INDEX FOR VOLUME 6 - 19

NORTHERN BYTES

Volume 7 Number 1

GREETINGS! Welcome to the first "official" issue of NORTHERN BYTES Volume 7. This issue is being produced relatively soon after Volume 6, Number 8, which explains the small number of "letters to the editor" in this issue.

Speaking of Volume 6, Number 8, how did you like our "new look"? We've heard your complaints about the print quality in previous issues (which upset your editor a great deal as well) so we are now sending NORTHERN BYTES out to be printed commercially. The paper's a little thinner, but I think the print quality is much better overall, and I hope that you're pleased with it. One thing I should mention is that because of this new process, we have to specify ahead of time how many issues of NORTHERN BYTES that we want printed, and when they're gone, they're gone! And, in any case, we can't get back issues reprinted as easily as before. So, if you want back issues, please consider getting them now or you may be disappointed (or at least have to wait for them) at a later date. If you've purchased a back issue of Volume 5, Number 4 you may have a bad copy and if so, you will be entitled to a free replacement. See T EXTERMINATOR column below for details.

Now for a few notes of interest to specific segments of our readership:

CREDIT CARD "SUBSCRIBERS": Please keep track of the expiration date of your credit card (it should appear on your NORTHERN BYTES mailing label) and send us your new expiration date (and card number if it changes) at renewal time. You may call the toll-free order line to leave this information, simply tell the operator that you wish to sign up to receive NORTHERN BYTES, then tell her to note that you're supplying a new credit card expiration date in the message space on her order form. Or, if you want to help us save a little money, jot your name and the new expiration date on a post card and send it to us. If your credit card expires and we don't have the new information, we can't continue to send you future issues of NORTHERN BYTES.

COMPUTER CLUBS AND USER GROUPS: If you missed our front ge announcement in the last issue and you have not been sending exchange copies of your group's newsletter to the correct address, it's probably too late for you to get reinstated to our published listing of clubs and user groups. But, if you want to find out for sure, call me at (908) 632-3248. If you've been sending exchange newsletters right along, don't worry about it. The list of clubs and user groups will most likely be printed in our next issue.

NEWDOS/80 USERS (and former users): Are you putting off the purchase of a Model 4 (or one of its newer revisions) because u don't want to give up the convenience of NEWDOS/80, but can't ind the thought of not being able to fully utilize the capabilities of the 4? Or, did you buy a Model 4 and find yourself using the Model III mode so that you can use NEWDOS? Maybe you switched to TRSDOS 6, but find yourself daily wishing that you had some feature that was present in NEWDOS/80, but is either unavailable or sold separately at additional cost under TRSDOS 6? Well, help may be on the way. I got a call from Bob Brumley recently, and he says he is nearing completion of a Model 4 version of NEWDOS/80! This will not be just a Model 4 adaptation, either, because he plans to add many new features not found in the present editions of NEWDOS (including compatibility with TRSDOS 6 format disks).

Right now Bob would like to know what features you might want to see added to NEWDOS/80 (such as new commands, etc.). If you have any thoughts on this subject, please send them to: Bob Brumley, 8522 Alden Lane, Windsor, California 95492. Naturally, Bob can't implement every suggestion that he receives, but I'm sure he'll consider all suggestions. It's not often that potential users of a piece of software get a chance to comment on how it should operate prior to its release, and by that point, it's usually offered on a "take it or leave it" basis. So, if you have any interest at all in a Model 4 version of NEWDOS/80, put your thinking cap on and send your suggestions to Bob, as soon as ssible.

THE EXTERMINATOR

Who would think that BUGS would pop up during the winter here in the Great White North? Well, we've had a few crawl through past issues of NORTHERN BYTES, so let's get 'em...

But first of all, I'd like to call your attention to THE GREAT REPRINT MISTAKE. Due to the ineptitude of someone in our printing department (which no longer exists due to errors like this!), NORTHERN BYTES Volume 5 Number 4 was reprinted with a page 18 that actually came from Volume 5 Number 3. If you have ordered back issues of NORTHERN BYTES and a copy of Volume 5 Number 4 was included, you may have received a bad copy. If the page 18 in your copy has an article on "MODEM PROTOCOL DOCUMENTATION", you have one of the bogus issues. Please drop us a post card or letter (do NOT call the 800 number unless you are ordering something else at the same time) and we will send you corrected copy of that issue, free of charge. If you have a copy of Volume 5 Number 4, please check for this and let us know if you got a bad one. Just for your information, the correct page 18 originally contained several zaps to NEWDOS/80 version 2, a patch to Model III Disk Scripsit to allow it to read 1500 baud cassette tapes, and an editorial on the dangers of computer programs that attempt to use high resolution graphics and sound to induce hypnotic states.

Now, let's stomp those pesky BUGS. Back in Volume 6, Number 7, on page 16 we published some zaps to the Allwrite! text editor (AL/CMD). I again must re-emphasize that these zaps are untested, should be considered experimental, are not supported by Prosoft, and could possibly even cause a loss of text. Having said that, I want to present a slightly different variation of the same zaps that may, repeat may, be a bit more reliable (that is, less likely to eat your text). These zaps treat all characters except spaces as periods were formerly treated. If you installed the previous zaps, you'll have to start over with an unzapped copy of AL/CMD (version 1.12) and then apply the zaps below:

```
M1/III FRS 20,83 - M4 FRS 25,9B:  change FE 2E 28 05        to FE 20 29 05
M1/III FRS 21,36 - M4 FRS 26,CE:  change FE 2E 23 03        to FE 20 28 03
M1/III FRS 23,66 - M4 FRS 28,7E:  change 3E 2E BE 23 20 0E  to 3E 20 BE 23 28 0E
M1/III FRS 28,10 - M4 FRS 33,28:  change 3E 2E BE 20 05     to 3E 20 BE 28 05
```

If you use these zaps and have any problems using Allwrite, please be sure to try using the original unzapped version of Allwrite before contacting Prosoft. Your problems could originate with these zaps, and the folks at Prosoft would have no way of tracking them down! By the way, should you experience any unexpected difficulties while using these zaps please drop us a line, we'd like to hear about them.

