you through assembling the file above, I suggest that you follow the steps on your own machine.

First create the file USELESS.ASM, either using a word processor or if you have followed my series carefully then try using ED, I did and it worked!!!!! Next you must call up the assembler by typing

```
A>ASM USELESS
```

The computer will reply

```
CP/M ASSEMBLER - VER 2.0
010C
000H USE FACTOR
END OF ASSEMBLY
```

(I don't know what all the values and words actually mean but still it works!

It has assembled your program and created a file called USELESS.PRN and another called USELESS.HEX. USELESS.PRN will contain the following

```
                    ; USELESS.ASM
                    ; A USELESS DEMO PROGRAM
                            ORG     100H
0100 =              FIRST:  EQU     0005H
0020 =              BLANK:  EQU     32
0100 CD0500                 CALL    FIRST
0103 215000                 LXI     H,80H
0106 23                     INX     H
0107 FE60                   CPI     60H
0109 C30500                 JMP     FIRST
010C                        END
```

This is the listing of the file plus the HEX bytes that are actually going to be loaded into memory, note that if there were any errors on assembling then they would also show up in the PRN file. The HEX file is much smaller, and simpler to understand now you have listed the PRN file. Just type

```
A>TYPE USELESS.HEX
```

The result is

(* The hex code for a colon is 3A, hence the 3A in the hex dump, LFH *)

```
:0C010000CD0500215000023FE60C3050067
:00000000
```

It is in fact the code from the PRN file packed together and ready for LOADing. A DUMP of the HEX file reveals a first line which shows

```
0000 3A 30 43 30 31 30 30 30 30 43 44 30 35 30 30 32
```

This bears no resemblance to the TYPE of the HEX file, and a little thought shows you why, the DUMP is the HEX values of 0C010000 etc from the TYPE of the HEX file. I don't know why the initial 3A is there. Lets now LOAD USELESS.HEX with the result

```
FIRST ADDRESS 0100
LAST  ADDRESS 010B
BYTES READ    000C
RECORDS WRITTEN 01
```

That should explain itself although the last line seems superfluous to me. DON'T try to run USELESS.COM that has now been created as it is useless, at this point you have to dream up your own assembly code or copy it from a book.

Well that's what ASM does, but now for some details on the options available and the errors that might occur. In the example you just typed in the name of the file to assemble, but there are some parameters that can be used, eg typing

```
A>ASM USELESS.AAA
```

will in fact tell the computer that the source and destinations for all the files (ASM,PRN,HEX) are on drive A. Yes I know it looks odd but the parameters are added instead of the file extension. This system relies on the fact that the file you are assembling has the extension .ASM. To break it down into parts, the first letter after the full stop is the letter of the drive that the source (ASM) file is on, above it is drive A. The second letter is the drive letter for the HEX file to be created on, and the last is the drive letter for the PRN file to be sent to. There are two more letters that can be used in the second and third positions and they are X and Z. A Z in the second position will prevent the HEX file being generated; a Z in the third position will prevent the PRN file being generated. Finally an X in the third position will cause the PRN file to be sent to the screen, something that I would have thought happened by default. Try these on your own machine and see if you can predict what they should and should not do

```
A>ASM USELESS.AAA
A>ASM USELESS.ABB
A>ASM USELESS.AAB
A>ASM USELESS.AZA
A>ASM USELESS.AZB
A>ASM USELESS.AZZ
A>ASM USELESS.AZX
A>ASM USELESS.ABZ
```

If you couldn't predict correctly what was going to happen then read the above section again, if still no go then either I have boobed badly or else you have a problem!!!!!

that's all there is to running ASM.COM, but now let's look at the Error Messages that you will almost definitely see, in fact you have probably already seen some of them. The first messages are those that occur when trying to run ASM.

```
NO SOURCE FILE PRESENT
NO DIRECTORY SPACE
SOURCE FILE NAME ERROR
SOURCE FILE READ ERROR
OUTPUT FILE WRITE ERROR
CANNOT CLOSE FILE
```

All of the above are fairly self explanatory, the last two normally caused by a write protected disk, note that no message to tell you that a disk is write protected exists, you have to guess that yourself!!!!

The next series of error messages are those that occur during assembly, and will be found in the PRN file.

D Data error, an element in a data statement cannot be placed in the specified data area.
E Expression error, normally incorrect syntax.
L Label error, possibly duplicated.
N Not implemented error, feature will be incorporated in future versions.
O Overflow, expression is too complicated to compute.
P Phase error, a label does not have the same value on 2 separate passes.
R Register error, often in incompatible or illegal opcode.

V Value error, probably a value outside the acceptable range.

That really concludes the bit on ASM. I have not intended to show how to write assembly language programs, but simply having written or copied one, I hope to show how to get a working .COM file from it.

DDT.COM

I have dreaded this moment because as I write this sentence I haven't even used DDT. So I will now have to take a break from typing and play with the Dynamic Debugging Tool. BAck soon.

Well that wasn't too painful and I found out an amazing fact.

I CAN TRANSFER ASCII TEXT AND BASIC FILES FROM TRSDOS TO CPM ! ! ! ! ! ! ! !

Yes who needs expensive transfer programs for only about 30 minutes hard slog and experimentation you can transfer ASCII files to CPM using DDT. Then all you need to do is make any changes to the file, such as different BASIC commands, and Abracadabra, it's done.

