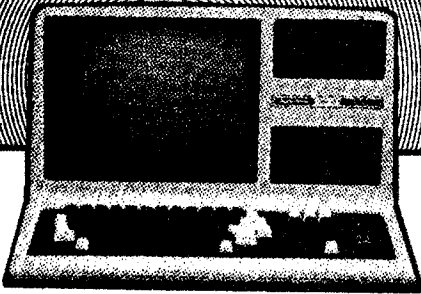


(67)

TRS-80 SYSTEM 80 COMPUTER

USERS GROUP INC.



POSTAL ADDRESS:
41 MONTCLAIR STREET,
ASPLEY, QLD. 4034.

REG. BY AUSTRALIA POST PUBLICATION # QBH 3667

CLUB NEWSLETTER
ISSUE: JULY 88

COMMITTEE 1987-8:

PRESIDENT / CHAIRMAN	: Barry ADAMSON	378 7408	CLUB BULLETIN BOARD
VICE-PRESIDENT	: Alf WEST	399 5620	
TREASURER	: John BURNS	379 9467	The OMEN
GENERAL SECRETARY	: Fred SECCULL	263 6313	
NEWSLETTER EDITOR	: Peter GOED	880 1136	Available 24 Hours a day
BOOK & MAG. LIBRARY	: Colin WOLF	374 1357	
PROGRAMS CO-ORDINATOR	: Noel HODGE	287 5020	PHONE 203 4882
DOCUMENT CONTROLLER	: Jim MELKSHAM	814 4333	
MEMBER PROBS. & TOPICS	: Bill STOMFAY	857 6353	All speeds to 2400 Baud
COMMITTEEMEN	: Bert ARTHUR (075)	48 2839	
	: Bill ALLEN	343 5771	Are YOU making use of
	: Alex BUIDEN	372 2934	This VALUABLE facility
	: Jimm WRAGG	273 5543	If NOT then WHY NOT??

PRESIDENTS REPORT
by Barry Adamson

Due to hardware problems with my 40 track disk drive, and apparently also my modem, the article introducing the Q & A section of the newsletter did not appear in last month's issue but appears with this one. I invite you to let the Editor have any questions you might wish answered, and to toss in any answers you feel have been left unsaid to previously answered questions as time goes by.

I, personally, would like to see more "chips and wires" articles in the newsletter, because that happens to be my particular interest. I hope to be able to contribute an article in the near future outlining the saga of the isolation of the reason I can't write to my 40 track drive. So far I've had time to prove the write protect circuit is functioning correctly and a suspect decoder chip was changed, all to no avail. In the process I'm being forced to learn how the circuits go together so it's a no lose situation if I don't hold my breath awaiting the final fix.

Am I the only one doing things like this without previous experience? If not, tell us about your experience in tracing faults and what you learned from it.

I was sorry to miss the last meeting, particularly as I've heard nothing but rave reports about the talk by Colin Wolf on his use of the TRS-80 in his business. Well done, Colin. Who's next?

What about you country members. Tell us a bit about yourself and the work you are doing with your computer and any hassles that go with being away from the Big Smoke. Any power supply problems?

I'll leave you with yet another reminder that we are a mutual help club and you are urged to get in to the thick of things and get your money's worth and more out of being a member. If you've got something to say by way of a talk and it's not in your nature to stand in front of a crowd, have a chat on the quiet with me or Alf or Lance and you may even find we are the perfect informal crowd for you to "break the ice" and commence a career of public speaking. Everyone had to make a start (take it from your editor that this is true, as I was shy till I joined the group, now I have become a BLOODY! extrovert). There's more things than Computing we can help you with, in all probability.

Until next month, happy computing and may the smoke never escape for you.

-----<TRS80SYS80CG>-----

MINUTES OF MONTHLY MEETING JULY 1988

Meeting held at St.Cintra's Hall,Bowen Hills on Sunday,July 3rd 1988 at 2 pm

Apologies received:- Barry Adamson.

The Vice-President, Alf West, opened the meeting, gave Barry's apologies and greeted members, new members and visitors. He called for the General Secretary's Report.

GENERAL SECRETARY'S NEWS:-

The Secretary reported on the month's correspondence, which had been very little. Apart from membership enquiries, the main incoming mail had been the Incorporation Certificate from the Justice Dept. However, it would have to be returned for correction as the name change was not satisfactory.

The outgoing mail had been a letter to the Dutch users group, of which we had just received the address, asking if they would like to exchange Newsletters. There had also been one or two exchange newsletters received.

EDITOR'S REPORT:-

Peter reported that he was short of articles, he has only three pages at this stage. He said "If you want a magazine, send in the articles. What you send is what you get." Bill Allen said that everyone is doing something with their computer, "when you find something new, a difficulty or a solution to a problem, jot it down and write an article for the magazine". That is how he had been operating. We are, after all, a self-help group.

DOCUMENTS CO-ORDINATOR'S REPORT:-

Jim told members that the photocopier would be disappearing in a couple of weeks. He explained the new system.

Members will be charged a deposit, which will be sufficient to cover the cost of replacement of the master, in case of damage or non-return, plus a rental fee of 2c per sheet. The rental fee is to cover replacement, due to wear and tear. The deposit will be payable in advance at the time of ordering, and will be returned, minus the rental charge, provided the masters are returned within the month, in good condition. The rental fee will be increased to 4c if the masters are kept more than one month, unless an arrangement for an extension has been arranged with the Documents Co-ordinator.

Country members will be looked after by the Committee.

PROGRAMS CO-ORDINATOR:-

Noel reported that he had an advertisement from Microsys in the U.S. for a high-res graphics board and programs for the model III and 4/4p. He asked for expressions of interest, with a view to a bulk purchase.

LIBRARY:-

Colin read out a list of members who had books outstanding for extensive periods. If anybody has a book which they have had on loan for more than one month, please contact Colin and arrange to return it, or make special arrangements for it's retention.

-----oOo-----

GENERAL BUSINESS.

Bill Allen reminded members of the availability of cleaning fluid for their disk drive heads. Just bring a 100ml bottle with a well fitting cap, he will return it at the next meeting.

Alf West reminded members that we have labels for sale. These are the same size as those on the envelopes of the Newsletter. They come 12 to a sheet, 8 sheets for \$1.00. Alex Buidan is looking after them. See him at the meeting.

MEMBERS PROBLEMS

The monthly discussion of members problems then took place, under the direction of Bill Stomfay.

This was followed by a discussion of B&B No.66, led by Jim Wragg.

The meeting closed at 3.50pm

-----<TRS80SYS80CG>-----

NEXT MEETING

The next meeting will be held on Sunday August 7th, at St Cintra's Hall, Bowen Hills, at 2 pm.

The Topic is: Printers by John Dow from Epson and Darryn Cook from Edcom
Covering available printers from Epson and their care and maintenance, etc.

-----<TRS80SYS80CG>-----

FOR INFORMATION:- If a member needs to be contacted at the hall, in an
EMERGENCY, the telephone no. is 854-1917

-----<TRS80SYS80CG>-----

MEMBERS PROBLEMS

by Bill Stomfay

PRIOR FOLLOW UP

John Aarsse - hardware problem with Model I expansion interface; circuit diagram obtained from Melbourne and problem now identified; waiting on part.

Ken Carkeet - problems under TRSDOS 6.1 with date stamping after 12/31/87. Formatting to 41 tracks resulted in errors when attempting to copy files. Still awaiting Mt. Isa contact.

Colin Wolf - Lazywriter on Model 4P and latest version Superscripts; has not yet tried new upgraded disk.

Dave Clarke problems transferring BASIC programs and word processing files using MODEM 82; see article.

JULY PROBLEMS

Jim Melksham is trying to locate a copy of SUPERMOD IV (a DOS enhancement that provides some of the features of NEWDOS86) for a country member - despite that members phone manners. John Hildyard has a copy of the program but no documentation.

Jim McPherson (a new member) is encountering problems with a PERTEC drive under OS80 - disk sectors keep going bad after formatting OK. Peter Goed advises that the Pertec drives have a critical part made of plastic that is subject to wear - once it goes the drive is useless. Peter also advised that he now has a spare computer and is prepared to fix disk drives as required for \$10 per drive (half of which goes to the Club, the remainder to Peter for replacement of equipment). Alf West thanked Peter for his generous offer and I'm sure that many Club members will benefit from this service.