Speaking of Prosoft, in Volume 6, Number 8 (page 7) we reprinted Chuck Tesler's letter on software piracy as it appeared in the Adelaide Micro-User News. Well, they misspelled Chuck's last name several times, and I didn't catch the error prior to publication. So, for the record, "Tesler" is spelled with one "s", not two. Sorry, Chuck. Also, I understand (through a telephone conversation with Ron Malo of Prosoft) that Allwrite is now sold to TRS-80 users in other countries (including Australia) provided that a prospective purchaser first obtains (from Prosoft) and signs a release acknowledging certain limitations on the amount of support he may expect to receive. As Ron explains it to me, this is due to the fact that there are certain differences in both hardware and (Disk Operating System) software as sold overseas, and Prosoft can't be expected to support every foreign variation of the TRS-80 (not to mention the TRS-80 clones that are popular overseas). Presumably the overseas user groups could help their users through any "tweaks" that need to be made. In any case, it appears that the original problem of lack of copyright protection afforded by Australian law is no longer an issue.

In the article "TELECOMMUNICATIONS NEWS" that appeared in Volume 6, Number 8, the two columns on page 19 were transposed during the paste-up process. If the article didn't make any sense to you, go back and re-read it, but when you get to page 19 just read the second column first and the first column last.

Also in Volume 6, Number 8, a minor bug appeared in the article "MODEL I LEVEL II LOWERCASE ON POWERUP". In that article a zap was given to permit the LC,(Y/N) command to work with the EPROM lowercase driver installed. Immediately following that zap, the text read, "When the above zap is used, the BASIC caps lock control is changed to POKE 16409,0 to set mixed (upper/lower) case, and POKE 16409,1 to set uppercase only (the same as under Model III BASIC)." The underlined portion of the text should have been omitted, since the memory location change is not a function of the NEWDOS/80 zap, but of the EPROM lowercase driver. The zap simply permits the correct functioning of NEWDOS/80's LC command when the EPROM modification has been installed.

Incidentally, Bob Seaborn (who assisted in the preparation of the above-mentioned article) has said that he will burn an EPROM containing the code revisions suggested in the article for $6.00 (U.S.) plus the cost of the EPROM (if he supplies it, or you can supply your own EPROM if you wish). For more information, telephone Bob at (306) 343-1305 or write to him at 2601 Dufferin Avenue, Saskatoon, Saskatchewan, S7J 1C7 or via MCI Mail (User I.D. 268-7906).

LETTERS DEPARTMENT

Reminder: Persons sending letters intended for publication should send them on magnetic media or via Compuserve [72167,161], Delphi [TASIO], or MCI Mail [109-7407] (especially if longer than a couple of paragraphs). If you are NOT using Allwrite (or Newscript) and your word processor offers the option to save your file in ASCII format, please do so (especially if using SuperScripsit!). Your cooperation in this matter will help us to bring you a better newsletter!

Date: Wed Nov 06, 1985 4:27 pm EST
From: Bob Seaborn / MCI ID: 268-7906

TO: * Jack Decker

Jack -

Here are some more zaps for Newdos 80/2.0.

This patch removes the '?' Prompt at the end of a directory page, clears the screen before the next page is displayed, and allows any key to be pressed when the next page is requested instead of only <ENTER>. <BREAK> is still used to abort the DIR command.

```
SYS8/SYS,03,86 (I)
                from  3E3F CDA5 50CD 4900 3DCA 2D40 FE0C 20F5
                to    CD49 003D CA2D 40CD C901 C3A5 5000 0000
```

This zap tidies up the directory display when in the expanded (,A) mode.

```
SYS8/SYS,04,C8 (I)  from  2E2E 2E2E   to  2020 2020
SYS8/SYS,04,F3 (I)  from  2E 2E2E 2E  to  20 2020 20
```

This is a zap to SYS5/SYS (DEBUG) to set the high value of the displayable characters including graphics (I find it's best to stay under BFH).

```
SYS5/SYS,03,69 (I)  from  C6 3F 4A   to   00 0E xx
```

(replacing "xx" with the Hex value of the highest displayable character.)

I think I (we) may have opened a can of worms here with suggested modifications to the ROM [see the article "MODEL I LEVEL II LOWERCASE ON POWERUP" in Northern Bytes Volume 6, Number 8]. I'm sure every reader who has a Model I will have suggestions or ideas to change ROM before blasting an EPROM..... On my part, I plan on blasting both my ROM's (I have the two chip set), so I can cure the problem of TABbing past 63. To do this, I simply changed the byte at 213BH from a 3FH to a 7FH for a TAB maximum of 127 (or to what ever you wish if you want even greater TABbing positions available). I also changed the 43H at 06FA to 42H to correct the printer paper line count for my printer. Furthermore I inserted all the ROM changes that you documented in Appendix III of your 'ROM Routines' book.

I'm also trying to find where BASIC stores the '?' token so we don't get a syntax [error caused by an invalid BASIC token when] typing 'L?' in a Basic line rather than 'LPRINT'. You must have lots of other suggestions or ideas too. Who knows, we may end up with a completely rewritten ROM set and I'm sure that a whole issue of Northern Bytes can be devoted to this. Who says the Model I is dead???

......I feel that one could use a 27128 IC instead of the 2764 and, by bank switching, have two ROM sets available. The possibilities seem endless when you start thinking of them.

Bob Seaborn

[Sorry about hacking up your letter, Bob, but at the last minute I inserted part of the text of your letter into my article in the last issue (as I'm sure you noticed). On reviewing it, I found that I omitted the NEWDOS/80 zaps and some other pertinent comments, so I decided to print most of the rest of your letter this issue.

One comment you made needs to be amplified. You indicated that the byte at 213BH could be changed to "whatever you wish" if you need to TAB past 127. Not quite true. The only other option that will work properly is 0FFH, which will allow TABbing to 255 (which I recommend). The reason is that this value is ANDed against the actual TAB argument used in a BASIC program. If you change this byte to any other value from 128 to 254, some of the bits of the argument will be zeroed and thus certain TAB arguments will not work properly (they will be changed to something else).

I'm open to other suggestions for ROM modifications, particularly those that fix bugs in the ROM. We can present them here, and leave it to the individual user as to whether to actually implement them or not. That goes for mods to the Model III/4 ROM (or 4P ROM image) as well. Bear in mind, though, that major changes to the ROM code may cause problems with software that makes ROM calls.

Readers, also see the note in THE EXTERMINATOR column about Bob's offer to burn EPROMs.]

Dear Jack,

... While you were "zapping" around the Allwrite files you may have noticed that the ALK/CMD keyboard driver can be modified to change the layout of the keyboard or can cause any ASCII character to be printed with a redefined key. We have a "DVORAK" version of ALK if anyone ever wants it. Just change the characters in ALK any way you want them. You might find some use for this information.