Anyway to get back to DDT, I will first explain how to run the program from scratch.

```
A>DDT
A>DDT WORDSTAR.COM
A>DDT DUMP.HEX
```

Usually only HEX and COM files are changed with DDT although as you will see later TXT files can be SAVEd after using DDT to operate on them. If DDT is called up with a filename then the following display will appear.

```
DDT VERSION 2.0
NEXT PC
1E00 0100
```

This tells you where the program ends, in this case it ends at 1400-1 ie 13FF, and also you are told where the Program Counter (PC) is currently pointing. The PC is of no real use to me but others will find the info useful. The NEXT value, that is to say the location where the next program in memory (if there is one) will lie, is useful when working out the length of a file for use in a SAVE command. To work out the value used in a save command the following method may be used. First if the figure under NEXT ends in 00 then subtract 100H from the value, in the above example making it 1D00H. Then ignore the 00 at the end and you have the hex value to use in the SAVE. The only thing left to do is to convert 1EH to decimal as SAVE uses decimal values. For those unsure how to do this, try looking at the table in your TRSDOS Manual, 1EH is in fact 30 decimal. If the last two figures of the NEXT value are not 00 then simply ignore them and convert the first two figures directly into decimal. So to save the program currently in DDT simply exit DDT by typing GØ, and then type

```
A>SAVE 30 FILENAME.COM
```

To recap that section, simply load a program by using the format DDT filename, then modify it and having worked out the amount of space used, save the program with the SAVE command.

Somewhere above I used a GØ command to exit DDT, well that is one of the commands found in DDT, it means Goto address 0000H and execute (not a nice word) what you find there. Other forms of the G command are

```
G       Start execution at the current PC value
Ga      Start execution at address a
Ga,b    Start execution at a and set a breakpoint at b
Ga,b,c  Start execution at a and set breakpoints at b and c
G,b     Start execution at the PC with a breakpoint at b
G,b,c   Work this out for yourself
```

Another command is to display memory.

D Display memory from the current address for 16 display lines
Da Display memory from address a for 16 display lines
Da,e Display memory from a to e

Also memory can be filled from point to point

Fa,e,b Fill memory from a to e with byte b

Blocks of memory may be moved from point to point.

Ma,e,d Move block of memory from a to e to a destination location at d

The final display series are those which disassemble memory.

L List 12 lines of disassembled memory
La It doesn't take an archbishop to work these two out!!!!
La,e

There are a number of other commands which manipulate the area of memory that you currently have in view. Note here that you are not actually changing the file on disk but modifying memory only. In fact DDT calls the program into memory at 0100H and that is where the mods are done. Remember that after leaving DDT the program MUST be SAVEd if the changes are to be made permanent. I will now list the remaining commands, some of which I still do not fully understand as all my references only say what they do and don't actually give examples of how and why they do it.

Aa Perform an in-line assembly at starting at address a ? ? ? ?
T Trace the next instructions ? ? ? ?
Tn Trace the next n instructions ? ? ? ?
U Untrace - similar to above but intermediate steps are not shown ? ? ?
If Insert a filename into a File Control Block (FCB)
R Read the filename contained in the FCB into memory starting at 0100H
Ro Read the filename in the FCB into memory offset o bytes from 0100H
Sa Set, ie examine and modify memory at address a
X Examine registers and flags
Xr Examine a specific register or flag, where r is
C Carry flag
Z Zero flag
M Minus or sign flag
I Intermediate Carry flag
A Accumulator
B BC register pair
D DE register pair
H HL register pair
S Stack pointer
P Program counter

Those of the above that I have put a series of ? after, mean little to me and I couldn't get them to work for me. The FCB lies in the first page of memory, doing a DD will show it. It enables you to read new files in from disk, but be careful with drive letters before the filename. Offsetting a file when loading it enables you to read in an ASCII file to a position in high memory, then exit DDT and CPM and run a monitor program under TRSDOS to read the file back and then save it to a TRSDOS disk. This is the reverse of loading programs into CPM from TRSDOS, and to do that you should first use a monitor or debugger in TRSDOS to read a file into memory. The file should be offset to at least 2900H, in fact to be safe I use locations around 7000H or 8000H. Then boot up CPM, it does not damage memory unless you switch off the machine. Display the area of memory and note its start and end point. Move it down to location 0100H, calculate the size of file in bytes, divide that answer by 128 to arrive at the size for SAVE. Exit DDT and SAVE the program. That's it you can now transfer ASCII text from

one to the other. Note that you have to manually work out the SAVE size for this operation.

The normal CPM CTRL characters apply within DDT eg CTRL Z to abort a command such as S.

That is all for this part of the tutorial. It is a part that I have left till late to allow me a little time to research the subject matter, however I would like to have done some more research into them, in fact a whole book exists for DDT. At this stage some of you may like me think well that's it for now I know all that I need to know. Well in the next issue I will discuss MBASIC and converting from one BASIC to another, also some words on other utilities and perhaps a little on the way that CPM actually works. Finally this time before I leave I will give you the address of the CPM User Group UK and do a small plug for them.

The CPMUGUK is in fact now called CPMSDOSUGUK! ! ! Try that after a few on a Saturday night! ! ! ! It has an excellent set of public domain (free) software on about 300 + 8" disks, some of these are from UK and Europe, most are of US origin. The only problem at this stage is that it is difficult to transfer these from 8" to 5" disks, although I believe at the last AGM someone volunteered to take on this mammoth task (come on who are you?) The programs on these disks are many and varied from business to games and numerous utilities and patches. Their newsletter is a very professional job, and sometimes contain the down to earth articles that we have come to expect from NATGUG. Membership is fairly cheap and I will certainly stay in for a couple of years. Their membership secretary is

D Fordred
72 Mill Road
Hawley
Dartford
Kent

DA2 7RZ

I hope you find them of some use to you.

