

NATIONAL TRS-80 & GENIE USER GROUP

VOLUME 4 ISSUE 10

APRIL 1983

CONTENTS -----

- 2. Information on the Group
- 3. Editorial
- 3. Guidelines for contributors
- 4. Members' letters
- 7. Problem section
- 8. Machine language addition in BASIC
- 10. Zap for EDTASM/CMD to give text recovery
- 11. Boot for Newdos+
- 12. Writing a smart terminal program in C
- 13. A TRSDOS/Newdos80 problem and other things
- 17. Edit
- 18. Using C
- 20. Dotprint: a brief review
- 21. FORTRAN tutorial Part 2
- 25. Building someone else's Maplin modem
- 28. LISP/80

NATGUG
NEWSLETTER

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 at the back of the Newsletter from time to time. Sub-groups also exist for medical and educational users.

The Group runs 80-NET, a computerised bulletin board service that is available to all users of microcomputers.

The Group maintains three software libraries (Models I, II and III) which are free to members. Library lists are obtainable from the Secretary. The medical and educational sub-groups maintain their own libraries.

A membership list is obtainable on disk from the Secretary.

Back numbers of the Newsletter are available.

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,
Milton Keynes.
Tel: (0908) 564271

Librarians:-

Model I: Leighton Davies, 105 Caerau Road, Caerau,
Glamorgan. Tel: (0656) 738337.
Model II: Jim Hutton, 25 High Street, Stroud, Glos.
Tel: (045 36) 4423.
Model III: Nick Clay, 6 Rectory Lane, Byfleet,
Surrey. KT 14 7LL. Tel: (09323) 42348.

EDITORIAL

By a superhuman effort I have managed to put this issue together by the 17 April, so you should get it well before the end of the month.

I hope members will excuse my self indulgence in concentrating on the C programming language, but I do feel it is the best general purpose language available at present, although you need floating point for most applications programming. The more expensive versions have floating point, of course.

Lots of members seem to be getting into communications, now that relatively cheap modem kits are available, and new computerised bulletin boards are springing up all over the place. This is all to the good, as with more bulletin boards around the country, more users will be able to access them without having to make long-distance telephone calls. Eventually, most users should be able to get onto one with a local call, provided we have enough people willing to provide such a service.

Leon Heller

GUIDELINES FOR CONTRIBUTORS

1. If possible, send material in printed form.
2. Script or Pencil disks/cassettes are acceptable.
3. Ensure your ribbon is in reasonable condition.
4. Printer output should be on A4 paper if possible.
5. If you send in handwritten material, write legibly.
6. Do not fold the sheets when posting them. Use a large envelope and keep them flat.

Leon Heller

All the front page was by John Carne, who is the contact for North London.

WORKSHOPS

MILTON KEYNES WEEKEND

JULY 29, 30 & 31

SWINDON WEEKEND

NOVEMBER 18, 19 & 20

MEMBERS' LETTERS

I think the time has come for me to put pen to paper, I have at last received the copy of the Newsletter that kept being sent back to Brian, as a matter of interest it has no month or volume reference on it.

I keep reading letters from people who complain that the Newsletter is too erudite for them, that is only because the only people that write letters now are those who have solved all the simple problems of a few years back & newcomers think they may be thought stupid if they write & ask for answers to simple questions, the ones we needed answers to some time ago.

It's no good complaining to you, you can only publish the letters you receive, if they want answers, they must ask the questions, there is always someone willing to give a helping hand especially if they have carefully filed all their back copies of the Newsletters.

That's enough in Brians & your defence for the time being.

Here are a few notes about the OMIKRON package 'THE WORDS'. Tim Bourne made some comments in the May issue last year & these are in addition to them.

CF/M is called an Industry standard, so are MBasic-B0 & CBasic, Wordstar is supposed to be the last word in wordprocessors, well, if they are Industry Standards no wonder there is a recession.

I would have thought that by now Digital Research could have provided a better Editor considering how much money they must have raked in, no wonder programmers walked round with a dazed look on their faces, fancy trying to write a CBasic program with 'ED'. They haven't even re-written their manuals.

I also though Wordstar would solve the problem but that is so SLOW & has so many different instructions just to move the cursor let alone anything else that I have given up in disgust & gone back to my trusty 'PENCIL', (the 'Dosalum' effect has been cured with a patch from R.H.Schubert System Software in America called FENMOD).

Mbasic-B0 syntax is much too pendantic, some places you MUST put a space others you MUST NOT, not at all user friendly, CBasic gives very misleading error messages when compiling, the errors are never where the messages say they are & that's just for starters, I could go on for pages.

All in all I am sorry to say it was a waste of money, still we live & learn. I suppose I have been spoilt, writing programs using NEWDOS-B0 V2 is so much easier.

When I eventually received Wordstar there was no manual with it only a book called 'Introduction to Wordstar' which states that it is not a lot of use on it's own, the manual is essential, still it was easier to understand than Digital's ED manual & gave me enough information to try the package. Why Omikron couldn't supply a manual makes me wonder how they got hold of Wordstar in the first place although it had all the correct labels & registration documents.

John Kilpatrick.

Dear Group:

Enclosed is a check (cheque) in the amount of \$15 (ca. 7 pounds) for a subscription to the monthly newsletter.

Obviously, I have good access to Model 1 programs on this side of the Atlantic; however, I assume there must be some interesting ones that are more British in flavor--running for the House of Commons, test match cricket, and whatever, which are not available here. I should be particularly interested in anything along those lines.

Jim Glanville,
P.O. Box 4591,
Roanoke,
Va. 24015,
USA

Many thanks for the use of your Password & account No. to give my friend and I access to Your Electronic Mailbox facility. It was very enlightening and gave us a taste of things to come. I must admit, however, that our success in linking up with the host system proved to be less than straightforward, nevertheless, we did eventually SUCCEED.

Equipment Used :- Tandy Model III
Acoustic Coupler
LDUS 5.1.3.

Whilst the keys are Tapping, I would like to give my heart felt thanks (& much belated) to a Gentleman named Laurie Shields. During my "L Plate" stages he was a great help in writing a Patch to enable my Genie to drive an ex moneywell Terminus 300 Printer using RS232. Again Laurie Thanks.

On another subject, is there a circuit diagram in the house, for a Joubler suitable for my Genie 3013. I would be very grateful for any help with this matter as I would like to try my hand at building one.

D.A.MORRIS
40 Oldstead
Crownwood
Bracknell
Berks
RG123JE
20/03/83

Having been a member for a few years, I thought it time to make a contribution, so please find enclosed a couple of items which should be of some interest. I hope the ribbon strength is good enough. It is for my use but may not be good enough for copying. If not, I can supply the text in script form if you let me know.

I would like to confirm the opinion of most members that the newsletter is of great use - well done to all concerned for their effort in turning out every issue.

I can supply more articles but what articles do members want to see? My good points are pretty good at basic and medium experience at machine code. I have written M/C Programs that work which surprises me at least! although I usually modify programs from other people for my own use

My bad point is chronic slowness at actually doing something about it!

Keep up the good work.

G. Britain.

Please find enclosed a short software review that may be of interest for the newsletter.

There seems to be a rather serious slant to the articles published, mostly on utilities & reviews, but rarely anything on the fun side of microcomputing.

Prior to joining NATGUG, I was involved in writing a regular column & articles for the now defunct 80 UK newsletter. The column, called 'THIS & THAT', covered any tips or hints & short routines for use in games etc., (do you remember the INKEY\$ simulator for Level 1 at one of the early PCW shows ?).

If you think that something along these lines on a regular basis would be suitable for the newsletter, I would be pleased to oblige, perhaps bi-monthly at first. It would rely on feedback from members of course.