Ron Maio
PROSOFT, Box 560, North Hollywood, California 91603

[Thanks, Ron. In checking my copy of ALK/CMD, I found this table starting at File Relative Sector 2, Byte A4 (depending on the version of Allwrite that you have, it may start in a slightly different place). It's easy to recognize, it starts with the characters @`aAbBcCdDeE ... and so on. Simply find the character you want to replace and zap the new character in. Be sure that you do this only on a backup copy of ALK/CMD, in case you zap the wrong byte and mess up everything! By the way, even though Ron tipped me off to this, such zaps to ALK/CMD should probably still be considered unofficial and unsupported by PROSOFT. In other words, zap at your own risk!]

DON'T THROW YOUR DAMAGED PRINT HEAD AWAY

If the print head of your dot-matrix printer goes bad, you might be tempted to toss it out and buy a new one. If you do that, you might wind up paying more than you have to.

For example, I recently saw an ad that read "PRINT HEADS - Rebuilt-Repaired-Exchanged-Bought-Sold. Obviously, a rebuilt or repaired print head should cost less than a brand new one. Since some print heads can cost nearly as much as a new printer, it might be worthwhile to investigate this service, should the need arise.

The ad that I saw was from Granada Data Systems, 2323 Ind. Parkway West, Hayward, California 94545-5005, telephone (415) 786-9007. However, I have never used the services of that firm, so this mention does not constitute an endorsement. Should you know of other firms that perform similar services, let us know and we'll pass the information along. By the way, these firms generally specialize in repairing certain makes of print heads, so before you send in a print head, call or write first to make sure that they can service your particular model.

PUBLIC PACKET RADIO SERVICE PETITION
SIX METER AMATEUR BAND SUGGESTED
by Frederick O. Maia, W5YI

[This article is reprinted from The W5YI Report, a newsletter edited by Fred Maia that features "Up to the minute news from the worlds of amateur radio, personal computing, and emerging electronics." The W5YI Report also distributes amateur dio license preparation materials that contain all questions, swers, and a discussion as to why each answer is right. For more information about The W5YI Report, write to P.O. Box #10101, Dallas, Texas 75207 or telephone (817) 461-6443.

Please note that this article should be read by all personal computer users that have an interest in computer-to-computer communications, not just by those readers presently involved in amateur radio!]

The FCC has accepted a proposal for public comment by Donald L. Stoner, W6TNS, for the creation of "the Public Digital Radio Service." The acceptance by the FCC is in keeping with a Commission decision reached December 14, 1983, at the Commission's "no code" proceeding - Docket #20282. The PDRS petition was assigned RM-5241 - comments close on January 6, 1986.

The FCC said at the 1983 open Commissioner's meeting that Stoner's proposal to establish a Computer Hobbyist Radio Service would be accepted as comments on the codeless class of Amateur Radio License, but that they would "entertain future proposals for allocating spectrum separate and apart from amateur radio quencies for a new Computer Hobbyist Radio Service." It was d still is) Stoner's contention that the Computer Hobbyist Radio Service would have brought in literally millions of people into amateur radio.

At the Commission meeting, Bob Foosaner, Chief of the FCCs Private Radio Bureau said "...it is a petition that we will look at." He indicated that the question had been examined several times in the past related to CB radio. "The Hobby Class may be something that replaces the CB type of thing," he said. "We are going to look at it. It's going to be difficult to find spectrum for this type of service." The radio spectrum is not allocated for such a service.

That was two years ago. On December 6th, 1985, the FCC released a Public Notice stating that they were now going to consider a public digital network. Stoner's extremely well done pposal ran to 28 typewritten pages. I called Don when I ceived the FCC notice. He had not yet heard that his proposal had been assigned an RM file number.

Don is uniquely qualified to put forth the computer-to-computer proposal. He is a well known technical author and educator and was a CQ magazine editor at one point. He has written hundreds of articles and several books on the subject of amateur radio and computer communications.

Don has been a licensed radio amateur for thirty years plus and is generally credited with the concept which grew to become the OSCAR satellite series. His ideas are usually far ahead of heir time.

Stoner is the Vice President of Engineering at the croperipheral Corporation in Redmond, Washington and currently heads up an effort that "sofcasts" personal computer programs via FM subcarriers relayed to commercial FM stations by satellite. He designed a $70 "black box" device (which he calls a shuttle communicator) that links airborne software with a radio and personal computer.

SUMMARY OF THE PDRS PROPOSAL

Stoner suggests a wide band channel (non-channelized) to send data at high rates of speed. "A single wideband channel can be thought of as a digital highway with addressed packets entering and leaving the route in a highly organized manner. [This] can only be accommodated within the VHF band or higher frequencies."

Stoner said that the 52-54 MHz frequency range is virtually unoccupied and therefore unused. "It is estimated that out of the 400,000 radio amateurs in the United States, less than 1,000 are active on the six meter band. Due to a potential for interference with adjacent television channel 2 (54-60 MHz), virtually all six meter users operate between 50 and 52 MHz. For all practical purposes the radio spectrum between 52 and 54 MHz is wasted," he noted.

PDRS POTENTIAL FOR TV INTERFERENCE

The petitioner maintained that no TVI can occur from a radio modem operating in the 52-54 MHz band if:

(1.) ...the TV station signal received strength exceeds 100 µv
(2.) ...the effective radiated power of the radio modem does not exceed one watt
(3.) ...the radio modem antenna is vertically polarized with respect to a horizontal TV receiving antenna and
(4.) ...all modulation and spurious products falling outside the authorized bandwidth conform to specified FCC rules

WHAT IS A PACKET RADIO NETWORK?

Stoner explains in his petition that a local area network (LAN) is a system of a limited number of computers connected together by cable in a manner which permits intercommunication.

"A packet network is an infinite number of LAN's connected together by radio waves. A packet radio network may be thought of as the digital equivalent of the U.S. Postal Service. The information to be sent to another computer is equivalent to a letter. The letter is placed in an envelope which includes a destination address. This is called the packet. This packet is sent along with those of other users into the network, which acts like a mailbag. The packet address also includes something like a postal zip code. Each radio modem (node controller) connected to the network is a destination mailbox."

"Each user is considered to be a 'node' in the Public Digital Radio Service. These stations or nodes constantly monitor the transmissions (mailbags), looking for packets (envelopes) which are addressed to them. If a message to the node is detected, it is held in memory (the recipients mailbox.)"