Dave Holman
160 London Road West
Bath
Avon

BA1 7QU

Model 4 Notes

I wrote in the February News of problems with getting the clock to keep accurate time under TRSDOS 6.2. I have subsequently had a reply from Tandy in the form of a new system disk. After using SYSTEM (HERTZ5) the clock now keeps pretty accurate time. Unfortunately, I have no idea what patches they have applied but if anyone needs a more accurate clock Tandy now have the answer.

E.C.Kilpatrick had some harsh words in April on the inability of the Bishopsgate Tandy shop to produce a copy of **TRSDOS 6.2** which makes me wonder a little. I actually purchased my copy whilst on holiday in Cape Town in December but it apparently suffered some x-ray damage in transit and I took it back to Bishopsgate on January 2nd and they kindly made a fresh backup for me on the spot. The Bishopsgate shop is my 'local since I travel in to Liverpool Street station and I have always found them to be most helpful in the past. I hope the merger with ACT has not changed the situation.

Incidentally, whilst in Cape Town I also obtained a copy of the **1985 Tandy Computer Catalogue** which I discarded whilst packing for the return flight since I knew I would be able to pick up another copy in London ! Alas, my optimism was unwarranted and here we are six months in to the year and despite letters to Tandy UK and US I am still without this year's catalogue. What is most annoying is that Tandy have now taking the plunge and started offering non-Radio Shack software and I would dearly love to look over this list.

In the May issue Keith Taylor complained about the clock with TRSDOS 6.2 - perhaps he could contact Tandy for an update. As far as his problems with the **type-ahead** feature are concerned I suspect he may have some sort of bug since I have had no problems in this department. In fact I pride myself on being something of a typist but I have not been able to fluster the system. The problem may well be one of the application program rather than the computer. A good time to change to a real word processor and throw away the Scripsit ?

I thought I collected software but John Miller (May News) seems to have acquired everything of note plus some ! Being in the scientific field myself I wonder what data base he uses for his references ? I am unable to comment fully on the comparison between **DOSPLUS 4** and TRSDOS 6.2 since I do not have a full set of the former. However, I purchased the **"6.X Plus"** package some time back which adds most of the DOSPLUS utilities to TRSDOS and can fully recommend them to anyone wanting a bit more from his OS. Being another owner of **Allwrite** by proxy I must admit that DOSPLUS seems to have a bit of an edge over TRSDOS in speed of saving and loading but since I spend most of my time writing rather than saving or loading I would not purchase it on this point alone. It is interesting to note that the Allwrite manual states that the keyboard driver of TRSDOS 6.2 is superior to that provided by Allwrite, presumably to supplement the driver that is deficient in DOSPLUS so it seems that what one gains on the swings one loses on the roundabouts. (This is written with regard to word processing - there may well be other applications where the edge provided by DOSPLUS may be the deciding factor.)

I must concur with John in his disappointment with **"LeScript"**. I also tried it but found that although it promised so much it's scrolling was chronically slow, some of the commands were difficult to interpret and it kept falling down on simple functions and I soon managed to lose or corrupt several files - something I have never managed to do under "NewScript" or "Allwrite". There is never a perfect program to satisfy all users but the products of the Prosoft stables have a rock-solid safety factor built into every step. If there is anyone who read the review of "LeScript" in 80-Micro and thought this was the ultimate WP then I can only advise them to take that over-optimistic review with the proverbial pinch of salt. (There has not been a peep from anyone yet about the other Model 4 contender **"Lazywriter"** - does this mean that the program is utterly worthless or just that the owners are lazy writers? (Asking for forgiveness somehow seems inadequate to escape from that dreadful pun !))

I have been keeping an index of Model 4 software for nearly 2 years yet I seem to have missed the window program **"Pronto"** from Misosys. I wonder where it was advertised ? Hi-resolution graphics is one of the few things that the Model 4 really lacks (I have never been sure about the real value of colour) and I would welcome any more comments on this adaptation - particularly the ability to do screen dumps since fancy screen graphics is marvellous for the individual viewer but it is the printout that one has to tote around to impress other people.

Has anyone out there experience of **"Datagraph"** which seems to promise so much ? It now prints out from Visicalc and Multiplan files and does log scaling of graphs but apparently still lacks the ability to utilise home-grown files. It's biggest disadvantage seems to be that one has to buy a different program for every printer unlike some WP packages which include all the necessary printer drivers in the same package.

My software list presently contains 196 titles for the Model 4 and, as indicated above, it seems that some programs still escape my vigil. How can anyone say that the TRS-80 is dead?

Ken Arntsen (0279)-730354.

MICRO-80 MAGAZINE

If like me as a subscriber you have been wondering about the continued non appearance of this Tandy/Genie orientated Australian magazine, you might like to know that I have finally managed to make contact by telephone today with the English distributor, Mr Atkinson of Tunbridge Wells.

He tells me that there have been many problems in continuing production because of a fall off in sales of Tandy equipement in general and the magazine in particular. However it is intended to honour all outstanding subscriptions by the expedient of producing bumper size bimonthly issues, but then production will cease once and for all. Under the circumstances no new subscriptions are being taken.