Pete Bamfield

Having recently pleaded for the continuance of Workshops, I was very disappointed in my fellow mankind at Swindon - a library case containing 9 DD,d/s disks of games somehow seems to have fallen out of my box which was on the floor. Since I don't have such time for games, I don't keep any backup copies of such disks, so I would be extremely grateful if the Member who has accidentally acquired this box would return it to me - I am fully prepared to refund postage and packing costs. The address is:

David Washford
1 Alexandra Road
Bournemouth
BH6 5JA

I do hope that someone out there will be kind enough to restore my faith in mankind! I don't want to bore the pants off you, but I only see my kids every three months.

Thank you

PROBLEM SECTION

Can anyone help me to get a copy of the TRS-80 Technical Manual suitable for the Model I, Level II? I thought I should get one before they went off the shelf, but seem to be too late. My local Tandy shop seem very doubtful of being able to help, and suggested the Users' Group.

Also, may I ask if it is possible to read program lines from tape as a character string, complete with line number? I had in mind "reading" a program and reporting the line numbers which include a specified character string (or even report the whole line). It could be useful, for example, if one were making changes involving a particular variable - rather like the facility offered by a "global edit".

Another sigh over the editor in Level II - can one somehow make it move on to the next line, or the next line which is not a null string?

Many thanks if you are able to help, either directly or via another member or the Newsletter.

Peter Knewstubb,
175 Huntingdon Road,
Cambridge,
CB3 0DH

Lowe Electronics kindly gave me your address and suggested that you may be able to help me with the more technical questions I have been raising with them. Firstly, I am eager to join the User Group so please send me joining instructions, literature or details of available literature, including past news letters, etc., as soon as possible.

Secondly, I would be grateful for some advice, but do not let this delay the above. I have been using a computer for three years, and now have a 16k Genie I with Lowe's high resolution graphics attachment. I am anxious to purchase a printer. Lowes tell me that the Print Screen function does not operate with the HIRES. My manual does not explain how to instruct a printer to print a normal graphics display, and certainly not how to print the HIRES display, nor whether it is even possible! Can you help? In particular, could you advise on a good printer for this purpose, that does not cost the earth? Lowes have recommended their own EG3085 (of course) which looks like a Seikosha and costs £357 + VAT. Is this the best value for money?

I have other problems, but these can wait until I am a member and have possibly made other contacts, and, who knows, might even be able to give advice! However, I really do want to get on with printing graphics.

B. Mason,
Highlands,
Brossash,
Ross-on-Wye,
Herefordshire,
HR9 7PR.
Tel: (098 981) 353

MACHINE LANGUAGE ADDITION IN BASIC

This is a machine language program to add a number of integers stored in an array. I needed such a facility to rapidly add many small numbers and this routine does just that. The program cannot simply be relocated as it requires 2 values to be entered before running, so one must be poked into memory. However I will indicate the areas you will need to recalculate. Assembler users should of course have no problem! the program should work with model 1 and model 3. This was originally designed on a model 1 with disc basic.

```
00010      OGC  0E800H      ;Anywhere suitable
00020
00030
00040  ADDINT  CALL  0A7FH      ;Get total entries to count
00050      LD  (TOTAL),HL      ;Save
00060      LD  HL,0000H      ;Zero accumulator register
00070      PUSH HL      ;Save it
00080      LD  HL,(S1)      ;Get contents of array
00090  NEXTAD  LD  C,(HL)      ;Into BC
00100      INC HL
00110      LD  B,(HL)
00120      INC HL
00130      EX  DE,HL      ;Save array position
00140      POP  HL      ;Get value so far
00150      ADD HL,BC      ;Add array value to total so far
00160      PUSH HL      ;Save it
00170      LD  HL,(TOTAL)      ;Get no of entries to do
00180      DEC HL      ;1 more done
00190      LD  A,H      ;Is it zero?
00200      OR  L
00210      JK  Z,DONADD      ;finished
00220      LD  (TOTAL),HL      ;No - so save it
00240      EX  DE,HL      ;Get next array seg
00250      JK  NEXTAD
00260
00270  DONADD  POP  HL      ;Get total back to HL
00280      JF  0A7FH      ;Return value
00290
00300  TOTAL  DEFW  0      ;Total entries to count
00310  S1     DEFW  0      ;Where to get them from
00320
00330      END  ADDINT
```

When entering from basic, the address of the array must be poked into memory which corresponds with 'S1'. The usr function is then issued with the total number of entries to count. If the array dimension is exceeded, the program will either crash or give false results.

```
100 CMD'LOAD ADDINT/ML' REM For disc users only
110 DEFINT A-Z
120 DIM A(200):DEFUSR=&HEB00
130 FOR X=1 TO 200:A(X)=X:NEXT
140 X=VARPTR(A:1):POKE &HEB29,PEEK(VARPTR(X))
150 POKE &HEB2A,PEEK(VARPTR(X)+1):X=USR(200)
160 PRINT'TOTAL ='X
```

```
200 DATA 205,127,10,34,(39),(232),33,0,0,229,42,<41>,<232>,78,35
210 DATA 70,35,235,225,9,229,42,(39),(232),43,124,181,40,6,34
220 DATA (39),(232),235,24,234,225,195,154,10,0,0,0,0
```

<<Total of 43 bytes :

Above are the decimal statements for basic users. The values with () are references to 'TOTAL' while < marks the use of 'S1'. If the program is moved, these values must be changed. 'TOTAL' resides at EB27H in the above program. To convert to decimal, treat as EB-27. EBH=232D (E*16+B where E=14,). 27H=39D (2*16+7). Your new addresses must be calculated and stored in place of the marked values. 'S1' is exactly 2 more than the address of 'TOTAL'

If you don't know how to store the data values into memory, a quick look through some back issues should help.

Line 140 is where it all goes on. Don't use any undeclared values after the 2 pokes as by doing so will move array A thereby making the just poked values incorrect. The address of the first required array value is stored in memory, and then X=USR(200) Means count 200 values. The program can count any number of consecutive values in the array. To count values 146-152, use A(146) not A(1), and use X=USR(7) to count them.

The sum of the values cannot exceed 32676 as only an integer value may be returned, but otherwise the program can count approximately 5400 integers per second. I hope that is fast enough!

If you have problems, please let me know. I'm on the phone at odd times. 021-358-2757. Leave a message if I'm not there.

Gary Brittain, 45, Waddington Ave, Great Barr, E'man, B43 5JF.

ZAP FOR EDTASM/CMD TO GIVE TEXT RECOVERY

Yes Laurie and I know that some of you NEWDOS 80 users (very wise) are using EDTASM (no comment), so here is a little something that might be of interest in case you ever "lose" or think that you have lost your text through a reboot. I can claim absolutely no credit for the clever bit that does the recovery, I think I came across it when reading back issues of the Newsletter. This is something I would strongly recommend - its amazing how much you learn in a short time, and when you come back a few months later, what was previously incomprehensible or of no use may now be just what you're looking for.

I'm afraid that I just cannot find the article in question so with apologies to the person who came up with the original idea, here goes. The actual recovery is done by making a jump to 5BDA (hex), and the original suggestion was to go into Debug and simply issue the instruction G5BDA. If you happen to use a DOS where Debug is not so easily entered, this is a bit of a problem, so my mind turned to trying to use the BASIC * idea. For those not familiar with this feature of NDBO, if you have a reboot or some other problem when in BASIC, you can often get the program back by typing BASIC * from DOS. It really does work, but do make sure there is a space before the asterisk. For those who are not interested in how these things are actually done, skip on to the Zaps at the end.

When you type any command in DOS, what you have typed is saved up ~~in~~ a buffer in memory, until you press <ENTER>, when DOS will start to examine this buffer to see what you've typed. It will look for letters and numbers making up a program name and as soon as it finds anything else (eg a space or some other symbol) it assumes that it has a complete name, which it will then look for in the disk directory(ies).