Alex Buidan has problems with a BASIC spelling program on his Model III - it keeps bombing out on the same line and the listing only shows rubbish. Two possibilities were suggested. Firstly, it could be that the program was written with a machine language routine for non-disk BASIC, and this is

overwriting the memory required by DOS. In this case, there are recent articles in Bits and Bytes on conversion. The second possibility is that there are embedded control codes in the program to discourage listing and/or modification. This can be gotten around by saving the program in ASCII format (rather than tokenised format, which is the default) and using a word processor to examine the resultant ASCII file. To be followed up.

Jim Wragg is after a copy of TRSDOS 2.3 DECODED.

Bert Arthur enquired about the use of printer control codes in Lazywriter. As with Lazywriter and Allwrite, most decent word processing programs allow the selection of a printer driver (from a range of printer drivers supplied with the program) that translates the word processing codes for bolding, underlining etc. into the appropriate printer control codes (which are normally the escape character CHR\$(27) followed by one or more other characters). However, if the printer driver is not exactly the correct one for your printer, or there is some special feature not supported by the standard printer driver, you may need to send control codes directly.

Frequently used codes should be set up as soft keys (e.g. the combination of CLEAR and I could be defined as a series of keystrokes to send the control codes CHR\$(27), CHR\$(45), and CHR\$(1) to turn on underlining). The soft keys, once defined, will save a lot of time later and are very versatile.

Greg Elkenhans is having problems accessing bulletin boards that use ANSI screen codes to produce IBM graphics - his Model 4P doesn't like them. He is in the process of trying a number of PD modem programs, but the consensus of the meeting is that the only way around the problem is to modify the filter or translation tables within the modem program to either translate or ignore the ANSI codes. Since many of the ANSI screen codes (e.g. those that modify screen colour) have no equivalent on the TRS80 machines, the easiest option is to convert all incoming ANSI sequences to nulls.

-----<TRS80SYS80CG>-----

PRESIDENT'S NOTES

by Barry Adamson

The Club newsletters contain a wealth of information on just about every aspect of Computing imaginable. In addition, we have a library of newsletters from our club exchange system to add to that fund of knowledge. The articles contained therein have been contributed by various members in the hope (certainly justified in my humble opinion) that they will be of assistance to other members, either now or in the future.

As a further service to Club members, we have a regular Member's Problems segment at our monthly meetings which is always put to good use by those Members present. I have been wondering lately if, perhaps, we have been placing too much emphasis on the "problems" aspect of this service, and possibly as a result, tending to stifle genuine enquiry and thus neglect an important area of service to members.

In a Club such as ours, it is possible for some Members to be completely overawed by the breadth of knowledge of others and, as a result, feel that they have nothing of value to contribute to the newsletter. Human nature has led me, over the years, to know otherwise. How many times have you heard someone apologise for their lack of knowledge and then proceed to ask a question only to find many ears straining for the answer.

Often, when approaching a problem of lack of knowledge in a specific area, we tend to go off in a somewhat blind search of our reference material, frequently being diverted by other interesting, but probably unrelated, facts. That happens to me quite regularly. It would be very handy at such times to be able to ring some very knowledgeable person and have the information sought and the references to it crystallised in one concise bundle. Since, as I said before, there would undoubtedly be more than one member interested, the logical forum for such queries is in the Newsletter.

I propose, in order to gauge the response to such a service, to institute a section of the Newsletter to be called Q and A or I/O or some such, to cater for general queries. It is not a new idea, just a redirection of emphasis. The answers will be supplied by a member or members with expertise in the particular area - I'm sure the Editor will never be heard to complain that he got several answers and will be quite happy to compile the various points made. This should serve two purposes: firstly, it will be of direct benefit to the Members concerned and secondly, it will provide a means for more Members to have an input into the newsletters. It will make life a little easier for the Editor as well. Naturally, it will be necessary to ensure that full information is given with queries e.g. versions of software, models of hardware etc.

To start the ball rolling, I'll throw in a few to get things going:

1. If I type in a Basic program from the keyboard, where in RAM is its start address and is it always kept together in contiguous memory - apart from variables and strings.
2. If I then save this Basic program to disk and reload it into RAM, will it return to the same locations?
3. If I type in a text file using Lazywriter and then examine memory, where do I look for Lazywriter and where will I find my text. Does LW always place the text at the same starting address?
4. If I save my LW text file to disk and reload it, it doesn't go back to the same location in memory and I have the text twice in memory - with apologies to the Prof, " why is it so?"
5. LMOFFSET will give the location in memory of a CMD file - what about Overlays etc?

-----<TRS80SYS80CG>-----

THE MARKET PLACE

HARDWARE ETC FOR SALE:

John Bird has the following items for sale.
Contact John on 2611952 at home (or 2251509 at work).

Model I 16K with lower case fitted. \$50

System 80 computer system. This is THE computer that Jakub built. Keyboard has 64K memory with triple speed and customized ROM monitor on boot up. Outboard unit has 256K plus real time clock. So that you can access disk drives. I'll include an expansion interface with RS232 (and another 32K of memory) for a package price of \$250. (Will separate).

HI RES unit for System 80. 384 x 192 pixels, if it works. Complete with manual and tech data, sockets alone worth \$15.

RS232 kit can be assembled to suit System 80, Mod 1 or 3. \$90.

Hard disk adapter kit for Model I. \$60.

Analogue input card to suit Mod 1. 8 analogue channels, 16 bi-directional digital ports. \$90.

Interface kit. To be assembled to mount 10 EPROM's. \$50.

Model 1 printer interface (use printer without expansion interface). \$25.

System 80 joystick interface. Sockets alone worth \$15.

40 way triple adapter. Short cable with one card edge socket and three card edge plugs all gold plated. Will separate if several people want gold edge connectors to suit Mod1. \$60.

Printer buffer with 256K buffer and switched output for connecting two printers to one computer. \$250.

Relay interface card. 4 relays mounted on PCB with space for 4 more. \$40.

-----oOo-----

MODEMS now surplus to requirements: 1 300 baud Dataphone II, \$100. 1 Sendata 1000 (300 & 1200/75 baud), \$190. Both complete with phone handset & all cables ready to plug straight in and use. 1 System 80 Mark II 16K kbd unit (numeric key pad) in pristine unmarked condition w/ extra screen memory, \$100 ono. Bill Allen, 343 5771.

AEM 4600 MODEM, 300, 75/1200, 1200/75 baud rates, New. \$250 Contact Noel Hodge Tel (07) 287 5020

TELEPHONE SAFETY DEVICE (Patented). This inexpensive little gadget could save you an eardrum or even your life, as well as your modem. Consists of a double plug safety device (inside) between telephone line and modem. Simple plug in installation plus connection to grounded water pipe. Protects against any voltage surges on the line caused by lightning strikes etc. Lifetime guarantee, virtually indestructible. Few available next meeting @ \$30 (normally \$39-95) each or order by mail (add \$1 for postage). Peter Goed, (07) 880 1136 a.h., 252 5988 b.h. or write: 22 Shields St., Redcliffe 4020 (SOLE Qld agent).

SOFTWARE FOR SALE:

#NEWDOS86 -- Warwick Sands' update package for NEWDOS80 v2 (Model I, System 80, Model III, Model 4/4P in Model III mode), has just about all the extra functions you've always wanted in an operating system, plus very powerful CUSTOM BASIC FOR '80 USERS programming and runtime enhancements to Disk Basic and screen oriented Enhanced Line Editor. Ongoing support always available, being followed up by further enhancements. Preliminary Hard Disk Version (Model 3/4) now also available as an extra. \$75 for either Model floppy version, \$5 extra for both versions (Installation disk(s) and 150+ page manual). Add \$5 for P & P if by mail in Australia. Obtain from Bill Allen at meetings, or 16 Laver St., Macgregor, Qld. 4109, (07) 343 5771 (make cheques payable to W.J. Allen, who is acting as Warwick's agent).