"The message may be intended for a nearby node as indicated by the 'zip code'. In this case, the node 'mailbox' does not activate. Rather the node becomes a 'post office' and passes (digipeats) the message to one or more other nodes. The reply from the destination computer is handled in the same manner but the direction is reversed."

"Packet radio transmission (movement of 'mailbags') can occur at very high speed. In fact, the speed is only limited by the bandwidth of frequencies allotted to the network."

The Stoner petition said that the 2 MHz requested "is sufficient to handle data rates in excess of 1 million bits per second" and could handle a large number of simultaneous users. "Actually, the number of simultaneous users is unlimited due to an advantage of FM radio called 'capture effect.' Each node will 'hear' (or capture) only the strongest stations in the immediate area of the node. More distant stations will be inaudible and will not cause interference."

A COMPARISON WITH CB RADIO....

"There is a major reason for the chaos which developed on the 27 MHz Citizens Band. The licensees of this service did not feel it was in their interest to abide by the Rules and Regulations. We can learn from this experience by designing a service where Rule compliance is in the interest of the user:

Modulation: There must be no provision for voice communications in the PDRS.

Identification: The FCC licensing workload will not be increased. Services which are essentially self-regulating (such as the remote control of objects, garage door openers, etc.) do not require the use of call letters.

Compliance: Each radio modem has its own unique ID code, that is, its packet address.

This is both the serial number and digital address of the unit. This code also identifies the manufacturer and the physical location of the radio modem. Violations of technical requirements can be easily correlated by manufacturer. If a unit is found to be non-complying, a message can be addressed to that radio modem advising the user of the problem. The Commission personnel sending the message receives the customary delivery acknowledgment of the message. Thus there can be no question that the user received the Notice of Violation.

Power Output: A major contributing factor to the 'CB problem' was the addition of power amplifiers to CB radios in an effort to increase the talk range. "Adding a power amplifier to a radio modem will produce no increase in performance" since "the unit will 'retrain' to reduce its power output to maintain the nominal signal level at nearby radio modems."

Antenna: No advantage would be obtained by use of high gain, directional antennas since the radio modem would retrain to produce the nominal signal strength at nearby nodes. Raising the height of the antenna would cause no noticeable increase in communication range.

Off Frequency Operation: There is only one channel or band. Out of band data would be destroyed by amateur radio or TV channel 2 signals.

TECHNICAL SPECIFICATIONS....

The radio modem (terminal node controller) shall meet the following specifications:

Frequency Band: Equipment authorized to operate in the Public Digital Radio Service shall be capable of receiving and transmitting data within the band 52-53.999 MHz.

Modulation: The data shall frequency modulate the carrier in a frequency shift keyed scheme. No provision for voice modulation or detection.

Modulation and Spurious Products: The data rate (which will be left to industry to determine), waveform and signal processing shall be such that all products which all outside the authorized bandwidth be suppressed by 43 plus 10 log10 (mean output power, in watts) decibels.

Power Output: The power delivered by the final amplifier stage into a 72 ohm load shall not exceed 1.0 watts. Further, the radio modem shall have an initial powerup 'training' mode. Upon powerup, the power output will be 1 milliwatt. The power will increase during 'training' in 3 db. steps until contact is established with nearby modems (node controllers). This value is stored in memory and becomes the nominal power output for the radio modem.

Antenna: Shall consist of a vertical radiator which does not exceed one-quarter wavelength. The antenna shall exhibit no gain or directional characteristics.

Transmitter Identification: Each radio modem shall have an imbedded identification which is transmitted as part of its packet address. The address will be used to identify the manufacturer, the serial number and the routing code of the equipment.

Packet Construction: The packet and destination address will be contained in the header. The header will be constructed to limit the number of destination addresses. This is done to specifically preclude the transmission of 'junk mail.'

Remuneration: Users of the PDRS shall be specifically prohibited from receiving any form of remuneration or compensation, either in the form of funds, goods or services. The purpose of this provision is to prevent the use of the Public Digital Radio Service for the benefit of common carriers. The restriction shall not be construed to preclude the use of the PDRS for business applications. "For example, the radio modem would be extremely useful within buildings to avoid the need for local area network cabling."

Type Acceptance: Type acceptance procedures, similar to those for Citizens Band equipment, will be required to "insure that commercially manufactured equipment used in the PDRS meets the specified technical requirements for this service."

INTERNATIONAL REGULATIONS - Stoner says that since the allocation is above 50 MHz, it appears that no international treaties will be involved..." (W5YI Report Editor's Note: We take issue with that! ITU agreed upon allocations specifically allocate the 50--54 MHz band to Amateur exclusive! No exceptions. It appears any FCC consideration to PDRS will have to be within the confines of the Amateur Radio Service. As we see it, it could be done, but would have to be a no-code unlicensed amateur class.)

AMATEUR RADIO OPPOSITION - Stoner: "...there can be no defense by amateurs of the inactivity on 6 meters. A reallocation of the frequencies requested would benefit the majority at virtually no expense to the minority."

AMATEUR RADIO COLLABORATION - "The principal purpose of this petition is to obtain an allocation for a public computer communications band. The writer would not object if this goal could be achieved as part of the Radio Amateur Service. The computer public would accept an administrative fee in return for access to the radio spectrum. However, they would never accept any sort of 'testing' to achieve this goal."

CONCLUSION....

So there you have it. A capsulized version of what has the potential to become an unbelievably popular public radio service. Did you know that their are more subscribers to BYTE magazine than the entire ham population of the United States?

The public's need to communicate with one another at a low price has never been realized. PDRS could provide immediate message delivery at far less than the cost of a 22¢ postage stamp (actually no cost) and no delivery time is involved.

Currently non-amateur computer-to-computer communications requires expensive telephone interconnection. You can assured that telephone and broadcast interests will oppose Public Digital Radio Service!

Be sure to send a copy of your comments to Don Stoner, W6TNS; 6014 E. Mercer Way; Mercer Island, Washington 98040. FCC rules require this so Stoner can respond to them in his reply comments.

[NORTHERN BYTES editor's note: The statement above that "you can be assured that telephone and broadcast interests will oppose the Public Digital Radio Service" is probably the understatement of the year. The United States Postal Service and packet network operators such as Telenet, Tymnet, Uninet, etc. probably won't think much of it, either. And, according to a more recent issue of The W5YI Report, the American Radio Relay League (an organization of amateur radio operators) "came in vigorously against the Stoner proposal" (another organization that I am not familiar with, called the "Association of Maximum Service Telecasters", also filed a pleading in opposition to the proposal). Obviously, all of these organizations (with the possible exception of the ARRL) are looking out for their own financial interests - if the public has access to free communications service v radio, it could affect their bottom line.