At this stage one of the Z80 registers is left pointing at the next character in the input buffer. Our little patch will look to see if the character happens to be an asterisk, and act accordingly. The code is as follows.

```
;;**** PATCH FOR EDTASM/CMD ****  
  
ORG 7660H  
  
7660 7E          LD  A,(HL)          ;EXAMINE THE CHARACTER  
7661 FE2A       CF  2AH          ;IS IT ASTERISK ?  
7663 C2006F     JP  NZ,6FO0H        ;NO - COLD START EDTASM  
  
7666 CDC901     CALL 01C9H          ;TIDY UP - CLEAR SCREEN  
7669 C3DA5E     JF  5BDAH          ;RECOVER THE TEXT
```

I have put the code at 7660H, a little bit that appears to be unused in EDTASM. All that has to be done now is to zap this along with the new execution address. Don't forget Zapper v. First Law - ALWAYS WORK ON A COPY ' I normally use SUPERZAP and the necessary instructions are as follows.

SUPERZAP <ENTER> from DOS
DFS <ENTER> This will display file sector(s)
EDTASBX/CMD <ENTER> Filename you have chosen for the copy
32 <ENTER> Sector in the file to zap
MODB1 Should display a cursor at byte B1 in
the sector display (no <ENTER> needed).
This and the next 11 bytes should be 00

Now type the following (without the spaces)
7E FE 2A C2 00 4F CD C9 01 C3 DA 58 <ENTER>

Y To confirm the modification
<ENTER> As prompted

! This will step on to sector 33
MOD53 This byte should be just after 02 02

Type the following (without the spaces again)
60 76 <ENTER>

Y To confirm
<ENTER> As prompted
EXIT No enter required - exits to DOS

If all is well that should be it. Run the new program, and put in some text. Press the reset button, and when DOS is ready, type EDTASBX \$ (or whatever name you've chosen) and the text should have returned. Type P. to discover it's even at the line number it was before the reboot !!

Don Bannister 41, Myrtlefield Park, BELFAST, BT9 6NF
Home Tel: 0232- 668879

BOOT FOR NEWDOS+

Hands up everyone using Newdos +. Type LIB. What on earth does 'DEVICE' do. Nothing, except to return to DOS. Well here is a use for this command. In Newdos 80, there is a command to reBOOT the computer, just like pressing reset. Here is a little modification to do just that for Newdos +.

The LIB commands are stored in SYS1/SYS and if one is typed in as a command, SYS1 loads SYS6/SYS and passes control to it. So to modify the system we have to alter both programs. I trust you are familiar with superzap. If not then spend some time learning to use it.

Take 1 fresh disc and make a backup of your original. Put your original disc out of reach and load the backup in a drive. Run superzap.

Modify SYS1/SYS.
=====

Select track 10, sector 1. At the lower part of the right hand display you will see LIB command words, such as FREE, DIR, DEVICE etc. First modify the word 'DEVICE' to say 'BOOT'. Type MODA4 to modify from byte A4.

Was :- 44 45 56 49 43 45
Is :- 42 4F 4F 54 20 20

The 20H are important!

Press enter and answer 'Y'. The display should now read 'BOOT' instead of 'DEVICE'. That's half the work completed.

Modify SYS6/SYS.
=====

Select track 13, sector 6. Type MDD6.

Was:- C3 2D 40 7E
Is :- C3 00 00 7E

Press enter and reply 'Y'. This change simply changes the JUMP address for the DEVICE command. Previously, typing 'DEVICE' did a JUMP to 402DH (JUMP to DOS). Now the JUMP is called by 'BOOT' which does a JUMP to 0000H (same as a reboot).

Testing - load the disc in drive 0 and press reset. This is necessary to load the new system files. Type LIB and one of them should be BOOT. If not, reload superzap and check SYS1. If all is ok, typing BOOT should reboot the system. If not, check SYS6.

When modifying SYS1/SYS, you could change the other command names by typing the hex equivalent for characters in place of those already there and filling excess characters with spaces. For example, FREE could become F and DIR could become D. It is simply a matter of choice. However, make sure that you only modify a backup of your dos. Never modify the original!!!, after all, you may not like the changes.

G. Britain

WRITING A SMART TERMINAL PROGRAM IN C

Here is the listing of a partially completed smart terminal program that I am writing in C, using the C/80 compiler. If anyone is thinking of learning a new programming language, I think that the best way of going about it is to write a few simple programs at first, just to make sure you have got the hang of the syntax, and the process of editing and compiling the code (assuming the language is compiled, of course), and then jump in at the deep end and write a fairly complex program in the new language. This technique works for me, anyway.

I don't claim that this program is particularly well written or efficient, but I have tried to use most of the features of C, and have laid it out in the approved fashion with indented lines to make the program easy to follow. I have not included any comments which is rather naughty of me. The program is very modular, and as written allows full duplex communication with the remote system, and the transfer of files to it. The modules not yet written, for opening and closing a buffer, and saving the contents to disk, are represented by dummy stubs, which just return to the main program.

A TRS80/Newdos80 Problem and other things by Roy Barber.

In the Feb issue of the newsletter Laurie Shields referred to a problem I had discussed with him. As this may be of interest to others I will give the full story. The newsletter was short on articles last month so I will also provide some additional information on my system which I hope will be of interest. At least it will help if Leon is short of material again.

The problem I had was that TRSDOS worked OK on my system but Newdos80 which I intended to use would not do a disk copy function. It appeared to perform OK in most respects but when doing a backup copy I got a verify error on track 99 each time. I later found an additional fault when saving BASIC programs. A low number line was added to the end of each program saved. When trying to delete the added line after loading the program the system would crash. I had been able to back up the master disk using the TRSDOS backup command with no problems.

My expansion interface is the LNW expansion board with 16k memory, disk controller, real time clock and RS232 sections only incorporated. The LNW has a TRS80 expansion plug on it to which I had attached an additional expansion box of my own, more on this later.

To track down the trouble I borrowed a friends TRS80 expansion box which when connected to my system instead of the LNW allowed Newdos to copy quite OK. The problem appeared to be due to the LNW interface (I thought.).

After some experimenting the solution was found. Sections of my system were disconnected one by one until the problem disappeared. In order to explain it is necessary now to describe my system in a little more detail.

I have seperated the keyboard and CPU board of my TRS80 and have the CPU board located in a box with the LNW interface. On top of these I have an additional matching box containing additional expansion boards as follows: Decoder board for EPROMS, R/W memory, Printer, Parallel I/O, Serial I/O, Battery maintained RTC and some other ports for leds etc. My eproms contain a number of utility programs such as a complex LC driver, A TC8 high speed cassette operating system, a renumber prog. and a driver for my RTC. The eproms are located at C000H upwards with about 8k currently in use. This is the reason I only have 16k of additional memory in the LNW. The eprom programs require some scratchpad memory during their execution and when I started on the system I decided I would not use the popular area at around 4050H just in case I ever wanted to use disks. (The original TC8 system used this area of memory). I configured the Eprom programs to use a 256 byte area of scratchpad memory at F000H. To provide this memory I incorporated a 256 by 8 R/W chip at F000H and redirected the eprom programs to use this new scratchpad area. As this new RAM is well seperated from the TRS80 RAM no memory protection was needed at powerup. I hate such requirements also the need to enter dates and times on powerup.

To get back to the problem with Newdos, it was found to be solely due to the extra block of R/W memory at F000H. It appears that Newdos looks at the top of each additional block of RAM at powerup to establish how much additional memory is

available and if it gets a reply from the top of the block of memory it assumes the whole block is there. As Newdos uses all available memory for a copy function it was trying to use memory that was not there. My 256 byte block at F85GH had not been completely decoded and repeated itself every 256 bytes up to FFFFH hence Newdos assumed there was memory below this from 32k up to a full 48k.

Well, thats the story. Removing the 256 byte RAM proved the point conclusively. However I have many programs I needed to keep on my TC8 high speed cassette system. I had built a remote controlled cassette unit using the Hart Electronics SF925 solenoid controlled deck which was quite a fast system and about as good as you can get using standard cassettes and for this reason I was not prepared to dispense with the scratchpad memory. Also I have a 4k eeprom emulator which I often plug in at E55GH to provide a further 4k of protected memory for system programs. I had installed a switch on the LNW disk controller to disable the disk system when required and as I did not want any more switches I decided to use the same switch to disable the additional RAM when disk operation was required. This was easy to do as spare chip enables were available on the memory involved. Obviously I can not go into much more detail here but I will happily discuss the subject on the phone or by letter.