#HELPPDISK -- a facility to interrupt most programs under Newdos80 (or Newdos86) or from Basic or Dos Ready (both Model 1 and Model 3/4 versions available) to comprehensively, rapidly and flexibly view information on any of the specially formatted and indexed /HLP files in the system and return with the interrupted status completely restored. Proceeds to CLUB FUNDS. Once only cost: \$10 (contact Bill Allen as above).

#MACASMON -- (By Glen McDiarmid) a very powerful MONITOR program for the Models 1, 3, and 4/4P (in MODEL 3 MODE ONLY). (See review in Bits and Bytes No.47 P.8) Full documentation files on issued disk. Features 6 separate screen displays, excellent disassembler (forwards AND backwards), screen snapshot capability, buffers last 7 executed instructions, calculates and keeps running total of T-states, etc etc etc. (Sold on shareware basis) Bring \$5.00 and pick it up at a meeting (see Bill Allen), or send \$10.00 P&P

#TIME MACHINE FOR THE MODEL 4 or 4P (Utility). BARGAIN !!! NOW ONLY \$20 + \$5 p & p anywhere in Australia. Includes Disk and manual. (See review on Page 10, Bits and Bytes No. 45). Add concurrency to Model 3 dos's on the 4/4P, run 2 programs simultaneously in the machine, cheat like crazy at games (learn to master even the most difficult key control -- e.g. would you believe over 200,000 scored from Volcano Hunter?), plus numerous other VERY USEFUL functions using banked memory and sophisticated key control of the machine available at all times. A MODEL 4 VERSION IS NOW AVAILABLE. Contact Peter Goed (07) 880 1136 or Bill Allen.

BOOKS:

#NEWDOS/80 HACKERS HANDBOOK by Kevin O'Hare. Packed with useful info for programming under this DOS and memory usage in our machines, VERY USEFUL FOR OTHER DOS USERS ALSO. \$25 (add \$4.00 if ordered separately by post). Contact Peter Goed at meetings or 203 4882 or Bill Allen.

-----<TRS80SYS80CG>-----

PUBLIC DOMAIN PROGRAM INFORMATION

by Noel Hodge

In the April 88 issue of BNB it was stated that I had available 4 copies of the Model 4 Public Domain Software Catalogue for perusal at our monthly meetings.

Now that sure enough is alright for members that turn up to the meetings, but "what about those members that can't attend" are they left in the dark as to what programs are available. No way, I intend to let all of our members know just what is available.

This article is directed mainly at our country members "after all there are as many members out there" (if not more) that should be kept informed and up to date with our library. Now I have thought a lot about how do we fix this so that you, may know what is programs are around. We could send you a catalogue, but that does cause a problem.? Catalogue for what Model 1,2,3, or 4 or some may have two computers, "I've got 3" so what do we do.

The Solution is that we will send a disk containing our P.D programs covering all Models to any member that asks, for a small fee of \$5.00 including postage. Contact me and I will arrange the rest for you.

I would also like to point out that I have recently been shown an ad from overseas for High Res boards for the Model 4,4p going at about \$250 landed here. A bulk order is in progress, as we have a few interested and the order will not be sent until after the August meeting thus allowing for any country members to get in on this offer as well. I would like to point out that anyone interested will have to pay when placing the order.

-----<TRS80SYS80CG>-----

IF YOU NEED HELP ON ANY OF THE FOLLOWING, CONTACT THE MEMBER NOMINATED.

Level II Basic	Lance Lawes
Disk Basic	Alf West, Bill Allen
NewDos86	Warwick Sands, Bill Allen
NewDos80	Alf West, Peter Goed, Bill Allen, Warwick Sands
TrsDos 1.3	Bruce Bussenschutt
MultiDos	Peter Goed
OsDos (Dick Smith)	Peter Goed
Assembly Language	Alf West, Bill Allen, Warwick Sands
Cobol	Noel Hodge
Pascal	Graham Leese
Fortran	Bill Stomfay
Visicalc	Bob Westerman, Lance Lawes
Superscript	Jack Bognuda, Laurie Lingard
Lazywriter	Alf West, Bill Allen
All Write	Bill Stomfay
Lscript	Bruce Bussenschutt
Time Manager	Bert Arthur
Data Bases	Alf West, Bert Arthur
Bulletin Boards	Peter Goed
Hardware Problems:	
Model 1 and System 80	John Bird
Model 3	John Bird
Model 4/4p	Peter Goed

Disk Drives	Peter Goed
Printers	Dave Clarke
Eproms & their Programming	Dave Clarke
Modems	Noel Hodge

This list is issued as a guide for those people to contact if you have a problem, it is by no means complete and the people referred to may be able to do no more than offer advice as to who/how your problem may be overcome.

For phone numbers for these people please consult your LATEST members list.

-----<TRS80SYS80CG>-----

From the EDITORS Keyboard
by Peter Goed

Let me thank you for your response to my request for articles, you have done me proud. That does not mean to say that you can all go back to your previous ways and not submit any articles. The 4% of you who submitted 140% of my accumulated articles are excused, well for the time being, the other 96% of you should now put pen to paper (well keyboard I hope, as I am not much of a typist) and give me more articles for future newsletters.

With your help, we will turn this magazine into a worthwhile bible for future generations of owners of our old and pre-loved Z80 machines. That is not to say it was not already doing this, but there is always room for improvement, isn't there? So let me know what you are doing, or for that matter what you are not doing, because you don't know how to do it. Your little GEM of a FIND might be just what countless readers WORLDWIDE have been looking for, remember that we are now one of, if not the only group, worldwide that supports the TRS80 Z80 based machines ONLY. I feel that a great responsibility to users on a worldwide basis is thrust upon us, don't let this daunt you as in the past such GEMS as NewDos86, MacasMon and HelpDisk have all been generated from within our group, we still have several people with a GIANT amount of knowledge within our ranks and these ranks are continuously growing. With a membership in the vicinity of 180 persons and growing, we can really do as the ads say "You can do it BRISBANE", sorry country members, but if you can help, I will also fit you in as an extended Brisban'ite.

The DEADLINE for the next magazine is WEDNESDAY August 17th, at 8pm.

-----<TRS80SYS80CG>-----

Just a reminder about the SIG held at Lance Lawes' place on the last Sunday of the month. Due to the magazine coming out after this day, you will have to remember the day for the next month. As accommodation is limited, give Lance a ring to confirm there will be room for you and to find out the times.

-----<TRS80SYS80CG>-----

BITS & BYTES

ISSUE No. 67

A QUICK REFERENCE CARD FOR NEWDOS 86 EXTENDED LINE EDITOR COMMANDS

By F. Seccull

I find that manuals seldom give me the information I require in the format that is most useful to me. The NWD 86 manual is very good, but I need to have the E.L.E. commands in the reverse order to that in the book (i.e. Action followed by Command rather than Command followed by Action), as I can't remember all the various keystroke combinations to get the best results. I decided that I required a ready reference card and thought that others might also find such a sheet useful. So out with Lazywriter and the NWD 86 manual and here is the result.

NWD86 E.L.E. COMMANDS

ACTION REQUIRED :	COMMAND :
1. <u>CURSOR CONTROL</u>	Use arrow keys
2. <u>PROGRAM LISTING</u>	
Print 1st page	CTRL SHIFT UP ARROW
Print last page	CTRL SHIFT DOWN ARROW
Print previous page	SHIFT UP ARROW
Print next page	SHIFT DOWN ARROW
Print current page from memory	CTRL@ or CTRLa
Move up 1 line (if at top of screen display previous line)	CTRL UP ARROW
Move down 1 line (if at bottom of screen, display next line)	CTRL DOWN ARROW
3. <u>INSERT MODE</u>	
Enter Insert mode	CTRL I
Insert a blank line	CTRL DOWN ARROW
End Insert mode	BREAK
4. <u>DELETE MODE</u>	
Enter Delete mode	CTRL D
Delete 1 character	CTRL D
Delete 1 char. & all blanks to next non-blank character	CTRL RIGHT ARROW (TAB)
End Delete mode	BREAK
5. <u>EDITING</u>	
Edit current line	CTRL E
Edit last entered command	CTRL :
Store current line with all spaces inside quotes converted to CHR\$(10)s or space compression codes	CTRL G

5. EDITING continued

HACK	CTRL H
KILL	CTRL K
nKc	CTRL SPACE n CTRL K char.
SEARCH	CTRL S
nSc	CTRL SPACE BAR n CTRL S char.
Enter ASCII character	CTRL C
Store current line as displayed	ENTER
Erase to end of screen	SHIFT CLEAR
Exit Basic Auto mode without entering line	SHIFT BREAK
Insert E.O.L.marker	CTRL SHIFT ENTER
Print [(open square bracket)	CTRL ,
Print] (close square bracket)	CTRL .