I don't know if it has ever occurred to many people, but there is no technical reason at all that you cannot send messages or computer programs across the country (or around the world) for FREE. Amateur radio operators do it all the time. But amateur radio isn't for everyone, since you have to have certain technical knowledge plus the ability to send and receive Morse code in order to get a "ham" license. Many more people would be interested in joining the ranks of amateur radio operators if the Morse code requirement were dropped, yet this is one thing that the present amateur community seems very opposed to (the only reason that I can see for this opposition is that many "hams" feel that if they had to learn the code, everyone else that wants to participate in the hobby should have to learn it too - yet this self-righteous attitude is the very thing that is killin support for amateur radio!) The FCC has a lot of input fr amateur radio operators (specifically the Amateur Radio Relay League and similar organizations) on this subject, but hears very little from those who would become "hams" if the code requirement were abolished. Thus an "elite" group (which, in some areas of the country, seems to be composed of a very snobbish group of individuals) is able to dictate what the entrance requirements to amateur radio will be for the rest of us.

Yet it's true that many people that do not wish to become amateur radio operators (at least not under the present terms) still would like to have some access to the airwaves for personal communications purposes. This was the reason that Citizens Ba radio was so popular for a time. Unfortunately, the immatu actions of some CB users, coupled with the technical limitations of the CB radio service ("skip", signal fading, etc.), made communications via CB so unreliable that most serious users soon left the band in disgust. The PDRS proposal would appear to be one way to allow more people to communicate via the airwaves, while still having safeguards to prevent certain users from turning the system into another electronic pigsty.

I won't go into the details here, but a couple of years ago a proposal was put forth that would have permitted everyone to have a portable telephone (similar to today's cordless phones, but using higher power for greater range) that would have been usable up to a few MILES from the user's home or office (their was even a provision for repeater systems that would have extended the range even further). This would have made a telephone in the family automobile affordable for everyone. Of course, the telephone companies and cellular telephone interests came down hard and heavy with opposition, and effectively managed to kill the proposal.

Now, my question is, why should the interests of the telephone companies, packet networks, broadcast interests, and/or postal service be protected to the detriment of personal computer users and/or the general public? Why should you or I have to pay to send data (a computer program or text file, perhaps) across the country, if the technology is present t allow us to do it for free? Why are we being forced to subsidize these institutions every time we wish to communicate? (The lawyers among us might even want to consider whether this is an infringement on our freedom of speech. I sure feel that is is; I know that the costs involved will make me think twice about sending an electronic message to a distant point. Sometimes I'll send it anyway, but many times I won't. Regular mail service through the post office is comparatively inexpensive, but does not have the immediacy of electronic communications, and thus tends to inhibit the flow of conversation.)

It is my hope that the Federal Communications Commission will recognize that there are many people that would like to have access to a non-commercial personal communications service, yet at present are effectively being denied free and efficient access to the airwaves. In a free country such as the United States, the ability to communicate freely should not be restricted to those who can afford to pay for the privilege. The commercial communications carriers will always have a place in our society serving the business user (which seems to be the type of traffic that they prefer to carry anyway), so why not give the hobbyists and "just plain folks" a service that they can use freely? If the FCC acts favorably on this proposal, it will open up a new world of communications for personal computer users. Let's hope that they make the right decision on this one!

If you'd like to write to the FCC and express your feelings on this subject, you should do so as soon as possible. The address is: Federal Communications Commission, 1919 M. Street N.W., Washington, D.C. 20554. Don't forget to send a copy of your comments to Don Stoner, as mentioned in the article.]

NULL INPUT

by Neil Yerkey

[Reprinted from the Western New York TRS-80 User's Group newsletter.]

One change from Model III to Model 4 BASIC is most unwelcome. On the Model III, a null input (ENTER key only) will leave the previous value of the input variable unchanged. Null inputs can then be used to accept default values. On the Model 4, a null input will wipe out the previous value of a variable. In the following example, hitting the carriage return in response to the prompt will print a 6 in Model III, and a 0 in Model 4:

```
10 A = 6
20 INPUT "TYPE A VALUE OR <ENTER> TO ACCEPT AS IS"; A
30 PRINT A
```

One way to accept default values is to add the following lines. This tests to see if the input variable is 0 and, if so, changes it back to the original value:

```
15 B = A
25 IF A = 0 THEN A = B
```

One problem: What if you WANTED to change the 6 to a 0? The above lines would always change an inputted 0 back to a 6! There would be no way to enter a value of 0. Does anyone know how to accept a default value and also allow a person to enter a [Yes, see below -editor].

For string variables, null inputs return a null string [in Model 4 BASIC]. That is not so much of a problem because you can test to see if the length of the variable = 0:

```
10 A$ = "SMITH": B$ = A$
20 INPUT "TYPE LAST NAME OR <ENTER> TO ACCEPT AS IS";A$
30 IF LEN(A$) = 0 THEN A$ = B$
40 PRINT A$
```

[NORTHERN BYTES editor's note: This gives us the solution to the question of how to accept a default value, yet still permit the user to enter a zero value. We simply input the reply into a string variable, test for a null string, and if the string is not null, then assign it to the appropriate numeric variable:

```
10 A = 6
20 INPUT "TYPE A VALUE OR <ENTER> TO ACCEPT AS IS"; B$
25 IF B$>"" THEN A = VAL(B$)
30 PRINT A
```

This isn't exactly the same as using a numeric variable for input (it won't prompt the user to redo the input if invalid characters are entered), but it's close enough for most applications.]

MORE ON THE PBUFF PRINTER BUFFER KIT

Don McKenzie's PBUFF parallel printer buffer kit has been mentioned at least a couple of times in recent issues of Northern Bytes (the first mention was in Volume 6, Number 4, page 12). Well, there have been a couple of new developments regarding that kit. Serial printer users, hang in there, we'll get to you in a minute.