If anyone is considering adding the disk disable switch to allow reset without a disk boot I can recommend it. The subject has already been discussed in the newsletter but details about adding the switch to the LNW appears incomplete. Also LNWs method requires removal of the FD1771 disk controller chip. This is not necessary if a small mod is carried out to the board. I can recommend the LNW expansion board as a low cost interface (if you are good at fine soldering). I did have some memory problems initially but these were overcome by using very short connections between the CPU and the LNW board. AS mine is located about an inch below the CPU board I got my connections down to about two inches.

Some of you may be wondering why I seperated the TRS80. The reason was to allow me to use an intelligent(?) keyboard when required. I can also plug in the original TRS80 Keyboard as necessary. My keyboard has its own 280, Eprom and 2k of R/W memory. It is based on a good, comprehensive commercial keyboard with 106 keys. This allows me to have many single entry keys for frequently used commands such as: gosub, return, delete etc. Also a full number pad is incorporated with all math functions such as +, -, /, =, <, > etc on single keys unaffected by shift. I have also put the Newdos 123, DFG and JKL functions on single dedicated keys. The board has 24 programable keys which can be loaded from the keyboard independantly. The big advantage with this is that I can have any command, phrase or line of text on a single key whilst using a dedicated program such as a word processor or assembler. I can put frequently used commands such as :

```
LC      A,(HL)      <enter> or:
NOP     <enter> on a key and call it 'in
one key press. When equired it will also be repeated at will if
the repeat function is in operation. Very handy if you want 20
NOPs in your text.
```

Well is that enough ? Maybe Leons got room for a little more...

No one has mentioned the Canon disk drives so far. I needed small drives so had a good look at most drives available

before settling on the Canons. The well proven TEACS and the like were simply too big for my small work area in my lounge. I already have a largish TRS80 system and with a Selectric typewriter and an Epson printer, cassette drive and other extras normal disk drives simply would not go in. As I also have a rather expanded Hammond organ in the lounge expansion of my share of the room would have been too much to expect of my wifes tolerance. I just had to get the disk drives into the space occupied by my ESF string floppy drive. I did not quite succeed but only used about a further half inch all round the ESF space. The drives are 2/3 height and look something like the Seimens units. The construction of the drives looks to be of typical Japanese quality. They are quiet and quite fast. I am using 4 millisec. track stepping time under Newdos with no problems. If anyone is considering these drives I can recommend them but get a service manual from the Importers, A.M. Electronics at Chiswick as the link selectable options inside allow almost any imaginable configuration of drive options to be set up. Also discussions with A.M. Elect. gave me the impression they were capable and willing to help users. I got my drives from Premier Publications of Croyden who were not very knowledgeable about them and only under pressure from me at time of purchase gave me a copy of a handbook on a totally different unit to the ones I was purchasing. They assured me that the information applied to my drives. It didnt !.

I went to disks rather reluctantly (They are expensive and do need a lot of memory for operation) but even using the TC8 or ESF high speed cassette systems I could not cope with large data files. In all other respects I was quite happy with my cassette systems. If you are not handling large data files I still think disks are an expensive luxury although I consider a high speed cassette system an essential for the model 1. I think that if I was stuck with only TRSDOS I would still be unhappy with disks. I must give credit to Newdos80. I consider this is a superb operating system and much prefer it to LDOS or DOSPLUS which my friends use. I must admit it will probably appeal most to those who are system meddlers rather than business users. I must admit to using SUPERZAP almost as soon as I got Newdos. I was able to Zap into my programs a number of alterations I wanted so easily. The inclusion of EDTASM has also saved me a lot of work patching my EDTASM plus for disk storage. I patched it for the TC8 system and did not want to have to do more work at once. At least I have a working assembler I can use on disk for no extra cost. Patching EDTASM+ can take a back seat while I learn more about Newdos files.

Now on another subject.

Epson MX printer users have been discussing the problem of the omitted CHR\$(5) and the problem of sending CHR\$(11) and (12) to the printer. Some suggestions for overcoming the problem have been provided but the solution I am using may be of interest. An Eprom can be programmed and fitted between the printer output data lines and the printer to convert an unused code to the (5) and any other two unused codes to (11) and (12). I put CHR\$(5) on (3) and (10) and (11) on (7) and (9). Most eproms are large enough to allow a number of 256 byte conversion tables to be stored. these can be alternative sets allowing the codes 5,11 and 12 to be put on different codes, switch selected so that if one needs the code (3) say, one can switch to a table which puts it on say (5). The only cost for

this mod. is an eprom, holder and switch. The eprom is installed in the printer on my friends system so that other computers can be connected and the printer responds in the way he is used to. Of course one of the tables can be a one for one in out copy reverting to the original codes.

In practice the 8 data lines of the printer port are taken to the lower 8 address lines of the eprom. The upper address lines go to a switch, a small dill switch in my case allowing the conversion table option required to be selected. The 8 data lines of the eprom go directly to the printer data inputs. The eprom can be permanently selected. Programming the eprom will present some with a problem, I could do a few if anyone was really desperate. The 2716 eprom I used will drive my printers through about a yard of miniature multicore cable but if longer cables are used its output may need buffering. It could be put in front of the existing interface buffer but this reduces the simplicity of the method a little. As described the eprom could be in a small box between the printer port and printer.

On one more subject...

I have just got the system program 'Trashman' from Prosoft in the States. This program reduces the 'garbage collection' hang-ups during string handling for which Microsoft BASIC is renowned. The demonstration program supplied shows really dramatic improvements in hang up times. Processing 1000 strings in the demo program takes about 180 seconds due to hang-ups whilst garbage collection occurs. With Trashman activated the time is reduced to 4 seconds. This is some improvement. The program occupies only about 570 bytes but requires a further 2 bytes for each string or string array it has to handle. The program is self locating and will load automatically below any other machine code programs already loaded. Memory protection is automatic.

One line has to be added to ones basic program to invoke Trashman as follows:

```
10 DEFUSR-578:X=USR(1000)
```

This line 10 calls the Trashman program at location -578 which is its auto-load location in a 48k system. (1000) would be the number used to allow Trashman to handle up to 500 strings variables or arrays. This program has reduced the run times of some of my BASIC programs by about 30 minutes and I have been keeping string handling to a minimum already to reduce the problem. Trashman seems to work OK with the ACCEL3 compiler but I have not yet had it long enough to know if this will always be the case. The program costs \$39.00 and appears to be very good value for money if you suffer those string clearing delays.

My apologies for not giving details at do-it-yourself level but probably not many will be affected by my particular problems or want to do the mods I have described. Anyway, I will discuss this further with anyone by phone or letter.

I hope all this has been of some interest.

PS The word processor is Word 4 word Plus.

Roy barber,
44 Dallin Road,
Bexleyheath, Kent.
DA6 8EJ. Phone:01 304 3856

EDIT

Over the past few months, I have been looking at various Basic screen editors. All have their advantages & disadvantages so I finally settled on EDIT as the best.

This utility, available on disk or tape, comes with an excellent little manual of 9 pages, neatly bound and easy to read. The manual starts with a description of the two versions, how to load & how to relocate for your particular memory size. The cassette version loaded without any problems, but it is wise to make a working copy first using COPSYS or similar SYSTEM tape copier. The program takes about 3.3k plus a variable workspace of about .5k.

Once loaded, the program can be called by typing /EDIT & then hitting ENTER. The screen is cleared & a frame appears on the two sides & bottom of the screen, together with a column of 5 digit line numbers starting at 100 with leading zeroes. A flashing cursor is on the first line. A Basic program can now be entered as normal, errors corrected by overtyping (the cursor is moved by the four arrow keys) & line numbers can be changed to suit in the same way. Care is needed when the end of a line is reached as there is a slight delay while a space is added after the line number, the display is opened up & any PRINT's that have been typed as ? are now shown in full. This means that the text will spill over to the next line & the cursor will not be at the last typed character.