[Ed] _____ I will have some sheets with this Quick Reference Card printed on Indestructible paper using a Laser Printer available at the next meeting for \$1 each.
Country members may order these direct from me at \$1-60 posted.

-----<BNB>-----

TRANSFERRING WORD PROCESSING TEXT FILES FROM TRS80 TO IBM

by Bill Stomfay

The subject of transferring files from our TRS80 machines to the new 'Big Blue' IBMs has been raised a number of times at recent meetings. I've been through this myself, so, thought I'd relate my experience in this area.

There are three basic steps involved - 1. preparing the files for transfer, 2. the actual transfer process, and 3. reading the transferred files into an IBM word processor. One of these three steps may be redundant, depending on what you want to do with the transferred text.

Firstly, preparation for transfer. There are two approaches; a transfer of a raw word processing file, or transfer of a formatted print file. The one you use depends on what you want to do with the file once you get it over to an IBM. If you simply want to print out the document on a higher quality printer, then you should transfer a formatted print file. A formatted print file is created by directing the word processor output to a disk file rather than a printer, so that the file is actually an exact replication of what would be sent to your printer.

If you're using Lazywriter, this is a standard print option. If your word processor doesn't have this option, you can achieve the same result via DOS. NEWDOS86 has the facility to create a phantom device (via the DEVICE utility). This may be a disk file to which printer output can be directed using the ROUTE command. If you're still using NEWDOS80, there is a utility called STD/CMD (written by Warwick Sands) which will achieve the same result. Both of these techniques are documented in past issues of Bits & Bytes (Issues 60 & 31). The advantage of this approach is that you don't need to worry about getting your text file into an IBM word processing program - you just PRINT from DOS. However, there are a couple of disadvantages, which I'll discuss later.

If, however, you want to do a significant amount of editing to the text file, you should transfer the raw word processing file. This is not a problem if you're using a word processor such as Lazywriter or Allwrite that saves their standard files as ASCII (although you should check the upper/lower case default with Lazywriter) and that don't have any special headers on their files. Superscriptsit, on the other hand, is a real pain because it scatters bits of text all over the place and has special headers - you should do a compressed ASCII save before attempting to transfer any such file to ensure that the text sequence is correct. If in doubt as to the format used by your word processor, just have a quick look at the file with SUPERZAP.

The second basic step is the actual file transfer from TRS80 to IBM. There are three ways of doing this - creating a disk in a format recognisable to an IBM machine on a TRS80, uploading your file to a bulletin board from your TRS80 and downloading it onto an IBM, or physically connecting the two machines via an RS232 interface. I've only done disk transfers myself, but I'll talk about the other two methods as well.

There is a program called HYPERCROSS that will create IBM readable disks on TRS80 machines, but to create directly readable disks you will need double density and a double sided drive. If you have one of these, you've got no worries - the program is available for all models of the TRS80 and I've used it extensively without any problems whatsoever. If you don't have a double sided drive and double density (or can't put the bite on someone who has), you may need to get a bit sneaky. Whilst IBMs will only normally read IBM disks, there are utilities available (i.e. PC-ALIEN and MEDIA MASTER) which will read other disk formats (although not standard format TRS80 disks). They will, however, read some CP/M formats which are only single sided and you may be able to use HYPERCROSS to produce these on your machine.

The second transfer method of uploading a file from your TRS80 to a bulletin board and downloading to an IBM requires a modem at both ends, but is relatively straightforward. This is probably the simplest and most convenient of all three methods, provided that you've got access to a modem and a (preferably private) bulletin board area. Of course, if you know the Sysop well enough, you could always ask them to put the file up on the bulletin board directly - that way you only need the modem at the IBM end.

The final transfer method of physically connecting the machines via an RS232 interface is the least convenient but requires the least resources. You would need an RS232 interface and a (probably custom) cable to actually connect the machines; then you'd need to write a short program on both machines (or use a proper modem program after making sure that they both support the same transfer protocol). However, the advantage of the direct connection is that you could probably support higher baud rates than you'd get via a bulletin board and that you aren't limited in the volume of data you can transfer in a single session - if a bulletin board throws you off after half an hour or limits the disk space available to you, it is going to cause problems if you want to transfer large volumes of data - particularly if they're spread over a lot of small files.

The third stage of the transfer process - what you do with the file once it's on the IBM - depends on whether you've transferred a raw word processing file or a formatted print file.

As I mentioned earlier, if you transfer a formatted print file you could just simply do a PRINT from MS-DOS. However, this might not have the desired result. When your TRS80 word processor is sending output to the printer, it should be sending the appropriate control codes for page throws, bolding, underlining, etc. to your printer. If the printer on the IBM uses different control codes, there's no telling what it's likely to do when it gets those TRS80-generated control codes - but you can bet that it won't be what you want it to do. There are three ways around this problem (aside from not using any special print features in the first place). Firstly, you could use a printer driver on your TRS80 for the printer that's connected to the IBM (so the correct control codes are already there). Secondly, you could edit the control codes in the file via a text editor and then PRINT from MS-DOS, or (thirdly) you could load the file into an IBM word processor and use its print facilities and printer driver - this is probably easiest, although you'll still have to take out the TRS80-generated control codes and fiddle with the margin and page length settings.

The other limitation of formatted print file transfer is that if you want to amend the text once it's on the IBM (other than to just alter some control codes) you have to use a pure text editor or a word processor that will read simple ASCII format with the justification and other features disabled. Because the file probably already has page headers and numbers and the text is justified, you can't upset your word length within a line or alter the number of lines within a page - if you do you will stuff up the justification within the line and/or your page throws, so you'll get short lines in the middle of a paragraph and/or page headings halfway down the page. If you just want to print your document on a laser printer and you're not using any fancy printer commands, this is fine. Otherwise, you should have done a raw word processing file transfer.

To understand what happens when you do a raw word processing file transfer, you need to know something about how word processors store their documents in files. Advanced TRS80 word processors such as Lazywriter and Allwrite store files in ASCII format as a continuous string of text and format them on the screen. This screen format does not necessarily resemble what comes out on the printer because the print formatting is controlled by embedded codes in the text file (e.g. >lm+5 or ;in+5 in Lazywriter and Allwrite to move the left margin in five characters). IBM word processors, however, try to emulate on screen what the printed output will look like - hence the term WYSIWYG (or What You See is What You Get). To do this, they normally don't display the commands that do the actual formatting (although they normally have an option which allows you to see the formatting commands). In either case, however, the word processor is inserting 'soft' line feed/carriage returns in the text according to its own software logic. The only line feed/carriage return actually stored in the text file is the one at the end of a paragraph.

Therefore, to use your favourite IBM word processing program on your TRS80 text files, you have to do a raw file transfer and then convert the embedded format codes into the (non-displayed) IBM word processor format codes. This is not particularly difficult but it is very time consuming if you use the IBM word processing program, especially if you use hanging indents for points and/or tabs for tables. You could write a small program to do it, but this could be quite tricky because you would probably need to use variable record lengths - if you tried to use fixed record lengths you would end up with expanded or truncated records (unless the length of the TRS80 and IBM format control codes were the same, which is pretty unlikely).

Of course, once you've carried out this conversion, it's just as difficult to go back the other way - so don't think that you can just keep updating the same document on both machines, because it just doesn't work that way.