The first of the bimonthly issues should be here in a couple of weeks for distribution in England, that would be about the end of July. The magazine is now air freighted in from Australia and is no longer printed in England. The distributor told me that he hoped that there would be an editorial statement about the non-appearance of the magazine in the next issue.

I suggested to him it would have been better for all if subscribers had been informed of the situation, and enquired why he had not at least answered my letters of enquiry to him, to which he had no answer but to apologise.

Sic transit gloria mundi!

Peter Roberts,
11 Spring Gardens,
Parkstone,
Poole,
Dorset.

WANTED - Visicalc EPROM

It's hard to resist the pressure to buy new hardware. And when anybody asks me what I use, I'm ashamed to say it's an old, high mileage, unwashed, Model I.

Then I think of its virtues: its BASIC doesn't mind whether you put spaces between words or not, its DOS will hunt through the drives without needing to be told, NEWDOS BASIC has a fast sort, MULTIDOS will read almost any disk, GEAP provides dozens of print fonts, SUPERZAP cures all ills, SCRIPSIT earns me a good living ... why should I buy an IBM PC ?

The only time I hit the Model I's limitations is with Visicalc's massive memory demands. 23K working space after the program is loaded in, just isn't enough.

There are a couple of remedies, but I'm not smart enough to do them.

Is there anybody in NATGUG who has put Visicalc into ROM? If we could switch out the redundant resident BASIC, because this isn't needed when programming in Visicalc, we

should have almost the full 48K RAM to play with. Not that this would be simple: a Visicalc EPROM would have to carry disk, printer and other routines.

Improvement Two, would be a modification so that blank cells didn't use memory. Psion did this with Abacus. If you make an entry in ZZ999, or whatever Abacus' bottom right limit is, only a miniscule amount of memory is taken up. But do that in Visicalc - even if the rest of the sheet is blank - and a huge, memory gobbling, array is set up.

So: has anybody done a patch to improve this ?

In the meantime, I'd like to hear of two other enhancements:- is there a version for the Model I that understands IF and AND? And is there a version in which you can alter a single column width without that being a global alteration?

Phone me on 0206 241725 or 0202 230337

Arnold Handley
Stanton House
West Bergholt, Colchester CO6 3JS

PENRAM for model 4

Those who have used Roxton Bakers Penram will appreciate the value of a full screen editor; not to be confused with a text editor. The majority of currently available utilities which have ram editing facilties only provide the means of editing character or line at a time. Ones that do provide page editing take up considerable space. Penram is just over 1K and allows editing, scrolling backwards and forwards in hex or ascii. I have modified this public domain program to run on the model 4. Because of the length of a source listing I have only enclosed a hex dump. For a disc and return postage I am prepare to supply the assembler listing.

Command summary:

On entry a start address is prompted for.
Arrow keys control the cursor movement.
Shift clear toggels between hex and ascii.
Shift left arrow prompts for new start address.
Shift left followed by return exits.
CTRL 'R'set ref address to current cursor address.