First of all, although the original version (the 8K-64K version, now known as version 2.0) is still available, a new version 3.0 is now also available. Version 3.0 is a 256K unit that uses eight 41256 (256K) RAM chips. The price is the same ($35.00 Australian plus $5 for postage to North America. Don can also supply the required 3.58 MHz crystal for $2.90 Australian if ordered at the same time). Of course, the 256K chips (which are NOT included in the kit) are more costly than the 8K-64K components, so the TOTAL cost for the 256K kit would be higher. This "PBUFF SHORT FORM KIT" consists of a printed circuit board and an EPROM programmed with the PBUFF software (you supply all other parts), plus full assembly instructions (including a hardware debugging section) and free hardware debugging advice via telephone (that is to say, Don won't charge you for giving the advice. If you live in the U.S., AT&T will charge you $2.17 for the first minute and 81¢ for each additional minute during the "economy" rate period. Carriers other than AT&T may charge less). The kit price may seem high until you remember that the current value of the Australian dollar is quite a bit below that of the U.S. dollar (check with your bank for the current exchange rate), which makes the cost much more reasonable.

The original PBUFF was designed only for parallel (Centronics) I/O. However, due to demand, Don has now released a serial board add-on for use with PBUFF, which performs serial to serial, serial to parallel, and parallel to serial conversions, all at standard RS232C levels (or TTL in). Serial baud rates (in and out) can be set at 75, 300, 600, 1200, 2400, 4800, 9600, and 19200. You can select a word length of from 5 to 8 data bits. Odd, Even, or No parity can be used, and 1 or 2 stop bits can be selected (or 1.5 stop bits at 5 data bits). These settings insure that just about any computer system can be connected to almost any printer.

The serial board can be used with any revision PBUFF board. It's a bare printed circuit board (you provide all components) and sells for $18.00 Australian (don't forget the $5 charge for shipments to North America).

For further information on any of Don McKenzie's hardware projects, write to him at 29 Ellesmere Crescent, Tullamarine, Victoria 3043, AUSTRALIA, or phone him at (03) 338 6286 (from the U.S. dial 011+61+3+338-6286). I will again mention that Don is a hobbyist, not a large money-making corporation, so when you write include at least two or three dollars for return postage (Don has a small "catalog" of hardware mods, many of which are specifically for the TRS-80 and "clones", that he will send to you if you supply sufficient return postage).

MODEL 4 AND 4/4P TECHNICAL REFERENCE MANUAL ERRORS

Page 17 of the Model 4 Technical Reference Manual (26-2110) and Hardware page 5 of the Model 4/4P Technical Reference Manual (26-2119) have the four high res ports reversed. Below are the correct port locations.

PORT: 80 hex
READ: reserved
WRITE: Graphics X Register write

PORT: 81 hex
READ: reserved
WRITE: Graphics Y Register write

PORT: 82 hex
READ: Graphics Ram Read
WRITE: Graphics Ram Write

PORT: 83 hex
READ: reserved
WRITE: Graphics Options Register

This information was posted on the Tandy Forum on Compuserve.

ARRANGER 2.1 ON THE MODEL 4P

by Gary Bryce (phone 011+61+2+628-5058)

[This article is reprinted from the SYDTRUG NEWS, P.O. Box 297, Padstow, New South Wales 2211, AUSTRALIA.]

The "ARRANGER" by Dan Foy is in my opinion one of the best disk indexing programs available for the TRS-80. It will "BOOT" and operate in unmodified form on the the Model I, III and 4, but it will not "BOOT" on the 4P.

Why? The MODEL III ROM must be available prior to booting, as the ARRANGER operates in Model III mode on a Model 4 or 4P, and this means that the MODELA/III file must be operational on a 4P before the ARRANGER can be loaded. You may, if it doesn't eventually give you a pain in the neck, use the ModelA/III file disk followed by the ARRANGER, or execute the procedure detailed below to copy the ROM image to a backup of your ARRANGER master disk. The resulting disk can still be used on a Model I, III or 4 without any problems.

1. Build the following patch file on drive 0.

```
.ADIR/FIX
.Patch to "LOCK OUT" all tracks below track 35
.
D00,00=FF FF FF FF FF FF FF FF
D00,08=FF FF FF FF FF FF FF FF
D00,10=FF FF FF FF FF FF FF FF
D00,18=FF FF FF FF FF FF FF FF
D00,20=FF FF FF
D00,60=FF FF FF FF FF FF FF FF
D00,68=FF FF FF FF FF FF FF FF
D00,70=FF FF FF FF FF FF FF FF
D00,78=FF FF FF FF FF FF FF FF
D00,80=FF FF FF
.End of Patch
```

2. Mount an unformatted disk on drive 1 then create and execute the following JCL.

```
.ARRANG4P/JCL
.
.Format a disk in drive 1 with the Directory on Trk 35
FORMAT :1 (NAME="ARRANGER",DIR=35,Q=N,ABS)
.
.Lockout Tracks below Track 35
PATCH DIR/SYS.LSIDOS:1 ADIR/FIX (O=N)
.
.Correct the Directory Data Address Mark
REPAIR :1
.
.Copy the MODELA/III file from drive 0 to drive 1
COPY MODELA/III:0 :1
//END
```

3. Mount the <u>original</u> MODELA/III File disk (not the disk created in step 2) in drive 0 and press RESET.

4. After the Model III ROM image has loaded, replace the MODELA/III with your Master ARRANGER disk and again press RESET.

5. Follow the normal initialization procedure and select the BACKUP ARRANGER option from the menu to create a copy of the ARRANGER on drive 1.

6. Replace the ARRANGER disk on drive 0 with the TRSDOS 6.x disk and press RESET.

7. With the new ARRANGER disk in drive 1, from TRSDOS READY type in the following exactly as it appears, noting that <BREAK>, <ENTER> and <SPACE> mean to hit those respective keys.

```
DEBUG <ENTER>
<BREAK>
1,0,1,R,8000,1 <ENTER>
H8000 <SPACE>
00 <SPACE>
<SPACE>
23 <ENTER>
1,0,1,W,8000,1 <ENTER>
O <ENTER>      (the letter O, not the number 0)
DEBUG (OFF) <ENTER>
```

You can now take the disk from drive 1 and put it into drive 0 and press RESET. The Model III ROM image will load followed by the ARRANGER.

Please note any backups of the newly created disk using the Backup option of the ARRANGER will not include all the additions required (ie: Directory and MODELA/III file) and therefore the above procedure must be repeated for all subsequent copies which are to be used on the 4P. However, only steps 1 to 5 need be performed <u>if</u> you use the modified ARRANGER in step 4, as the patch to the boot sector will be duplicated by the BACKUP routine in the ARRANGER.