Finally, I've tried to outline the basic processes involved in transferring files from the TRS80 to IBMs to allow the unwary to avoid some of the traps that I found out about the hard way. I suppose that the basic message is that it can be done, but you need to have a damn good reason for doing it or it simply isn't worth the hassle. Furthermore, regardless of the way various IBM adherents might sneer at the older 8 bit machines, the reality is that the USABLE functionality of the TRS80 machines relative to their cost is a damn sight better than that of the IBMs. With the kind of configuration I've got at home (based on my trusty old System 80), I'd have to spend at least ten times what it's worth to upgrade to an IBM system that's SIGNIFICANTLY more useful to me. So, to quote a hackneyed but true phrase - "When you're on a good thing, stick to it!".

-----<BNB>-----

QUICK REFERENCE CARDS

by John Bird

This idea was inspired by reading the manual for the "ALLWRITE" word processor program. Quick reference cards are provided with the program to label the number keys because they can be used as multi-function keys in that program.

On a Model I keyboard, the keys are set into a black escutcheon which forms the main working area of the grey computer case. It is obvious that you can slip a piece of paper between the number keys (above QWERTY) and the black escutcheon. It is not as obvious that you can also slip a piece of paper into the gap where the black escutcheon meets the grey case.

Take a strip of thick paper (or cut a piece from a manila folder) 240mm long and 35mm wide. Form it into a 240mm long "U" shaped piece by bending down along a line 10mm in from one long edge and then bend along another line 5mm in from the other long edge. The 10mm tab fits between the keys and the escutcheon, the 5mm tab fits between the escutcheon and the case. This leaves an area 240mm long by 20mm wide that forms a notebook area to label the number keys or record other reference information.

Make several labels for differing applications and write the application name along the 10mm tab.

You could make a larger label the full width of your computer with tabs above the number keys and also above the numeric keypad. System 80 owners could get a label 280mm by 28mm (start with a strip of paper 280x43). I presume the same concept can be extended to Models 3 and 4.

-----<BNB>-----

Wak's Column

by Warwick Sands

Firstly I must apologise for the non-appearance of this column over the last couple of months. What with getting NewDos86 ready for distribution in England and other duties has kept me quite busy. While this was being done a couple of other problems surfaced and the zaps below are to fix these.

This month I want to talk about the use of filters and introduce another NewDos86 utility called, FILTER/CMD. But first the zaps.

Processing TYPE/CMD

These zaps to TYPE/CMD are purely cosmetic. My thanks to Ray Whitehurst of Melbourne for being the first to point it out to me. If the specified printer driver is not found, then when the error is reported, rubbish may appear on the screen.

The changes are small and are detailed below:

Sector number 3

This changes the version number from x.1 to x.2

Change from:-

E0 7273 2076 6572 7369 6F6E 2031 2E31 2020

To:-

E0 7273 2076 6572 7369 6F6E 2031 2E32 2020

Sector number 6

The garbage on the screen was caused by the filename being printed from the wrong place in memory. Although the data was present, it was being overwritten by stack data in some circumstances. This zap fixes the problem and introduces another in that we now need to move to the next video line.

Change from:-

20 EBCD 6744 D12A 6552 0600 CD24 44C2 0944

30 E1C3 1B5C F521 935B CD67 4421 C75B CD67

40 44 F1 F5CB FFCD 0944 2A20 40E5 2101 0200

To:-

20 EBCD F876 D12A 6552 0600 CD24 44C2 0944

30 E1C3 1B5C F521 935B CD67 4421 C652 CDF8

40 76 F1 F5CB FFCD 0944 2A20 40E5 2101 0200

Sector number 34

This routine resides at 76F8H, it prints data then a carriage return.

Change from:-

B0 0000 0000 0000 0000 0000 0000 0000 0000

C0 0000 0000 0128 0077 3A40 38CB 57C0 CD22

To:-

B0 0000 0000 0000 0000 0000 0000 CD67 443E

C0 0DC3 3300 0128 0077 3A40 38CB 57C0 CD22

Processing SYS1/SYS

This is a small change to make FILTER/CMD work. The change forces either an XOR A : RET or an OR A : RET depending on the setting of SYSTEM option AG, which enables/disables the BREAK key. FILTER uses the Carry flag to indicate the success/failure of a search. If DEVICE has not been installed then the Carry flag may be set even though the search has failed.

Model 1

Sector number 0

Change from:-

D0 2806 2A5B 4022 1640 CB6F 21B0 4522 1343
E0 3EC3 2802 3EC9 3212 43

To:-

D0 2806 2A76 4422 1640 CB6F 2160 4A28 022E
E0 1722 1343 3EC3 3212 43

Models 3/4/4P (including HardDisk version 2.5)

Sector number 0

Change from:-

D0 2806 2A76 4422 1640 CB6F 2146 4522 7944
E0 3EC3 2802 3EC9 3278 44

To:-

D0 2806 2A76 4422 1640 CB6F 21FE 4728 022E
E0 8722 7944 3EC3 3278 44

Processing DEVICE/CMD

This change is required to allow FILTER/CMD to function correctly. The search routine didn't function correctly under some circumstances.

Sector number 3

Change from:-

70 237E B7C8 BA23 7E23 2323 20F2 BB20 EF 5E

To:-

70 237E B7C8 BA23 7E23 2320 F3BB 20F0 23 5E

Processing SYS14/SYS

This change is required to allow FILTER/CMD to function correctly. The search routine for a specified *ab name didn't function correctly under some circumstances.

Sector number 0

Change from:-

20 ODCA AE4E ODCA 3A4F 3E2A C9AF D901 03F0

To:-

20 ODCA AE4E ODCA D851 3E2A C9AF D901 03F0

Sector number 4

Change from:-

D0 5100 0000 0000 0000 0000 0000 4FE1 7CB5

To:-

D0 5100 0000 D5CD 614D D1C3 3A4F 4FE1 7CB5

The following zap is to fix a problem in INSTR(!, the bit-map testing routine.

Processing sys26/sys

Sector number 0

Change from:-

```
D0 0123 2929 29C5 0600 09C1 EBD5 1E08 7E28
E0 07 1F 1C17 1D0D 20FB 4843 D1A9 1730 0E13
To:-
D0 0123 2929 29C5 0600 09C1 EBD5 CDDC 6E00
E0 00 1F 1C17 1D0D 20FB 4843 D1A9 1730 0E13
```

Sector number 4

Change from:-

```
F0 0000 0000 0000 00 00 0000 0033 0202 006A
To:-
F0 1E08 7E00 0E08 C9 00 0000 0033 0202 006A
```

Some notes on the format of the zaps. The actual bytes to change are **BOLDED**. To do the zaps I suggest that you use the following method:-

```
Reset all the update flags with a PROT,:x,RUF
Apply all the patches to the disk on drive :x
Set the datestamp of the changed file with a
NEWDATE :x,=07/13/88 i s
```

After making certain that there are no problems with the zaps, copy over the changes to your other SYSTEM disks by

```
COPY :x,:y,,CBF,NFMT,DFO,UPD
```

-----oOo-----

On to the concept of filters. What are filters used for? According to Roy Soltoff "Device I/O can also be massaged with Transformation functions, called filters". That mouthful simply means that you can alter data being sent to the video, or typed in from the keyboard, in a selective manner.

A simple example of a filter is the ND86 BQ SYSTEM option. When set to Yes, a filter routine is permanently installed that causes a CHR\$(10) to be sent after every CHR\$(13) going to the printer. Another simple filter is activated by the BP SYSTEM option. This causes a filtering routine to be installed that converts a Top Of Form (CHR\$(12)) into a series of linefeeds (CHR\$(10)'s).

A more complex example of a filter is the Margin Routines activated by the BQ system option. This keeps track of the number of characters and lines printed and inserts spaces and linefeeds to format the text.

When I started to write ND86, I had to make a decision on how to implement these options. I decided to use SYSTEM options as this was the most memory efficient way to implement these kinds of often used filters. The disadvantage is that the only way to remove these 'filters' is to set your SYSTEM options accordingly and re-boot the computer.

FILTER/CMD allows you to install removable filters. The routine to remove filters will probably appear next month. FILTER/CMD uses DEVICE/CMD to create a high memory Device Control Block to hold filter information.