There are some 30 commands available. Some of these cover deleting & inserting characters, lines or blocks of lines. Single or blocks of lines can be moved or copied either before or after an existing line. Any line numbers that are now out of sequence are shown clearly for re-numbering.

Other useful commands are Find which enables you to enter a string (up to 8 characters + 2 delimiters) along the bottom edge of the display. Pressing ENTER will now cause a search for the string from the current cursor position, the screen scroll stopping when the string is found. Find can be continuously repeated by holding down @ & F. In conjunction with Find is Global change which allows replacement of the string defined with a new version. Split is used to break up long lines into short ones while Join is useful for uniting several short lines into one. BREAK is used as a special command key, which includes return to Basic.

There is also a lower case driver routine. Basic commands can be typed in this mode, they are converted to upper case when the line is entered unless within quotation marks.

Perhaps the one thing I find EDIT lacks is the provision of macro keys for Basic commands, which save a lot of typing. However, it should not be too difficult to incorporate this without increasing the memory size too much. A minor

criticism is the bind of typing /EDIT each time the editor is needed!

On the whole, I can thoroughly recommend this utility, a little expensive at £17.50 inc., for the cassette version, but 'once used, always used'.

EDIT is available from Southern Software.

Pete Bamfield

USING C

One of the first things I did when I got the C/80 compiler mentioned in the last issue was to write a program to count the number of words in a text file. Here it is:-

```
#include "printf.h"
#define EOF -1
#define BLANK ' '
#define TAB '\t'
#define NEWLINE '\n'
#define YES 1
#define NO 0

main(){
    char c,inword;
    int nw;

    nw=0
    inword=NO;
    while((c=getchar())!=EOF){
        if(c==BLANK||c==NEWLINE||c==TAB)
            inword=NO;
        else if(inword==NO){
            inword=YES;
            ++nw;
        }
    }
    printf("no. of words = %d",nw);
}
```

Using the I/O re-direction feature of C programs I just have to type WORDCNT <FRED.DOC to count the number of words in the text file FRED.DOC.

I am currently using C to write a smart terminal program, and when that is finished I shall write a bulletin board program.

I find that C/80 works extremely well in conjunction with Macro-80, and I am gradually building up my own library of useful C functions, which do things like clear the screen, position the cursor anywhere on the screen (like PRINT# in BASIC, but using X,Y co-ordinates), make BDOS calls to CPM, and perform some rudimentary string handling. I have also written a very useful analogue of the BASIC INKEY function. This was needed to get unechoed keyboard input for my terminal program.

These functions are all contained in one relocatable object module. Using the Macro-80 Library Manager I then create a single library containing my functions, the C/80 printf function, and the C/80 run-time library. I only need to

link to the one library when I produce an absolute load module, which saves a bit of time. I actually use the CP/M SUBMIT facility to perform the tedious sequence of operations required to compile the source code, assemble the intermediate assembly language file, and link the resultant relocatable object module with the library. Using the C I/O re-direction feature the compiler error messages are placed in a file. This works because the C/80 compiler is in fact a C program.

I have done some interesting experiments with C/80, which is running on a Model II under CP/M. I have been able to compile C programs written for the Model I on the Model II, convert the object code into Intel Hex ASCII format using the /X switch when linking with L80, download the Intel Hex file to the Model I via the serial ports, convert it to binary using a Macro-80 program I have written, and load it into memory for testing and subsequent saving to Model I disk using the TRSDOS DUMP command. Of course, I had to write some C functions for the Model I I/O, as they are completely different from those for the Model II under CP/M. This process should work for other micros which use the 8080 processor, such as the Spectrum, Colour Genie, and Lynx.

Molimerx sell a C compiler (LC) that is used in conjunction with the EDAS editor/assembler on the Model I/III. This product also requires you to have LDOS. It looks rather over-priced at about £115 when compared with the C/80 compiler for £50, especially as it lacks things like multi-dimensional arrays, structures and unions. It does provide floating point, by accessing the ROM routines, however, and a much better run-time library is supplied.

I think I can recommend C to anyone who wants a change from that dreadful language BASIC. How about some other users of this interesting language writing about it. I know at least three of you are using it.

Leon Heller

FOR SALE

Green screen monitor £40; Single drive cable £5; Printer cable (Centronics 39) £7.50; Cassette software £5; Disk software £10; Joystick £5; "The Book" Vol. 1 £2.50.

Patrick Thomson
01-77 3552

DOTPRINT : A BRIEF REVIEW

Last year I purchased GEAP, a wonderful little program that allows one to compose graphics, draw, rotate, magnify, or shrink pictures to be printed out on either an Epson or an OMI. I have an MX80 to which I have recently added the Graftrax type chips.

I was very interested at the Swindon meeting to see the latest version (2.1) of GEAP which incorporated a secondary program called DOTPRINT. Dotprint basically consists of a series of fonts, or typefaces, which a Graftrax-equipped Epson can reproduce. These font programs apparently can be purchased, or alternatively, the user can create his very own letterset! I saw around thirty variations, all of which greatly impressed me, since in my other (full-time) occupation as a Sales Manager, I use sheet upon sheet of Letraset to compile printers proofs for mailing shots; Dotprint will allow me to create, and re-create, documents which can go straight to the duplicator!

Apart from an endless variation in the number of fonts available, one also has the option of either, or both, magnification and reversal - as in the heading above. If one wishes to use these options, then the Dotprint program is used, but more advantage is gained by using another program called Dotwriter, from which a 'Newsprint' format document is written which can access as many of these fonts as necessary:

The example following this one is Greek!

ABX&BBTHIKW&NDOPZTTTR&Z

**REMEMBER THAT
THIS PRINT IS SHRUNK**

When producing the Newsletter

Which is why I didn't use the Microprint Font!

There are many, many other styles available, and - as I said above - you can create your own. I don't know if the program is available in this country since Wordsworth, from whom I purchased my original GEAP, aren't advertising it. It is advertised in 80-Micro by JF Software, the publishers.

Happy Printing!
David Washford

FORTRAN tutorial PART 2

9.0 ANSWER to Part 1 Problem.

There now follows the answer to the problem set in 8.0 of part 1. I hope you got somewhere near it !!

```
C *****
C *      THIS PROGRAMME IS THE ANSWER *
C *      TO THE FORTRAN TUTORIAL PROBLEM *
C *      IN NATGUG NEWSLETTER FEB 83 *
C *****
C
C This part sets up the type of variables (see 3.0)
C
C      INTEGER HEIGHT,LOOKS,CAR,STYPE,MONEY
C      LOGICAL TALL,VGL,DL,RICH,BROKE
C      LOGICAL ANYCAR,JAG,COLL,ELEC
C
C This writes to the screen the prompt in between the ' '
C in format label 2.(see 6.1 and 6.3)
C
C      WRITE(5,2)
2      FORMAT(' TYPE IN HEIGHT,LOOKS,MONEY,CAR & TYPE ')
C
C This reads the variables HEIGHT,LOOKS,MONEY,CAR,STYPE
C from the keyboard according to the format of label 3.
C The format tells the program to read in 5 variables,
C all of which are Integer type.Further explanation is
C given in this tutorial.
C
C      READ(5,3)HEIGHT,LOOKS,MONEY,CAR,STYPE
3      FORMAT (I3,I2,I5,I1,I2)
C      TALL=HEIGHT .GE. 6*12
C
C This chunk sets up all the logical variables.
C Due to the nature of the problem these may differ
C slightly from your values.For example your definition
C of Very Good Looks could be 8 out of 10 instead of my
C 9 out of 10.(see 4.0)
C
C      VGL=LOOKS .GE. 9
C      DL=LOOKS .GE. 5
C      RICH = MONEY .GE. 30000
C      BROKE = MONEY .LE. 0
C      ANYCAR=CAR .GT. 0
C      JAG=CAR .EQ. 3
C      COLL=STYPE .GE. 2
C      ELEC=STYPE .EQ. 3
C
C The best way to decide who should date whom is by
C logical variables.This chunk does the rest.(see 5.0)
C
C      IF(TALL .AND. DL
C +      .OR. RICH .AND. JAG
C +      .OR. COLL .AND. VGL) WRITE(5,5)
5      FORMAT(' MAY DATE MARY')
20     IF (DL .AND. ANYCAR
C +      .OR. .NOT. BROKE .AND. VGL
C +      .OR. ELEC) WRITE(5,7)
7      FORMAT(' MAY DATE JANE')
C      END
```

FORTRAN tutorial PART 2

10.0 The FORMAT statement

The first part of the tutorial introduced the FORMAT statement on a very limited scale. In fact only Integers could be read in or written out. This clearly is inadequate for many situations and so there are many more Format descriptors. The most commonly used ones will now be described while the rest will be left for you to look up at some future date. The general form of a format descriptor is Descriptor w or descriptor w.d, where w stands for the width of the field and d the number of decimal places. In all cases a blank will be represented by a `_`. In the examples external is what is read in or written out while internal is what is actually in the computer.