This procedure should work for many "Self Booting" Model III programs that you want to operate on the 4P. Some possible conflicts could arise with programs that use all forty tracks, but these should be rare (and if you are able to use an eighty track drive as drive 0, the directory could be placed above track forty). As these "Self Booting" programs must have a Boot Sector readable by the normal ROM routines, accessing sector 1 for modification should not be a problem.

A FEW TIPS ABOUT PASSWORDS AND THEIR USES

by Warick Sands

[Reprinted from the TRS-80 SYSTEM 80 Computer Group newsletter (16 Laver Street, MacGregor, Queensland 4109, Australia).]

Most users of Disk Systems view passwords with mixed passions. If you are limited to a TRSDOS type system, they are a nuisance since it becomes next to impossible to do anything with the file unless you know the password. Hoorah for NEWDOS/80. No more worries about passwords. Anything is copyable. We set the SYSTEM option AA=N and forget evermore about passwords.

At least that is what I did. Until the other day that is. I am in the middle of writing a Word Processor. I have the 80k or so of source code in Drive 1, while drive 0 has a copy of Lazywriter and an editor/assembler. Having finished the printing portion of the package, I began work on the editing half. Brought the file in which is called EDIT/ASM, assembled it with a filespec of EDIT/CMD. Unfortunately, Lazywriter has a file called EDIT/CMD. Instant destruction of 40% of Lazywriter. Naturally I had a backup. But what if I hadn't?

How can one prevent this sort of accident from happening? Passwords to the rescue. Go to DOS ready type SYSTEM 0 AA=Y. Instant protection. But problems started to arise. I wanted to change the PDRIVE specifications. DISK ACCESS DENIED was the response. I wanted to change the SYSTEM options. DISK ACCESS DENIED. The simplest answer is to use a different DOS disk. But I find that is a nuisance. So when all else fails read the manual!! Here is what I found.

I didn't really want to password protect the disk. All I needed was to password protect the files of interest. Use the ATTRIB command.

```
ATTRIB EDIT/CMD,PROT=READ,UPD=PASSWORD
```

protects the file. I can use the file as usual but it can't be overwritten, renamed or killed without specifying the password. That isn't likely to happen accidentally. Use a password easy to remember.

SYSTEM option SYSTEM AA=Y is still required. I still couldn't change the PDRIVE specs without specifying the password. When you're changing the PDRIVE's for every other disk it becomes somewhat tiresome (and time consuming) to type in

```
PDRIVE PASSWORD:0,1=5,A
```

Back to the manual! Set the DISK PASSWORD to null by

```
PROT,PASSWORD:0,PW=
```

That cured the trouble. I could change PDRIVE parameters and SYSTEM options to my hearts content, and the file was still protected.

A week later I wanted to do a disk copy. DISK ACCESS DENIED. Back to the manual!! Set SYSTEM AR=Y. This allows the full disk copy to occur without the checking of passwords. Problem solved. So if you want to protect files on a disk from accidental erasure, it can be done easily.

MORE MATHEMAGICAL RECREATIONS from MAGIC MATH PLUS

by Dr. Michael Ecker
and Recreational Mathemagical Software

Hello everybody! Sorry to have missed my contributions to Northern Bytes, but I've been busy. Anyway, I've got a few teasers for programs over the next couple of issues. They are drawn from Magic Math Plus, a collection of about 40 programs for TRS-80 Model III disk (also for 4 or 4P in Model III mode). That disk is a self-booting disk, and all programs are in menu-generated format. There is also a smaller MSDOS version, as well as a specialized Sanyo 550/555 version. I sell the software to schools licensed for $87.50, or to individuals for $37.50 (TRSDOS version) / $27.50 (smaller MSDOS version). However, Northern Bytes readers who so identify themselves may have the TRSDOS version for $29.75 or the MSDOS version for $22. You can also get a subset of the programs, including six of the best programs (either version) for $10.95.

After the programs and text which follow, I'll tell you how to get the programs for even less than that, as well as details of a brand new newsletter created to make up for a bit of the void created by the death of such publications as Creative Computing and Popular Computing. As some of you may know, I was the "Recreational Computing" columnist for each of these, as well as the founder of "Mathematical Recreations" in Byte.

Okay, enough commercials. Let's get on with it!

Digital Delight

Here is a cute program involving digits of an integer. For all the abuse that Basic gets (I won't debate the issue here), it is quite good at string manipulation. Some of the best ecreations involve digits of a number. I don't claim that these activities will make you a better person, but I think you'll share my enjoyment of them. I won't offer an explanation of why the tricks work, although the full program as it appears on Magic Math Plus does contain a complete explanation. To save you typing - and perhaps to arouse your curiosity and make you think - I won't include that explanation which comes as part of the program ordinarily.

In Digit, the computer asks you to: 1) pick a number; 2) add up its digits (thus the title); 3) subtract the sum of the digits from the original number; and 4) pick a digit from this, and cross it off from that last answer. The computer then tells you the digit which you were thinking of (i.e., the one you cross off).

To illustrate, suppose that we take any integer. Let's take 1234. Add up the digits. They add to 10. Now subtract the sum f the digits (10) from the original number (1234) to get 1224. entally select any digit of this bunch. Let's suppose that we pick the 4. The computer program, Digit, will ask you to indicate the number left after you delete the digit you're thinking of. In this case, you would knock out the '4' and so you would indicate that the number left is 122. When prompted, type in 122 and hit the Enter key.

From that, the oh mighty and psychic computer will somehow mystically divine the digit you have picked!