The syntax for the filter command is:-

```
FILTER,*ab,[WITH,]*xy,[USING,]filespec,parameters
  where *ab is the DCB name to filter
        *xy is the name of the filtering DCB
        'filespec' is the name of the routine that does the filtering
        'parameters' are those required by filespec
```

The WITH and USING parameters are optional. A practical example is:-

```
FILTER,*PR,WITH,*TR,USING TRAP (CHAR=10)
```

This command filters the printer (*PR) with a routine, TRAP/FLT, via a DCB called (*TR) which provides a linking mechanism between TRAP and *PR. FILTER is a PSF type file, therefore if you are installing it on your working disk, after copying it over, you MUST give it the PSF attribute:-
ATTRIB FILTER/CMD,PSF=Y

FILTER performs the following functions: If required it calls DEVICE/CMD to create a DCB called *TR in high memory. It then loads the file TRAP/FLT and passes control to it. TRAP calls some routines in FILTER to install the actual filter routine in High Memory. TRAP then parses the command line (the CHAR=10 indicates that no LineFeeds are to be sent to the printer) and installs the parameters into the routine.

If called from BASIC the routine will probably overwrite already existing strings -- in all cases string space will be reduced to 1 byte. So after you install a filter, execute either a CLEAR statement or a CMD"F=SASZ" to reset your string space to the desired size. Using CMD"F=SASZ" will allow you to keep strings already defined.

As I write various /FLT files I will put them into the Public Domain section of the Club's Library. The only file written thus far is the TRAP routine. I would like to look in detail at it to give you some idea on how you can write your own filters.

FILTER/CMD resides from 6B00H-6EFFH. I suggest that you ORG your filter routines from 6F00H-7200H. This will allow you to install them from within BASIC.

Before we look at the actual filter code, we will examine the layout of the DCB that ties the filtering routine to the FILTERed DCB. This is 17 bytes long and has the following format:-

Byte	Function
0	The TYPE byte. This is a flag byte. The meaning of the various bits is given later on.
1,2	This holds the address of the filtering routine.
3,4	This is the size of the DCB and is used by DEVICE when clearing DEVICES.
5,6	This holds a pointer to the i/o byte in the BIN. The i/o byte is discussed later.
7,8	Pointer to the DCB being filtered.
9,10	High Memory setting after the BIN, DCB and filter code has been installed.
11,12	The High Memory setting before the BIN, DCB and filter code was installed.

13,14 Pointer to the instruction that JP's to the next routine in the chain. This allows you to attach the filter to a different DCB without having to remove and then reload it.

15,16 A pointer for your use. This allows the filter data to be changed at will. It can point at the actual location to change, as in TRAP/FLT, or it could point to the start of a data table.

```
00100        TITLE    <TRAP/FLT>
00110 ;
00120 ;        W. S.   &   D. Sands
00130 ; June 26, 1988
00140 ;
00150 @HIGH$ EQU       6C00H               ;high memory pointer
00160 @IOBYTE EQU       6C02H               ;copy of i/o byte
00170 @PARAM EQU       6C03H               ;parser routine
00180 @RELOC EQU       6C06H               ;relocates code
00190 @STORE EQU       6C09H               ;sets the vectors
```

Just a couple of points on the above vectors.

@HIGH\$ stores the highest memory address that you wish to reserve for your filter code. As with DEVICE/CMD, FILTER/CMD tries to organise memory so that it can at a later time be released. This may involve moving the location of the buffer used by the Printer Spooler. Hence we need this local high memory pointer. FILTER uses this value to determine where the new Spooler Buffer should reside.

@IOBYTE holds a copy of the i/o byte of the DCB we are filtering. This will be discussed a little later.

@PARAM allows you to call the DOS parser routine, without having to worry about whether the filter is being installed on a Model 1,3,4/4P.

@RELOC moves your filter code from its nominal load address into high memory. A few rules are required. We will discuss these later.

@STORE stores a some values in the FILTER DCB and stores the address of the filtering routine into the DCB to be filtered, this is what activates the filtering mechanism.

```
00200 DOS_ERR EQU       4409H
00210 ;
00220        ORG       6F00H               ;BASIC #1 overlay
00230 ENTRY
00240 ;HL -> ENTRY
00250 ;IX -> DCB
00260 ;BC -> DCB address of DEVICE to Filter
00270 ;DE -> Command line following
00280 ;
00290 ;Initialisation code must:-
00300 ;        1) Move the filter code to himem {use CALL @RELOC}
00310 ;        2) Initialise the BIN and DCB
00320 ;        3) Set the high memory pointer {LD (@HIGH$),himem}
00330 ;
00340        PUSH       DE               ;save command line
```

```

00350      PUSH    BC                ;save DCB to filter
00360      LD      DE,ENTRY          ;called by FILTER ?
00370      SBC     HL,DE             ;should be Z if so
00380      JP      NZ,ENT_ERR        ;if called from DOS
00390 ;

```

Since this file modifies memory based on the contents of the above registers, and since the filter filename can be invoked from DOS, the filter routine must make the above test. FILTER loads the file and HL holds the entry address. However when the filename is invoked from DOS or as a DOS Command, HL will be pointing at the DOS command line.

The byte before the DCB (IX-1) contains some flags that determine what the filter initialisation routine has to do. The functions are as follows:-

bit	Function
7	It is a *ab! bin/dcb combination
6	The filter code has been installed
5	The filter is active
4-2	Reserved
1	The bin has a name associated with it
0	The bin is in use - DEVICE/CMD

Bits 7 & 0 are set by DEVICE when creating the BIN. Bits 6,5 & 1 are of interest to us. Using these bits as flags allow us to change the character we wish to trap without having to remove the filter first.

```

00400      BIT     6,(IX-1)          ;already installed ?
00410      CALL    Z,INSTALL        ;install it if not
00420 ;
00430      POP     BC                ;get DCB to filter
00440      BIT     5,(IX-1)          ;filter in action ?
00450      CALL    Z,SET_UP          ; if not set it up

```

Now that the filter has been installed and set up, we can parse the command line following the filespec.

```

00460      POP     HL                ;get the command line
00470      LD      DE,PARAMS          ;go see if char specified
00480      CALL    @PARAM            ;check params
00490      JP      NZ,BAD_PRMS        ;if error
00500      LD      BC,000EH           ;the character
00510 TRP_CHR EQU    $-2
00520      LD      A,B               ;check if numeric or string
00530      OR      A
00540      JR      Z,NUMBER           ;go if number
00550      DEC     BC                ;check if prev chr a string
00560      LD      A,(BC)
00570      CP      '"'
00580      JR      NZ,BAD_PRMS
00590      INC     BC
00600      LD      A,(BC)            ;get char
00610      LD      C,A               ;into C
00620 NUMBER
00630      LD      L,(IX+15)          ;get the address of the
00640      LD      H,(IX+16)          ; trap character
00650      LD      (HL),C             ;store the character

```

```

00660      RET                      ;exit back to FILTER/CMD
00670 ;
00680 SET_UP
00690 ;Since this is an output filter DEVICE we must make certain that
00700 ; the DCB to be filtered has output capability.
00710      LD      A,(BC)           ;get the FILTERED DCB
00720      AND     2                ; TYPE byte and check
00730      JP      Z,BAD_DCB       ;if not an output device
00740      LD      L,(IX+5)         ;get ptr to i/o byte
00750      LD      H,(IX+6)         ; into HL
00760 ;The Bin i/o byte is used by ROUTE to determine validity
00770 ;of the requested ROUTE command.
00780 ;The Bin i/o byte must be set up as follows:-
00790 ;      bit      function when set
00800 ;      7      can source data
00810 ;      6      can accept data
00820 ;      5      MM= device
00830 ;      4      CLEAR status
00840 ;      3      A ND86 created Device
00850 ;      2      can be linked to a disk file
00860 ;      1      capable of accepting data
00870 ;      0      capable of sourcing data
00880 ;
00890 ;Since we are filtering another DCB with this DCB, the
00900 ; bin i/o bytes should be identical with one exception.
00910 ; Bit 2 must be reset, as this indicates that the DCB is
00920 ; associated with a disk file and that the FCB address is
00930 ; stored in bytes 3&4 of the DCB. This is not true in the
00940 ; case of a filter and all kinds of troubles could result
00950 ;
00960      LD      A,(@IOBYTE)       ;get the i/o byte
00970      AND     0FBH             ;lose bit 2
00980      OR      01001000B        ;set bits 6,3
00990      LD      (HL),A          ;set up i/o byte
01000 ;
01010 ;Must also set the DCB+0 byte to indicate what
01020 ;functions this filter can support.
01030 ;      bit      function when set
01040 ;      7      Open Disk File
01050 ;      6      Filtering Device
01055 ;      5-3      Reserved
01060 ;      2      can handle CTRL requests
01070 ;      1      can handle PUT requests
01080 ;      0      can handle GET requests
01090 ;
01100      LD      A,(BC)           ;reget the TYPE byte
01110      AND     7                ;isolate data
01120      OR      40H              ;set bit 6
01130      LD      (IX+0),A         ;set it up as a filter
01140 ;
01150 ;We must also flag the fact that the filter is installed
01160 ; and active.