10.1 Iw (INTEGER)

This is an easy one as we have come across it before. It means read in or write out an integer of width w. Blanks being ignored on reading and output on writing in any unused positions. For example I6

External	=>	Internal	:	Internal	=>	External
__1042		1042	:	3546		__3546
___-27		-27	:	-86		___-86
-27___		-27	:	-27000		-27000
__+__6		6	:	6		_____6
-3__4_		-34	:	-340404		*****
_____		0	:	3.14		illegal
___-3+44		illegal	:			
__3.14		illegal	:			

10.2 Fw.d (REAL)

This is the first new format descriptor and has a catch to beware of. It means that the input or output should be treated as a real number of width w and to an accuracy of d decimal places. This means that F3.1 describes a field of the form `__.` and NDT `____.`. The example is for F6.2

External	=>	Internal	:	Internal	=>	External
__3.14		3.14	:	3.14		__3.14
_-2.79		-2.79	:	-2.79		_-2.79
+_90.1		90.1	:	90.1		_90.10
0.1234		0.1234	:	49.2**3		__0.05
___372		3.72	:	49.2**6		__0.00
__372_		3.72	:	482.143		482.14
9001+1		900.1	:	55.2**6		*****
2.1E-3		0.0021	:	3927		illegal
_____5		.05	:			

FORTRAN tutorial PART 2

10.3 Lw (LOGICAL)

This is a really simple one. As its name suggests this is for Logical variables. The examples will be using L6.

External =>	Internal	:	Internal	=>	External
__T__	.TRUE.	:	.TRUE.		____T
TANDY__	.TRUE.	:	.FALSE.		____F
____F__	.FALSE.	:			
F_____	.FALSE.	:			
__TRUE__	.TRUE.	:			
__FALSE__	.FALSE.	:			
__TRUE__	.TRUE.	:			
__FALSE__	.FALSE.	:			

10.4 nX (SPACING)

This is the last descriptor I am going to deal with and it simply adds n spaces to the output or input. For example:-
FORMAT (I3,4X,I3)

Used as an output Format would add 4 spaces between the two integers output. Thus the output would be e.g 123____678. It can be used as an input Format however it is not recommended. It can be done though, e.g. If 1234567890 would result in the first I3 being 123 and the last I3 being 890.

10.5 Repeating Formats.

By now you are probably thinking what a bore to type in loads of F6.2's and I3's. You are quite correct and in this version of FORTRAN there is no way round it. However repeating Format descriptors go some way to help by allowing you to repeat the same descriptor n times. For example 2I3 is the same as I3,I3 and 2F6.2 is the same as F6.2,F6.2. That's all there is too it!!

11.0 SUBROUTINES

If you have programmed in another computer language then you probably know that a subroutine is one of the most useful parts of a computer language. Unfortunately TANDY BASIC only allows you to GOSUB to a line number, the subroutine being a part of the main programme. FORTRAN allows a more powerful version and it is this that has helped keep the language flourishing today.

A subroutine in FORTRAN is a separate programme. That is it has its own declarations, statement labels, END statement etc. They are kept entirely separate from the main programme unit. Even if your main programme and your subroutine both have label numbers of 100, they won't be confused.

Let's look at the way the main programme communicates with the subroutine. That is how the main programme CALL's the subroutine. Naturally this is via the CALL instruction. For example:-

```
CALL NEWPGE(PGENUM)
```

Where the CALL is the keyword telling the programme what to do. NEWPGE is the name of the subroutine to be called and PGENUM is the argument to be passed. This is fairly straight - forward

FORTRAN tutorial PART 2

in this example so let's look at the subroutine which is denoted by the keyword SUBROUTINE. For example:-

```
SUBROUTINE NEWPGE(PAGE)
```

Where the SUBROUTINE is the keyword telling the programme what to do. NEWPGE is the name of the subroutine, notice the correspondence with the NEWPGE in the CALL statement. Lastly PAGE; this is the formal parameter passed from the CALL. This need not be the same as the one in the call as it is only the value you are passing across. The concept I usually use is the umbilical cord between a mother and her child. The CALL being the mother, the SUBROUTINE the child with the argument at one end of the umbilical cord and the formal parameter at the other. This reinforces the view that the main programme and the subroutine are entirely separate programmes.

Thus a typical programme using subroutines will look like this:-

```
C Main programme
.....
.....
CALL ROTATE(FLAG,NUMBER)
.....
.....
END

C
C Subroutine programme
SUBROUTINE ROTATE(METHOD,NUM)
.....
.....
END
```

This means that the values held by the variables FLAG and NUMBER in the main programme unit are passed to the subroutine as the variables METHOD and NUM. It must be emphasised that the variable TYPES MUST BE THE SAME. Thus for example FLAG must be declared an integer in the main programme unit or METHOD must be declared a real in the subroutine.

Since SUBROUTINES are completely separate program units they can cause all manner of subtle errors. It would take a considerable amount of space describing the trials and tribulations possible. In fact an article as long as this one could easily be written on the pitfalls of CALLS and SUBROUTINES. It is even more frustrating that most FORTRAN compilers usually don't pick up these mistakes either. MORAL :- If you have a programme involving CALLS and SUBROUTINES and is not working look at these bits first.

Jeffrey Goldberg
458-3025

Building someone elses Maplin modem
=====

The Maplin modem was ordered by David Washford in the dim and distant past and arrived after many 'phone calls and finally a letter to Maplin. David had waited about six weeks to get his hands on the beast and called me one Sunday evening to tell me that it had arrived. Later that evening another call revealed that the kit used close tolerance resistors and such were coded in a standard but not common 5 band colour code. The kit arrived on my doorstep not an hour later with the question, Can you identify these?...About a dozen resistors left over from the bulk that were neatly labelled on a piece of paper. David left me with the kit to identify the remaining components. (A call the next day put me in the position of component identifier and kit constructor).

A first comment must be that some of the resistors did certainly look odd with their 5 band colour codes on a light blue background. Having identified the remaining resistors with the occasional help of a multimeter I was just about ready to start construction. I did find that a set of ten 10K resistors had been supplied as 6.8K but that was the only error I encountered during construction. Construction is straight forward as the two printed circuit boards supplied have screen printed legends on them making component location simple.

Tools required for construction are at small soldering iron (15-25 Watts), a pair of cutters and small pliers. Reading the article I noted that they recommend the use of sockets for the integrated circuits - these were NOT supplied and examining the I.C.s used, two are priced at £7.75 & £5.75 so use of sockets on these two I would recommend. The more expensive device is a 14412, a member of the CMOS family and as such is sensitive to static damage so caution should be exercised when handling this particular device. The remaining I.C.s should not be 'thrown' around but are a little more hardy. The use of sockets on the 8 pin devices (741s & similar) makes life easier, unless you are very good at desoldering!. The 1MHz crystal has fairly stiff leads and care should be exercised when bending the leads to fit it to the board. It is a good idea to glue it to the board, to prevent it waving in the breeze!