In ascii, any valid ascii character entered will overwrite character at cursor.
In hex, any valid hex character will overwrite the first nybble at the cursor, two nybbles must be entred before the cursor will continue. Break will restore the original first nybble, move the cursor back to in front of the byte, allowing reentry of a correct character.

```
7000    C3 AD 70 FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7010    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7020    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7030    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7040    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7050    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7060    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7070    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7080    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
7090    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF FF FF FF
70A0    FF FF FF FF    FF FF FF FF    FF FF FF FF    FF 3E 69 EF
```

```
70B0    21 18 01 CD    A5 74 21 F5    74 3E 0A EF    21 17 03 CD
70C0    A5 74 21 FE    74 3E 0A EF    CD AF 74 21    07 05 CD A5
70D0    74 21 C9 74    3E 0A EF 21    07 06 CD A5    74 3E 1E CD
70E0    9E 74 06 04    21 03 70 3E    09 EF 78 A7    C8 00 00 FE
70F0    04 20 E4 21    06 70 CD 39    73 38 DC 32    45 70 CD 39

7100    73 38 D4 32    46 70 AF 32    4D 70 32 4C    70 32 4B 70
7110    2A 45 70 22    51 70 06 02    21 0F 06 0E    3E 3E 0F EF
7120    06 02 2E 14    0E 3C 3E 0F    EF 06 02 21    19 06 0E 30
7130    3E 0F EF 2A    45 70 22 43    70 06 10 21    00 08 25 CD
7140    A5 74 D9 C5    06 04 3E 0F    EF C1 24 2E    00 CD A5 74
7150    2A 43 70 CD    0F 74 21 03    70 3E 0A EF    01 10 00 CD
7160    7A 74 D9 10    DD 3E 01 32    50 70 2A 47    70 CD A5 74
7170    3E 80 CD 9E    74 3A 4C 70    3C 47 21 03    07 11 00 01
7180    D9 2A 45 70    01 11 00 AF    ED 42 11 10    00 D9 19 D9
7190    19 D9 10 FA    3A 4B 70 3C    47 11 03 00    19 D9 23 D9
71A0    10 FA 22 47    70 0E 8A 06    02 3E 0F EF    D9 22 49 70
71B0    D9 21 10 06    CD A5 74 21    07 70 36 03    2B ED 4B 49
71C0    70 CD 52 74    3E 0A EF 21    2D 05 CD A5    74 21 06 70
71D0    ED 4B 51 70    CD 52 74 3E    0A EF 21 1E    06 CD A5 74
71E0    3E 1E CD 9E    74 2A 49 70    ED 5B 51 70    AF ED 52 38
71F0    09 11 53 70    3E 2B 12 13    18 11 ED 5B    49 70 2A 51

7200    70 AF ED 52    11 53 70 3E    2D 12 13 3E    61 EF 3E 44
7210    12 13 3E 0D    12 21 53 70    3E 0A EF 2A    47 70 CD A5
7220    74 3E 01 EF    FE 18 CA D7    70 FE 14 28    32 FE 12 20
7230    08 2A 49 70    22 51 70 18    5E 32 4E 70    FE 1F 20 0B
7240    3A 4D 70 EE    01 32 4D 70    C3 33 71 3A    4E 70 FE 0B
7250    28 02 18 05    CD 54 73 18    3E FE 0A 28    02 18 05 CD
7260    AB 73 18 33    FE 08 28 02    18 15 3A 4B    70 3D FE FF
7270    32 4B 70 20    22 3E 0F 32    4B 70 CD 54    73 18 18 FE
7280    09 28 02 18    15 3A 4B 70    3C FE 10 32    4B 70 20 07
7290    AF 32 4B 70    CD AB 73 C3    6A 71 3A 4D    70 A7 20 4B
72A0    3A 4E 70 CD    08 73 DA 6A    71 07 07 07    07 32 4F 70
72B0    2A 47 70 3E    8A CD 9E 74    3A 4E 70 CD    4C 73 CD 9E
72C0    74 3E 01 EF    FE 80 CA 33    71 CD 08 73    38 F3 4F 3A
72D0    4F 70 B1 2A    49 70 77 7E    CD 4C 73 4F    CD 61 74 EB
72E0    7C CD 9E 74    7D CD 9E 74    C3 85 72 3A    4E 70 4F CD
72F0    2B 73 30 A3    2A 49 70 77    7E 4F CD 2B    73 30 06 CD

7300    4C 73 CD 9E    74 C3 85 72    32 4E 70 4F    3E 46 B9 38
7310    0F 3E 2F B9    30 0A 3E 40    B9 38 0A 79    FE 3A 38 02
7320    37 C9 D6 30    C9 3A 4E 70    D6 37 C9 3E    7F B9 38 05
7330    3E 1F B9 79    D8 3E 5F A7    C9 7E CD 08    73 D8 5F 2B
7340    7E CD 08 73    D8 07 07 07    07 B3 2B C9    2A 47 70 23
7350    CD A5 74 C9    3A 4C 70 A7    28 05 3D 32    4C 70 C9 3A
7360    50 70 A7 28    06 01 10 01    CD 8A 74 2A    47 70 CD A5
7370    74 3E 80 CD    9E 74 2A 43    70 CD 0F 74    26 18 06 0F
7380    C5 25 25 0E    01 CD 04 74    24 0E 00 CD    04 74 C1 10
7390    EF 21 00 08    CD A5 74 21    03 70 3E 0A    EF 01 10 00
73A0    CD 8A 74 CD    94 74 AF 32    50 70 C9 3A    4C 70 FE 0F
73B0    28 05 3C 32    4C 70 C9 3A    50 70 A7 20    06 01 10 01
73C0    CD 7A 74 2A    47 70 CD A5    74 3E 80 CD    9E 74 2A 43
73D0    70 CD 0F 74    26 07 06 0F    C5 24 24 0E    01 CD 04 74
73E0    25 0E 00 CD    04 74 C1 10    EF 21 00 17    CD A5 74 21
73F0    03 70 3E 0A    EF 01 10 00    CD 7A 74 CD    82 74 3E 01

7400    32 50 70 C9    06 09 11 59    70 E5 3E 0F    EF E1 C9 E5
7410    E5 DD E1 C1    21 06 70 CD    52 74 21 07    70 36 3A 23
7420    36 20 23 06    10 36 20 23    DD 4E 00 3A    4D 70 A7 20
7430    08 CD 61 74    72 23 73 18    07 CD 2B 73    77 23 36 20
7440    DD 23 23 10    E0 36 AA 23    06 08 36 19    23 10 FB 36
7450    03 C9 CD 61    74 73 2B 72    2B 48 CD 61    74 73 2B 72
```

```
7460    C9 79 CD 70    74 5F 79 0F    0F 0F 0F CD    70 74 57 C9
7470    E6 0F F6 30    FE 3A D8 C6    07 C9 2A 43    70 09 22 43
7480    70 C9 2A 45    70 09 22 45    70 C9 2A 43    70 AF ED 42
7490    22 43 70 C9    2A 45 70 AF    ED 42 22 45    70 C9 D5 4F
74A0    3E 02 EF D1    C9 C5 F5 06    03 3E 0F EF    F1 C1 C9 11
74B0    01 70 1B 21    53 70 3E 63    EF 3E 0D 77    21 2D 03 CD
74C0    A5 74 21 53    70 3E 0A EF    C9 53 54 41    52 54 20 20
74D0    20 43 55 52    52 45 4E 54    20 20 4D 45    4D 20 20 20
74E0    4F 46 46 53    45 54 20 20    20 52 45 46    20 3D 20 20
74F0    20 20 20 48    03 2D 45 44    49 54 4F 52    2D 03 41 44

7500    44 52 45 53    53 45 53 20    20 20 20 20    20 20 4C 4F
7510    43 40 3A 20    20 20 20 20    48 03 00 00    00 00 00 00
7520    00 00
```

NOTE.
It was intended to allow alternate memory bank editing, and mem 0 is displayed on the header, it has in fact not been implemented.