```

Don't worry about the missing lines 1170-1270. In the source code they repeat the data on the byte (IX-1).

```

01280 ;
01290 ;We set the active bit here and the installed bit in INSTALL
01300 ;
01310         SET      5,(IX-1)           ;flag routine active
01320 ;now that all tests have been passed we perform the actual
01330 ; filtering by calling a routine that stores the DCB to filter in
01340 ; (IX+7,8) and also updates the address in the routine (pointed
01350 ; to by (IX+13,14).
01360 ;BC contains the address of the DCB to filter.
01370 ;
01380         JP        @STORE              ;go update entry address
01390 ;
01400 ;
01410 INSTALL
01420         LD        A,(IX-1)           ;check that the DCB is of
01430         AND        81H                ; the correct type
01440         CP        81H                ;bits 7 & 0 must be set
01450         JP        NZ,BAD_BIN
01460         LD        HL,ENDFLTR         ;end of pgm
01470         LD        BC,LENFLTR         ;length of filter code
01480         LD        DE,TABLE           ;relocation table
01490         CALL      @RELOC              ;relocate the code

```

The relocation routine is based on one I saw in Northern Bytes and I thank Jack Decker for this. We will discuss the rules a little later.

```

01500         SET      6,(IX-1)           ;flag filter routine installed
01510         PUSH      DE                 ;do highmem init
01520         RET                          ;from installing filter

```

It is assumed that the high memory initialisation is at the start of the highmem code. Thus we can exit to the HM routine by doing the above.

```

01530 ;
01540 BAD_DCB
01550         LD        HL,FCB$           ;give the message
01560         JR        PR_ERR             ;display error
01570 BAD_BIN
01580         LD        HL,BIN$           ;tell the world
01590 PR_ERR
01600         CALL      4467H
01610         JR        NO_FLT            ;give DOS error
01620 ENT_ERR
01630         LD        HL,ENTRYS$
01640         CALL      4467H              ;display message
01650         LD        A,2AH              ;ILLEGAL DOS FUNCTION
01660         DB        01H
01670 BAD_PRMS
01680         LD        A,2FH
01690         DB        01
01700 NO_FLT
01710         LD        A,8                ;DEVICE not AVAILABLE
01720         OR        A
01730         JP        DOS_ERR
01740 ;
01750 BIN$     DB        10,31
01760         DB        'The bin is of the incorrect type -',10

```

```

01770      DB      ' use a different name',13
01780 FCB$   DB      10,31
01790      DB      'The FCB is of the wrong type -',10
01800      DB      ' an output FCB is required.  E.g. *PR',13
01810 ENTRY$
01820      DB      10,31
01830      DB      'TRAP/FLT must be installed with ',10
01840      DB      ' FILTER/CMD ',13
01850 ;
01860 PARAMS
01870      DEFB      'CHAR      '      ;parameter name
01880      DEFW      TRP_CHR      ;parameter address
01890      DEFW      0
01900 ;

```

See the section in the ND86 manual on the TRSDOS parser for a discussion on how to build the parameter table.

```

01910 ;***** RELOCATION TABLE *****
01920 ;Must be located prior to "START" label of main program!
01930 TABLE
01940      DEFW      CHG_1-OFFST+1      ;relocate the address
01950      DEFW      CHG_2-OFFST+1
01960      DEFW      CHG_3-OFFST+1      ;fix load address
01970      DEFW      CHG_4-OFFST+1      ;fix xfer address
01980      DEFW      0
01990 ;
02000 ;end of table
02010 ;

```

The table above contains pointers to all the addresses that must be changed when the code has been moved into high memory. The entries must point to the first byte to be changed, i.e. the actual memory reference. This explains the +1 in the table since the actual LD code (21H) must not be altered.

All references to addresses within the code must have the -OFFST appended to them. The principle of the routine is based on the use of offsets. The table of pointers actually hold the distance into the routine of the address to change. Thus CHG_1-OFFST+1 has a value of 1. When the code is moved into high memory a variable points at the first byte of high memory code. This is added to the value of the table entry to find the address of the instruction to change. The routine then extracts the address out of this location. But note that the instruction also has the -OFFST in it. So the address retrieved is also an offset into the program. This value is added to the start address and the total is stored back into the instruction.

It is harder to explain it, than it is to use it.

```

02020 OFFST   EQU      $
02030 ;
02040 STRTFLTR
02050 CHG_1    LD        HL,FILTER-OFFST ;get the entry address
02060          LD        (IX+1),L        ; and set up the filter
02070          LD        (IX+2),H        ; entry address
02080 CHG_2    LD        HL,$NAME-OFFST ;top high mem limit

```



```

02090      DEC      HL          ;adjust it and
02100      LD        (@HIGH$),HL ; reset topmem ptr
02110 CHG_3  LD        BC,TRAP-OFFST ;points to trap chr
02120      LD        (IX+15),C    ;store in the
02130      LD        (IX+16),B    ; DCB
02140 CHG_4  LD        BC,ACCEPT-OFFST+1 ;set the Xfer address vector
02150      LD        (IX+13),C
02160      LD        (IX+14),B
02170      RET
02180 ;
02190 $NAME

```

If you wish to have an ASCII name displayed when a yet to be written routine examines DCB usage: store the name here, terminate it with a 03H and set bit 1 of (IX-1).

```

02200 FILTER JR        NZ,OUTP1    ;go if not put
02210      PUSH      AF            ;save chr & flags
02220      LD        A,C          ;get chr
02230      SUB       0EH          ;chr to lose
02240 TRAP   EQU       $-1
02250      JR        NZ,OUTP1    ;if not the chr to lose
02260      POP       DE           ;clear stack
02270      RET
02280 OUTP1
02290      POP       AF            ;get flags
02300 ACCEPT JP        0000H      ;go to the next routine
02310 ENDFLTR EQU      $-1        ;last byte to save
ENDFLTR is used to identify the last instruction to move into high memory.
02320 LENFLTR EQU      $-STRFLTR
LENFLTR sets the number of bytes to move into high memory.
02330      END      ENTRY

```

-----<BNB>-----

WHAT'S IN THE CLUB LIBRARY?

by John Bird

The club library has a good collection of books, magazines and also newsletters from other clubs to provide you with information. However, the collection is not complete. The following magazines are popular with other members. If you have any magazines that are no longer useful to you, why not consider donating them to the club library for the benefit of other members. Older (previous) issues to those listed below would provide valuable introductory material for newer members..

Micro 80 (Australian)	14 issues from 9/80 to 8/84
80 Micro (American)	21 issues from 8/80 to 5/85
Softside	10 issues from 3/80 to 7/83
Aust. Personal Computing	23 issues from 1/80 to 11/85
Creative Computing	5 issues from 6/79 to 12/82
Practical Computing	6 issues from 10/81 to 2/83
Kilobaud	8 issues from 2/77 to 4/80
Byte	22 issues from 1/77 to 10/86
Your Computer	15 issues from 5/81 to 6/85

-----<BNB>-----

NEWS from the PAST

by Peter Goed

I was leafing through some material, just donated to the club library by Jim Forayter, when the first issue of Tandy's 'Microcomputer Newsletter' caught my attention.