Once all components have been fitted to the board it is always advisable to check for solder splashes and dry joints as with any electronic construction project. Holding the board up to a strong light is a quick and easy way of detecting any solder splashes. Following the diagram supplied I wired up the connector that plugs into the modem for connection to power supply and telephone line. The kit does not include any sort of enclosure for the modem so with the LEDs wired directly to the connector, a birds nest of wires connecting the power supply, Full-Test-Half Duplex switch and Originate/Answer switch together with the line isolating transformer I applied power to the system...WITHOUT the modem board plugged into the connector. I checked the power supply voltages, +5,-12,+12 were all present and correct. Once satisfied that the wiring was O.K. I switched off and plugged the board in, without the chips plugged in. (shows you how useful sockets are!). If the pin supply voltages are ok, then switch off, wait 5 minutes, plug the chips in, and repeat the test. With all connected I switched back on...no smoke!, so with a pair of Hi-Fi 600 ohm headphones connected across the output line I listened - silence, looking back at the circuit it was immediately obvious that I wasn't going to hear anything. The GPO line has 50 Volts on it normally and this was being used to power a LED indicating that the modem was holding the line. The circuit uses a bridge of diodes to give correct polarity to the LED and with no 50V present, they were not conducting hence I heard nothing. Applying the headphones to the isolating transformer got more results, a single note was heard which altered when switching from Originate to Answer.

So with the main construction over I looked back at the text for

details on aligning the modem. Reading them through revealed nothing to tricky but the use of an Oscilloscope is needed. Do not let this put anyone off because local groups nearly always have someone who has such a device or at least can get their hands on one when the time comes. One error in the text is obvious when you sit on the emitter of TR1 expecting to see a sinewave, they mean TR2 and not TR1. Having followed the setting procedure in the text, I was almost ready to go off into the local GPO network to find someone to try this new toy on. One problem I did encounter with the system is that most communications software checks the RS232 card for silly little things like CD (carrier detect) and other signals which it is expecting to read. The connection of this modem to the computer is a three wire job (initially), Transmit data, Earth and Receive data. So to use the Tandy communication package and similar, you need to do something with those lines. A quick read of the Tandy RS232 manual revealed that by connecting CD,DSR & CTS to Vcc. The software was well and truly convinced that all was well. CD - Carrier Detect I will take from the phase locked loop lock output (i.e. the chip says - all OK to receive). Since we have a Carrier Lock LED on the modem we can use this point to tell the RS232 card when things are all locked up and ready to go (See diagram). CTS - Clear to Send is much the same, the software will wait until this line goes high before it will send anything so by pulling CTS to Vcc the RS232 will speak to the modem. DSR - Data Set Ready must also sit at Vcc to enable data to flow between the RS232 card and modem.

With all three lines tied to Vcc I tried to 'phone Distel - the Display Electronics data base available free to anyone with the right equipment. No reply...A nice girl in 'sales' told me that their computer had blown a fuse and that it may be back within a week or so'. Next I tried REWTEL, run by Ambit International again certain parts are free. The number just wrang & wrang. After getting the right number - yes it had changed I got a tone on the end of the line.... I connected and bingo, my first introduction to expensive phone bills! The introduction below was transmitted to me at a nice reading pace:

WELCOME TO EXTERNAL REWTEL.

PLEASE TYPE IN YOUR SUBSCRIPTION NUMBER

PUBLIC SESSION

TYPE HELP <RETURN> FOR HELP AND
TYPE QUIT <RETURN> WHEN YOU HAVE FINISHED.
TYPE <RETURN> AT THE END OF EACH PAGE
BEFORE ENTERING NEW KEYWORDS, OR ANYTIME TO ABORT PAGE

*** REWTEL 24hrs a day from 29th Nov ***
*** TELEPHONE NUMBER (0277) 232628 ***

WRE Spring catalogue out soon; on REWTEL soon

Subscription Services required
for REWSHOW and BINGO facilities, but
you can always leave a BULLETIN.

Public session timeout after 8 minutes

13.49 REWTEL ..

And away you go into the massive data bank available.

A few points to note when building & testing the modem are that that parts list describes one R36a as a 470 Ohm resistor - this is correct but

no R36a exists on the board, it should read just R36. If as mentioned earlier, the status lines CD, DSR & CTS are tied to Vcc the RS232C will receive and transmit to the modem even when nothing is connected to it because it has been told that a carrier is being detected (CD), it is Clear To Send (CTS) and data can be received (DSR) and the outcome of this is that occasional rubbish will be printed on the screen if you are running a terminal program. One simple solution is to use the carrier lock LED to tell the RS232C card when a carrier is present. Looking at the circuit, the output of IC8 goes low to bring the carrier lock LED on, this signal is then inverted via IC10a. A connection to IC10a output can be taken to the CD line on the RS232C card (ie pin 8 of the RS232 connector). The following table is a list of pin connections on the 25 way D-type connector as supplied with the Tandy RS232C card:

MODEM SIGNALS	25 WAY D-CONNECTOR PIN NO.
=====	=====
RS232 OUT (10)	3 - RD - Receive Data
RS232 IN (17)	2 - TD - Transmit Data
IC10a output **	8 - CD - Carrier Detect
+12V (PSU 5)	6 - DSR - Data Set Ready
+12V "	5 - CTS - Clear To Send
0V (PSU 8)	1 - PGND - Protective Ground
0V "	7 - SGND - Signal Ground

** - The pin numbers on IC10 do not correspond with the circuit diagram - The following table gives corrections to the circuit diagram:

IC10

=====

Old Pin Nos.	New Pin Nos.
1	5
2	6
5	1
6	2

Should you not have access to an oscilloscope for alignment of the modem, you can use the self test mode (middle position of S2) to adjust the device:

Connect the modem to the computer and using any terminal program (preferably with a repeat key!) adjust RV1 and RV2 so that the modem echoes back the characters typed. Start in originate mode and adjust RV1 for correct echo, then switching to answer adjust RV2 for correct echo. This will allow you to use the modem although at some later stage correct alignment with the aid of an oscilloscope is recommended. RV3 may be left at mid travel as it seems not to have any 'great' effect on performance.

I hope that these comments will be of use to anyone considering buying one of these modem kits or to the constructor who may be having difficulties.

A list of Maplin Modem owners compiled at the Swindon Workshop is given below and we hope that Bulletin board operators may allow a Maplin Modem users group corner within their systems to allow owners to exchange ideas and to promote the use of these devices between users directly and not always via a Bulletin Board system.

David Washford
1 Alexandra Rd
Bournemouth
BH6 5JA

Mac Randle
17 Langdale Close
Leamington Spa
Warwickshire CV32 7QB

Don Bannister
41 Myrtlefield Pk.
Belfast
BT9 5JA

0202 423064

Pete Smith
7 Wootton Cres.
St. Annes Pk.
Bristol 4

0926 39871

Dave Sampson
4 The Coots
Stockwood
Bristol 4

0232 668899

Nic Robinson
The Residency
Northwick Park Hospital
Harrow Middx.
01 864 5311 Ex 2588

Carl Rabe
40 Felton Road
Parkstone
Poole Dorset
0202 730617

Jim Roden (has built autoanswer unit)
36 Ribbesford Drive
Stourport
Worc's

Please excuse any errors here as most of the addrs. were very roughly scribbled onto a scrap of paper. Nic Robinson informs me that both John Newgas (01 348 6518, ringback modem) and Jim Roden have recently placed a Maplin Modem users corner on their bulletin board services so any comments, difficulties, and/or ideas about improving the modem could be left here for the benefit of all.

Carl Rabe (TOPIC)

LISP/80

I was so pleased with the C/80 compiler from The Software Toolworks that I ordered their LISP/80 interpreter. Here are some preliminary notes on it.

LISP/80 is written in C, and comes on a standard 8" IBM format SS/SD disk.