John Coyne
7 Woodside Cres.
Bordon
Hants.
GU35 0EZ

OGGY OGGY OGGY

'Tis proper awful being down 'ere in the sticks, one is so far behind with the news and the gossip, and trying to keep up with changing print deadlines. First there was the Autumn NATGUG meeting, and then there was the changing story of 80-Micro; I don't really expect that anyone at WHS reads my letters (does anybody?), but in the middle of July 80-Micro magazines appeared in Smiths branches up and down the country. Take heed, Leon, it looks as though we might have a revival on our hands at last!

I believe that there has been renewed interest in the Group concerning Spreadsheets lately - anyone who has missed the last couple of issues might not know that 80-Micro now has a regular ss article intending to feature template usage. Also on ss, I hope to be doing a review soon on 'LONGVIEW', a Visicalc utility from Prosoft that allows you to print a very wide Visicalc report on an ordinary 80-column printer - you just turn a DIF file sideways and then print it out using DotWriter. Such a simple solution, why didn't I think of it first!

I had two surprises last week, one pleasant and one very nasty. The nasty one came first, with the postman knocking on the door and demanding £6.80 for customs duty on a package from the States! The pleasant surprise came when I opened the package and started to play with my copy of D.O.T.S. Read on....

D.O.T.S. is another printing utility for use with an Epson compatible dot matrix printer; it is available for the Model III, something called an IBM PC, and also in 8" format for CP/M 2.2. and its main purpose is to facilitate the use of the dot-addressability of these printers. Many people take one look, maybe two, at the printer manual and then decide to forget all about dot-addressing, and it is largely for them that this package is intended - no programming or knowledge of printer operation is needed. One merely lays a transparent grid over the

picture or design to be digitised, and tells the computer whether a dot is to be printed or not. A data file is created and the information stored until printing time. As an alternative, the package does include printed grids that one may draw a design upon, and the grid has been proportionalised so that the printed masterpiece will resemble the screen picture - no more circles turned into ellipses!

Three different grid sizes are provided, 7.5-fold, 10 fold, and 15 fold, and these will affect the size of the finished article, the largest print size being approximately 15/16" tall by 1 1/4" wide. This was my only disappointment - I had harboured hopes of a $29.95 package deputising for a $1000 digitiser! However, an enlarging photocopier or joining up multiple works will provide a larger picture if really needed.

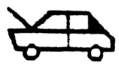
D.O.T.S. comes as a 22-page manual with 6 printed grids and three overlay sheets, and a disk containing 10 files. 4 of these are demonstration files, with 2 sample programs. Of those remaining, DOTS/CMD is the main Dot Graphic Editor, DOTSUB is a Dot Print Subroutine which can be included in your own Basic programs, and DOTEXT and DOTSHOW are Dot Printer files - the latter being used with DOTSUB in your own work. The programs are all in Basic, the Editor being compiled, and where necessary the line numbers and variables are given to help in merging routines in your own programming. When you look at the examples shown below, remember that they have been reduced to half-size by the Newsletter production.

D.O.T.S. is produced by MicroSmith Computer Technology, of PO Box 1473, Elkhart, IN46515, U.S.A. It is priced at $29.95 plus $10 p&p - and watch the customs duty!

Regular readers will know that I have no reason to give Molimerx any good publicity, but I note that their latest Special Offer includes LABELMAKER and NAMETAGR at around half-price. NAMETAGR is similar to D.O.T.S. in that you can design and print your own logo, but relies purely on cursor movements to draw the design on the screen - however it you have a symmetrical design then you need draw only one quarter of the frame and allow 'inverse' and 'reverse' commands to complete the desired pattern. D.O.T.S. allows the inclusion of normal-size text above, below, before, and after, the printed graphic whereas NAMETAGR allows the inclusion of both Jumbo and double-width text. For larger pictures than either of these programs will allow, then Powerdot II is the best that I have seen so far. To the best of my knowledge, none of this software runs in Model 4 mode - but what's wrong with dropping back to Model III now and again? At least one can appreciate a fast DOS!

Finally, on the subject of Epson-type dot graphics, I would like to hear from anyone who has used the 'Graphic Solution' software - for printer, not the hi-res screen package.

D.O.T.S. Sample prints:



David Washford, 6 Houston Way, Frome, Somerset BA11 3EU. Tel (0373) 72739

```
================================================================
REV 1.1      : UTILITY                    TITLE   : FHEX
================================================================
SUMMARY      : Convert a file to hexadecimal digits

----------------------------------------------------------------
DESCRIPTION :

Converts an input file to a hexadecimal output file.

----------------------------------------------------------------
SYNTAX    :

FHEX infile outfile

    where : infile  : file to be converted (ASCII)
            outfile : hexadecimal conversion

----------------------------------------------------------------
NOTES     : None
----------------------------------------------------------------
EXAMPLES :

(1) FHEX infile outfile
         converts input file to hexadecimal output

(2) FHEX infile *PR
         lists input file onto printer in hexadecimal

----------------------------------------------------------------
COMPONENTS : FHEX/CCC
----------------------------------------------------------------
AUTHOR       : LC User Group                  : xx xxx xx
REVISED      : Edward Rashbrooke              : 10 Apr 84
================================================================
```

```
/*-------------------------------------------------------*/
/* FHEX : list a file in hexadecimal                     */
/*                                                       */
/*                                                       */
/* Created : Edward Rashbrooke : 19 Mar 84               */
/* Revised : Edward Rashbrooke : 09 Apr 84               */
/*-------------------------------------------------------*/
#include stdio/csh
#option KBECHO OFF
FILE *fp1, *fp2;
main(argc, argv)
        int argc, *argv[];
{
        int c,d;
        if (argc != 3)
                {
                        puts("Usage: FHEX infile outfile\n");
                        exit();
                }
        fp1 = getfile(*++argv,"r");
        fp2 = getfile(*++argv,"w");
        while ((c=getc(fp1)) != eof)
                {
                        d = (fprintf(fp2,"%02.2x ",c));
                        if (d != NULL)
                                abort("Output file write ERROR !\n");
                }
        fclose(fp1);
        fclose(fp2);
}
```

```
============================================================
REV 1.1      : UTILITY                    TITLE   : FMOVE
============================================================
SUMMARY      : Move input file to output file

------------------------------------------------------------
DESCRIPTION :

Copies a file to another file. The original file is
deleted. This utility uses LDOS commands from within
the utility.
------------------------------------------------------------
SYNTAX    :

FMOVE file1 file2

    where : file1 : file being moved (ie copied/killed)
            file2 : target file being created

------------------------------------------------------------
NOTES     : None
------------------------------------------------------------
EXAMPLES :

(1) FMOVE file1 file2
        simple move of file 1 to file2


------------------------------------------------------------
COMPONENTS : FMOVE/CCC
------------------------------------------------------------
AUTHOR       : Edward Rashbrooke              : 19 Feb 84
REVISED      : Edward Rashbrooke              : 09 Apr 84
============================================================


/*-------------------------------------------------*/
/* FMOVE : file moving utility                     */
/*                                                 */
/* Created : Edward Rashbrooke : 19 Feb 84         */
/* Revised : Edward Rashbrooke : 09 Apr 84         */
/*-------------------------------------------------*/
#include stdio/csh
#option INLIB
#define CLEAR (0x01C9) ()
FILE *fp1, *fp2;
main(argc, argv)
     int argc, *argv[];
{
 int c;
 char buf[100];
 CLEAR;
 puts("FMOVE : File move utility : ER 19 Feb 84\n");
 if (argc != 3)
     {
       puts("Usage : FMOVE oldfile newfile\n");
       exit();
     }
 buf[0] = 0x00;
 fp1 = getfile(*++argv,"r");
 printf("%s will be moved to ",*argv);
 fp2 = getfile(*++argv,"w");
 printf("%s\n",*argv);
```

```
 while ((c=getc(fp1)) != eof)
  {
   if (c != putc(c,fp2))
   abort("Output file write ERROR !\n");
  }
 fclose(fp1);
 fclose(fp2);
 --argv;
 strcat(buf,"KILL ");
 strcat(buf,*argv);
 puts("\nFile moved\n");
 puts(buf\n);
 cmdi(buf);
}
```

```
================================================================
REV 1.1      : UTILITY                    TITLE   : FCOMPARE
================================================================
SUMMARY      : Compare two files for differences

----------------------------------------------------------------
DESCRIPTION :

Compares two ASCII files. All differences are listed.
This allows two slightly different versions of an
ASCII file to be compared.
----------------------------------------------------------------
SYNTAX    :

FCOMPARE file1 file2

    where : file1 : first file being compared
            file2 : second file being compared

----------------------------------------------------------------
NOTES     : None
----------------------------------------------------------------
EXAMPLES :

(1) FCOMPARE file1 file2
         compares two ASCII files

----------------------------------------------------------------
COMPONENTS : FCOMPARE/CCC
----------------------------------------------------------------
AUTHOR       : Edward Rashbrooke             : 11 Nov 83
REVISED      : Edward Rashbrooke             : 27 Feb 84
================================================================
```

```
/*-------------------------------------------------------*/
/* FCOMPARE/CCC                                          */
/*                                                       */
/* File compare utility                                  */
/*                                                       */
/* Created : 11 Nov 83 : ERCL                            */
/* Revised : 07 Apr 84 : Edward Rashbrooke               */
/*-------------------------------------------------------*/
#include stdio/csh
#define cls (0x01c9) ()
FILE *fp1, *fp2;
main(argc, argv)
    int argc,*argv;
{
    int c, d, cbytes, dbytes, nbytes, sbytes;
    cls;
```

```
puts("FCOMPARE : file compare utility : ER 11 Nov 83\n");
if (argc != 3)
       {
        puts("usage : FCOMPARE infile outfile \n");
        exit();
       }
fp1 = getfile(*++argv,"r");
fp2 = getfile(*++argv,"r");
c = d = nbytes = cbytes = dbytes = sbytes = 0;
while ((c = getc(fp1)) != eof)
      {
       ++cbytes;
       if (d == eof) ++nbytes;
       if (d == eof) continue;
       if ((d = getc(fp2)) != eof)
         {
          ++dbytes;
          if (c != d)
           {
            ++nbytes;
            printf("Inequality at byte - %d\n",cbytes);
            while ((d = getc(fp2)) != eof)
             {
              ++dbytes;
              ++sbytes;
              if (c == d) break;
              ++nbytes;
             }
            printf("Skipping bytes     - %d\n",sbytes);
            sbytes = 0;
           }
         }
      }
while (d != eof)
      {
       if ((d = getc(fp2)) != eof)
        {
         ++dbytes;
         ++nbytes;
        }
      }
printf("Total bytes file 1  = %d\n",cbytes);
printf("Total bytes file 2  = %d\n",dbytes);
printf("Total unequal bytes = %d\n",nbytes);
```

Model 1 Software, Books, etc. For Sale

All software is in good used condition on original manufacturers disks. Exact condition (eg documentation or lack of it, any faults) is as listed below. Some discs have been labelled for indexing, usually in felt tip on the manufacturers labels.