It is now just over ten years ago that the Expansion Interface and Level 2 Basic was released by Tandy. I could hardly believe the price of some of the items in this newsletter, so thought that reproducing it in BNB might remind some of you older members, just how much your computer cost you and some of you newer owners of machines will get an idea of the expense involved in being a computer hobbyist back in 1978. Keep in mind that these prices are in \$US and you can add 50% to this figure to get the \$A cost.

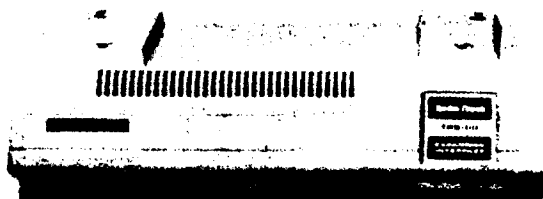
Radio Shack

**MICROCOMPUTER
NEWSLETTER**
SPECIAL EDITION — January, 1978
Radio Shack One Tandy Center Fort Worth, Texas 76102

PRE-ANNOUNCEMENT NOTICE FOR TRS-80 OWNERS

Radio Shack will introduce its new line of microcomputer products to the general public in February. We plan to begin shipping new hardware and software this March.

Since you now own a TRS-80 or have one on order, we want you to have an early opportunity to place your order for "first come — first served" delivery on this new line of products. Here's what you can order right now.



Expansion Interface — lets you add 16 or 32K RAM, dual cassettes, mini-floppy disks, line printer.

RADIO SHACK MICROCOMPUTER PRODUCTS

Any of the following four micros (CPU only) may be ordered through your local Radio Shack, and will be shipped complete from our factory.

TRS-80 COMPUTER

Includes 4K RAM and LEVEL-I BASIC.
26-1001 400.00

Includes 16K RAM and LEVEL-I BASIC.
26-1003 690.00

Includes 4K RAM and LEVEL-II BASIC.
26-1004 499.00

Includes 16K RAM and LEVEL-II BASIC.
26-1006 789.00

TRS-80 VIDEO DISPLAY

199⁰⁰
26-1201

A 12" display designed especially for the TRS-80 system. Displays 16 lines of 64 characters, also software selectable to 32 cpl.

REALISTIC CTR-41 CASSETTE RECORDER

49⁹⁵
14-841

FREE! The CTR-41 Cassette Recorder is included at no charge when a computer and video display are ordered together.

Your TRS-80 system can be expanded to suit your specific needs with peripherals and added software packages. Take your pick of the following "goodies":

16K MEMORY KIT (RAM)

290⁰⁰
26-1101

With this kit you can convert your 4K TRS-80 to 16K of RAM. The EXPANSION INTERFACE (26-1140) will accommodate one or two additional 16K kits which will make possible a 32K or 48K RAM TRS-80.

LEVEL II BASIC KIT (ROM)

99⁰⁰
26-1120

If you own a LEVEL-I BASIC micro, this ROM kit will step you up to the more powerful LEVEL-II BASIC. Improved graphics, print formatting, text editing, key-board roll-over (you can

type as fast as you like and not lose a single letter), and faster (500 baud) cassette transfer rate with named files are some of the features. A manual is included so you can fully utilize LEVEL-II. Also included is a conversion program which will allow you to convert your LEVEL-I programs and cassettes to LEVEL-II.

TRS-80 EXPANSION INTERFACE

299⁰⁰
26-1140

This is the key to the expandability of the TRS-80. The Expansion Interface (which usually requires LEVEL-II BASIC) enables you to add the following to the TRS-80:

1. Additional RAM (16 or 32K)
2. Dual cassettes (program selectable)
3. Mini-floppy disks. It will handle up to four minis!
4. Our new line printer (Centronics parallel port)

The EXPANSION INTERFACE includes a real time clock, a space for an additional PC board (to add whatever?), and continuation of the TRS-80 bus. The EXPANSION INTERFACE will give you the capability to fully utilize your TRS-80 system.

Continued on Next Page

KEYBOARD REPAIRS

by John Bird

Those people who are having problems with their keyboards such as stiff keys or keys that don't work (i.e. you type certain letters but nothing happens) should note that it is a relatively easy matter to install a new keyswitch.

For those people who haven't got access to a soldering iron or are just too timid to open up the innards of their computer, I intend running a keyboard repair session immediately before the next meeting.

Ring John on 261-1952 at home (or 225-1509 at work) to let him know what type of machine you have so that he can arrange stock of the right type of switches for the various machines.

Country members who are having key problems that cannot be fixed locally may contact John to arrange repairs to their keyboard.

-----<BNB>-----

TEST FOR MODEL I LOWER CASE

by John Bird

I can't claim credit for this one but neither can I remember who it was that did show some of us the answer before the May meeting.

Question: Is there a simple way to test if the lower case modification has been fitted to a Model I (or System 80); presuming the unit is not connected up to disk drives?

Answer: While in Basic, CLEAR the screen, then type POKE 16000,97 and ENTER. About half way down the left hand side of the screen, you should get a lower case letter "a".

Reason: Address 16000 is a convenient number to remember that gives an address somewhere in video memory. Value 97 is the ASCII code for lower case "a".

-----<BNB>-----

AN excerpt from a New Zealand member's letter: The extra \$5 for my subs is en-closed, along with the questionnaire I received with the latest newsletter yesterday. Keep up the good work with the club -- there is NOTHING remotely like it available here in Wellington so the information you guys publish is pretty valuable to me -- even though at this stage I do little more than read it and file it away for future reference. [Nice to know our efforts are appreciated -- thanks for the pat on the back, Murray.]

Remember that if you want to see your name in here, you had better get fingers to keyboard and write an article. I was hoping that someone would take me up on the offer of other methods of recovering corrupted disks, surely there MUST be others out there that do this in a different way to that described in last month's BNB. If nobody writes me an article on this, you might have to suffer one from me instead.

DEADLINE FOR AUGUST ISSUE IS 8pm 17th AUGUST

TEST YOUR COMPUTER SYSTEM

by John Bird

There are various diagnostic disks around to test your computer system when it is working. But how do you test your system when it is not working properly and can't even load a disk.

Try the following:

STEP 0: Remove all diskettes

STEP 1: While holding the <BREAK> key down, turn everything ON.

i) If the power LED on the keyboard isn't illuminated, then you haven't got power. Check it out.

ii) if the disk drives spin for much more than five seconds (they shouldn't spin at all), then you have one end of the drive cable upside down.

iii) if the monitor does not show the message "MEM SIZE", then the monitor isn't connected or powered up.

STEP 2: press the <ENTER> key. You should now be in Basic.

STEP 3: type ?MEM and hit <ENTER> you should get 48338 on a 48K machine.
31956 on a 32K machine.

If the answer is 15570, then the Expansion Interface isn't connected or is not powered up

STEP 4: type POKE 14304,1 and hit <ENTER>. For the next second, all disk drives should spin and Drive 0 will light up.

STEP 5: type POKE 14304,2 all drives spin but only Drive 1 lights up

STEP 6: type POKE 14304,4 all drives spin but only Drive 2 lights up.

POKE 14304,8 for Drive 3.

POKE 14304,15 makes them all light up at once!

STEP 7: type PRINT PEEK(14312) to test your printer.

An answer of 255 means the printer isn't connected or powered up.

An answer of 143 means the printer is offline.

An answer of 207 means out of paper or faulty.

An answer of 63 means the printer is ready to go

Try POKE 14312,67 (this might have to be followed by POKE 14312,13).

This should print the letter C on the printer.

Try the above test now while your computer is healthy so that you can tell where the difference is when your system next plays up.

I checked the above on my 48K Model I. System 80 users will get a different message at Step 1 iii) and should use OUT 253,67 to test their printer. Don't panic if your memory size is one or two bytes different in size, that indicates the different ROM's that are about. Model 3 users should use OUT 244,1 at Step 4 (and OUT 244,2 etc for Steps 5&6).

Postscript: After writing this article but before submitting it to the editor, I had a problem with disk drive 1. When in DOS, I kept getting the message "device not available". Running the above test would cause the drive to light up but it took a while for me to realise that the drive wasn't spinning - it was Drive 0 that I heard spinning when Drive 1 lit up. --Murphy's Law--.

-----<BNB>-----