Most of the standard LISP data structures and functions are provided. 16 bit arithmetic is supported. There are the usual string functions such as PACK and UNPACK.

File I/O is available, and functions can be loaded from disk as required. The total LISP environment cannot be saved, however, so the language cannot gradually evolve to your own requirements. You have to load the new functions each time.

A simple editor is provided (written in LISP) and there are some useful debugging aids such as a trace facility, and optional break on errors. Machine-language functions can be loaded and called from LISP programs, so it would be possible to add things like floating point, with a lot of work.

A couple of example programs are provided; ANIMAL which tries to guess the animal that the user is thinking about, and a simple version of ELIZA, the simulated Rogerian psycho-therapist that Weizenbaum originally developed (and then wished that he hadn't).

LISP/80 seems very close to MACLISP and INTERLISP, and most of the programs in Winston and Horn's classic book, LISP, would probably work, with the exception of those using floating point.

It is rather slow, however, so it is not suitable for serious work. In fact, 8-bit micros are not really suitable for LISP anyway. Ideally, you should have a 32-bit machine, but a 68000 implementation wouldn't be too bad.

Anyway, LISP/80 is a good way to learn the language, and it only costs £42.

C/80 and LISP/80 can be obtained from:

System Science,
54 Enfield Cloisters,
Fanshaw Street,
London N1.
Tel: 01-739 0540

Leon Heller

Contd. from p. 12

Note the use of the switch case C construct. This is like the case statement in Pascal, but is a lot nicer. Also note the use of in-line Z80 code where I have to interact directly with the Model II hardware. Most of this code could in fact have been written in C by using assembler input() and output(c) functions.

```

#include "printf.h"
#define ACMD 0F6H
#define ASTAT 0F6H
#define ADATA 0F4H
#define EOF -1
#define CR 13
#define ESC 27

main(){
    char c;
    cls();
    while(1){
        c=kbrd();
        if (c==ESC) cmd();
        else{
            if (c!=0) serout(c);
            c=serin();
            if (c!=0) dsply(c);
        }
    }
}

```

$f \equiv \#$

```
cmd(){
    char c;
    c=getkey();
    switch (c) {
        case 'a':quit();
            break;
        case 'Q':quit();
            break;
        case 'o':open();
            break;
        case 'O':open();
            break;
        case 'c':close();
            break;
        case 'C':close();
            break;
        case 's':save();
            break;
        case 'S':save();
            break;
        case 't':xmit();
            break;
        case 'T':xmit();
            break;
        default:return;
            break;
    }
}

quit(){
    cls();
    exit();
}

open(){
}

close(){
}

save(){
}

xmit(){
    char c, fname[12];
    int chan;
    printf("\n<TRANSMIT FILE>");
    printf("\nFILENAME: ");
    gets(fname);
    chan=fopen(fname,"r");
    if (chan==0){
        printf("<OPEN ERROR>\n");
        return;
    }
    while (c!=EOF){
```

```
        c=getc(chan);
        if (c=='\n') serout (CR);
        else serout(c);
    }
    fclose(chan);
}
```

```
kbrd(){
    return (bdos(255,6));
}
```

```
setkey(){
    char c;
    do c=kbrd();
    while (c==0)
    return (c);
}
```

```
dsply(c) char c;{
    bdos(c,2);
}
```

```
bdos(DEparm,Cparm) int DEparm,Cparm;{
    fasm
```

```
        .Z80
        POP     HL
        POP     BC
        POP     DE
        PUSH    DE
        PUSH    BC
        PUSH    HL
        CALL    5
        LD      L,A
        LD      H,0
        .8080
```

```
    fendasm
```

```
    }
    serout(c) char c;{
        c;
```

```
    fasm
```

```
        .Z80
S3:    XOR     A
        OUT    (ACMD),A
        IN     A,(ASTAT)
        BIT    2,A
        JR     Z,S3
        LD     A,L
        AND    7FH
        OUT    (ADATA),A
        .8080
```

```
    fendasm
```

```
    }
    serin(){
    fasm
```

```
                Z80
S1:   XOR      A
      OUT      (ACMD),A
      IN       A,(ASTAT)
      BIT      0,A
      JR       NZ,S2
      XOR      A
      JR       S4
S2:   IN       A,(ADATA)
      AND      7FH
S4:   LD       L,A
      .8080
fendasm
}
```

Leon Heller

STOP PRESS

In last week's Computing it was reported that Onoly Enterprises Inc. are negotiating with Tandy Corp. for the purchase of 65,536 Model I 16K CPUs. They intend to connect the Model Is into a giant multi-processor system, with every processor connected to every other processor. This will give them a system about 10 orders of magnitude more powerful than the Cray I. In fact, it will use four Cray Is as front end processors. The system will be used for Artificial Intelligence research, and it is hoped that within 10 years it will have accumulated the entire corpus of knowledge of the human race. It will then be used as a super intelligent Expert System, able to answer questions on topics in every field of philosophy, science and technology, and medicine, such as the origin of the universe, the achievement of controlled nuclear fusion, a cure for the common cold, etc.

J. Torofex.

LOWER CASE FOR THE 779 PRINTER

One of the main drawbacks of the Tandy model 1 and Centronics 779 printer is its lack of lower case characters. In the past only three options have been available to overcome this problem: pay 95 dollars and send to America for a ready built unit, copy the article in 80 US and build it yourself costing about 70 pounds or sell the printer and buy a new one. None of these options appealed to me so I designed my own cheaper and more versatile unit to produce lower case.

I am willing to market this product but would like to know what form is preferable, from circuit diagrams and character generator to ready built unit and what sort of demand there would be. If sufficient demand exists I feel that a price of as low as 20 pounds could be achieved.

Please contact me with your views in the near future on the following number 06845-67494 (Malvern)

Paul Everington

CDDS AND ENDS

Milton Keynes Weekend will see Peter Tootill from Liverpool, and Alan Redhouse from York coming for the first time, Alan on LDOS and using mixed discdrives on the Model one, and Peter on DCSPLUS.

Laurie Shields and Leo Knaggs will be doing NEWDOS/80 with John Newgas on ULTRADOS. Brian Pain will do a proper session on CORFLAN, which will last around six hours, and will need 4 systems with 3/4 persons to each system.

The booking form and full programme will be sent out in the May issue. The date will be the last weekend in JULY.

During the summer holidays Brian Fain & Leon Heller will be found at MKMUG on Tuesday evenings, which is the local club in Milton Keynes. The meeting place is a factory unit laid out for micros, and members as a rule bring their systems.

John Silver on 0273-501901 has just acquired discs, and seeks help from those around Brighton area.

Geoff Henshall from Milton Keynes rang to advise "80-MICRO" is now available from major newsagents. I bought March in Euston the other day.

Richard Marks on 01-609-1340 has got an Clivetti Fraxis 30 and is delighted, and solved a tricky out of paper problem. Richard can help over the phone.

As I told you my Model III is now fitted with disk drives and upgraded to 48K. I was under the impression that the cassette version of Scripsit and Edtasm could be made to run on the disk with some help. If you know of any suggestions please let me know. Another problem is that I am unable to produce different typesetting on my Epson 82FT, the type I see in your monthly newsletter. Could you suggest anything?

Andy Cammack has moved to Plymouth, and can be reached on P 706935.

Mike Orr on 0908-612766 is interested in TRS - OSBOURNE transfers.

Luton College have their Tandy room now live, and welcome members to drop in. 0582 - 34111 Dept of Accounting AND Finance.

John Imam would like a copy of an article on TBUG for II 80-MICRO APRIL 80 01-309-1486

Coming next month a new simple timetabling program using the NDBO SORT & RUN,V facility. It works but lecturing has come before getting it ready for the newsletter. BCP.

FOR SALE

Model I keyboard, Expansion interface SK GFS printer, cassettes & manuals.

£500 ono Mike Russell 0908-604398.

DR. C. F. A. J. SALDANHA
131 Thornbridge Avenue,
great Barr,
Birmingham, B42 2AP.