PROGRAMMING, UTILITIES, ETC

SUPER SCRIPSIT word processor; manuals, 4 disks, lessons	I/III	£20
LDOS operating system; Tandy package, 2 disks and manual	I	£15
VISICALC spreadsheet; Tandy package; disk, manual, card	I	£10
FORTRAN language; Tandy package; 2 disks, manual	I	£12
ADVANCED STATISTICAL ANALYSIS; 7 tapes, disk, manual	I	£ 3
MICROFILES data base; disk, manual	I	£ 5
TIME MANAGER business diary program; 4 disks, manual	I/III	£20
QUICK QUIZ teachers utility; 2 disks, manual	I/III	£ 6
ADVANCED GRAPHICS algebraic plotting; disk, tape, manual	I	£ 4
ALPHAKEY teaches letter recognition	I	£ 4
MMS FORTH programming language; extensive disk-based FORTH with unique DOS structure; 2nd disk won't boot but can be read and all programs used. 2 disks, documentation, book	I	£10

All the above are in ring binders; please add £1.50 per order for P&P. (eg., only pay £1.50 regardless of the number of packages ordered)

SMALL FORTH language (Molimerx); Disk, documentation	I/III	£ 3
ANIMATION package (Molimerx); Disk, documentation	I	£ 3
AJEDIT word processor (Molimerx); Disk, documentation	I	£ 5
EDIT basic screen editor (Molimerx); Disk, documentation	I	£ 5
COLPLOT CGP-115 driver (Molimerx); Disk, documentation	I	£ 5
GRBASIC adds sound, graphics. Disk, documentation	I	£ 5
ACCEL 4 compiler (Molimerx); Disk, documentation	I	£ 5
PILOT PLUS language (Tandy); Disk, card, no documentation	I	£ 2
HEXSPELL checker (Molimerx); 2 disks, documentation	I	£ 7
INFOSCAN data base (Molimerx); Disk, documentation	I	£ 5
THE DOCUMENTER flowchart program; Disk, documentation	I/III	£ 3
POWERMAIL mailing list (Molimerx); Disk, documentation	I	£ 5
11 CLOAD 80 DISKS Oct 82-Jan 83, Mar 83, Apr 83, JnJl 83 Aug 83, Sept 83, Feb 84, Mar 84 AS ONE LOT ONLY	I/III	£15
DISK INSTRUCTION COURSE (Tandy); 4 disks, leaflet	I	£ 5
TYPING COURSE (Molimerx); 2 disks, no documentation	I	£ 3

Please add 50p P&P per order from the above list

GAMES

SEA WOLF (Molimerx); Disk only	I	£ 2
ULTRA TREK (Molimerx); Disk, documentation	I	£ 3
ARMOURED PATROL (Adventure International); Disk only	I	£ 3
FURY (Computer Shack); Disk, instructions	I	£ 3
STRONGHOLD (Computer Shack); Disk, instructions	I	£ 3
DIGOUT (Computer Shack); Disk, instructions	I	£ 3
ROBOT ATTACK (Big Five); Disk, instructions	I	£ 3
ADVENTURES 1,2,3 (Adventure International); Disk only	I	£ 3
MICROSOFT ADVENTURE; the original; Disk, documentation	I	£ 3
ADVENTURES 7,8 (Adventure International); Disk only	I	£ 3
SARGON II chess (Heydon); Disk, instructions	I	£ 3

Please add 50p P&P per order for games

BOOKS

TRS-80 Assembly Language Routines	W. Barden Jnr	£ 5
TRS-80 Disk and other mysteries	H.C. Pennington	£ 5
Basic Faster & Better and other Mysteries	L. Rosenfelder	£ 5
TRS-80 Microcomputer technical handbook	Radio Shack	£ 4

Microsoft Basic Decoded and other Mysteries	J. Farvour	£ 5
TRS-80 Basic self-teaching guide (2 volumes)	Albrecht etc.	£ 4
TRS-80 Graphics for the model I and III	Kater & Thomas	£ 5
Fast Basic beyond TRS-80 basic	Gratzer & Gratzer	£ 2
TRS-80 Assembly Language Programming	W. Barden Jnr	£ 1
Programming Techniques for Level II	W. Barden Jnr	£ 1

Please add 50p P&P per book, to maximum £1.50

HARDWARE

Microdeal light pen (one tape, no documentation)	£ 2
Joystick, modified Microdeal model (repaired) no documentation	£ 3
Amplifier, modified Tandy model (socket for 9V added)	£ 2
CTR 80-A recorder (excellent condition)	£ 5

Hardware prices include P&P

To order from this list work out the total price, including postage. Send me an itemised list (if possible with alternate choices) and a cheque. DO NOT make out the cheque for a fixed amount. Leave the amount spaces blank, but write NOT TO EXCEED £xxx, with your initials, under the space for the amount in words. If I can't supply everything you want I'll make the cheque out for a smaller amount, or return it if I can't supply anything. Please don't send cash or postal orders; it's very difficult to make refunds if I don't have what you want. Make sure that your name and address are CLEARLY written or printed. If possible I'll reply by return of post. Thanks;
MARCUS L. ROWLAND, 22 WESTBOURNE PARK VILLAS, LONDON W2 5EA