Here is the program listing. I realize that some of you will recognize this as just a computer implementation of "casting out ines", but I find that even those who have heard of this process n't really understand why it works. I will defer explanation at this time, although it is on Magic Math Plus.

```
99 REM TRS-80 VERSION SHOWN HERE
100 CLEAR500:S$=STRING$(18,42)
110 CLS:PRINT@342.S$:PRINT@406,"*DIGIT
PREDICTION*";:PRINT@470.S$;:PRINT@960,"RECREATIONAL MATHEMAGICAL
SOFTWARE    (C) 1985 DR. MICHAEL ECKER"
120 FORJ=1TO1500:NEXTJ
220 CLS
230 PRINT:PRINT TAB(17)"THINK OF ANY WHOLE NUMBER.":PRINT
240 FOR Z=1 TO 1000:NEXT
250 PRINTTAB(13) "ADD UP THE DIGITS OF YOUR NUMBER."
260 PRINT:FOR Z=1 TO 2000:NEXT
270 PRINT TAB(9)"NOW SUBTRACT THE RESULT OF THE
ADDITION":PRINTTAB(16)" FROM THE ORIGINAL NUMBER."
280 PRINT:PRINT:PRINT
290 PRINTTAB(14) "PRESS <ENTER> TO CONTINUE";:INPUT XS
300 CLS
310 PRINT "THINK OF ANY DIGIT IN THE LAST ANSWER (but not
zero)."
320 FOR Z=1 TO 800:NEXT
330 PRINT
340 INPUT "WHAT IS THE NUMBER LEFT IF YOU DELETE THAT DIGIT";N$
350 S=0
360 FOR I=1 TO LEN(N$)
370 S=S+VAL(MID$(N$,I,1))
380 NEXT I
390 IF S>=9 THEN S=S-9:GOTO 390
400 A=9-S
410 FOR Z=1 TO 700:NEXT
420 CLS
430 PRINT "CONCENTRATE NOW ON THE MISSING DIGIT..."
440 FOR Z=1 TO 1000:NEXT
450 PRINT:PRINT "AHA!... THE MISSING DIGIT IS";A;"!"
460 PRINT:PRINT:PRINT:PRINT:FOR Z=1 TO 1000:NEXT
470 INPUT "PLAY AGAIN (Y OR N)"; RE$
480 IF RE$="Y" OR RE$="y" THEN RUN ELSE END
```

Kaprekar's Constant - The Remarkable Narcissistic Number 153

I'm going to be brief on this digit-delving program and problem, although there is a lot I can say about it. This problem or question deals with a phenomenon of attraction which I call "black hole attraction". There are two elements to this, and I'll describe these in a moment. First, let me give you the idea.

The key idea is you do some process to a number to produce a new number, and then you ask whether the answers obtained by iteration (i.e., repetition of the process) must eventually hit some number. An even more key question is whether there is any number which has the property that, when you do the process, you get the same number you started with.

The two ingredients needed for a number to be a "black hole" (my terminology), with respect to a given process, are therefore:

1) The new number obtained in the next step is the same number.

2) If you start with any number at all, within a finite number of iterations (steps), you must eventually hit the special number found in 1) - at which point, continued iteration just keeps producing the same number.

There are actually many instances of this, and I can explain a popular card trick with respect to this, but let me get back to Kaprekar and his constant.

Look at the number 153. Suppose that you take the cube of each of the digits. That is, take 1x1x1, 5x5x5, and 3x3x3. These are 1, 125, and 27. Add them up. What do you get? Why, good old 153 again!

Lest you think this isn't special, try it with another number and this will probably fail. If you agree to restrict yourself to positive whole numbers which are greater than 1, the only other numbers with this narcissistic property (as we call it in recreational mathematics) of the number equalling the sum of the cubes of the digits, are 370, 371 and 407.

Of the four solutions, only 153 is a multiple of 3. Here is now what I claim in order to have the "attractor" part, or second part, of my "black hole" definition. I claim that no matter which integer multiple of 3 you take, if you take the sum of the cubes of the digits, either that answer will be 153, or, more likely, you'll have to repeat taking the sum of the cubes of the digits, and so on, but eventually you'll get 153 (at which point further iteration keeps producing 153). It is not that obvious why this is so, so don't berate yourself if you can't see why it has to happen.

The following program, Trick153, (again, from Magic Math Plus), tests this. The program contains sufficient error-checking to disallow impermissible inputs, most notably numbers which are not multiples of 3. Hence, don't worry about inputing only multiples of 3; the computer will warn you and stop you if you goof up.

Why not play around with the program a bit? It's short enough not to be too tedious. Enjoy it! I'll rejoin you after the program for some closing words from our sponsor (which is Recreational Mathemagical Software in this case).

```
99 REM TRS-80 VERSION HERE
100 CLEAR500:S$=STRING$(27,42)
110 CLS:PRINT@338,S$:PRINT@402,"*THE REMARKABLE NUMBER
153*";:PRINT@466,S$;:PRINT@960,"RECREATIONAL MATHEMAGICAL
SOFTWARE    (C) 1985 DR. MICHAEL ECKER"
120 FORJ=1TO1500:NEXTJ
200 DIM A(15)
205 DEFDBLN
```

```
210 CLS
220 PRINT "IN THIS TRICK, I WILL ASK YOU TO GIVE ME A WHOLE
NUMBER"
230 PRINT "WHICH IS A MULTIPLE OF 3. THE COMPUTER WILL THEN
TAKE EACH"
240 PRINT "OF THE DIGITS AND CALCULATE THE CUBE OF EACH DIGIT."
250 PRINT:PRINT "THEN THE SUM OF THESE WILL BE DISPLAYED. THE
PROCESS WILL"
260 PRINT "THEN BE REPEATED WITH THAT NUMBER, AS NEEDED, UNTIL"
270 PRINT "WE START GETTING THE SAME NUMBER. AMAZINGLY, THIS"
280 PRINT "MUST ALWAYS HAPPEN, AND ONE ALWAYS GETS THE SAME
ANSWER!"
285 PRINT: PRINT "FOR THAT ANSWER. NOTE THE IMPLIED RESULT THAT
THE SUM OF THE"
286 PRINT "CUBES OF ITS DIGITS IS EQUAL TO THE NUMBER ITSELF.
SUCH A": PRINT "NUMBER IS SAID TO BE NARCISSISTIC (of order
3).":PRINT
290 PRINT:INPUT "PRESS <ENTER> TO CONTINUE";XX
300 CLS:INPUT "GIVE ME A WHOLE NUMBER WHICH IS A MULTIPLE OF
3";N
310 PRINT
320 N$=STR$(N):L=LEN(N$)
330 IF ABS(N-INT(N))>.00001 THEN PRINT "A WHOLE NUMBER
PLEASE!":FOR Z=1 TO 700:NEXT:GOTO 300
340 IF ABS(N/3-INT(N/3))>.01 THEN PRINT "A MULTIPLE OF 3
PLEASE!":FOR Z=1 TO 700:NEXT:GOTO 300
350 S=0
360 FOR I=1 TO L
370 A(I)=(VAL(MID$(N$,I,1))):A(I)=A(I)*A(I)*A(I)
380 S=S+A(I)
390 NEXT I
400 PRINT "THE SUM OF THE CUBES OF THE DIGITS OF LAST NUMBER
IS";S
410 IF S=N THEN PRINT:PRINT "PROCESS IS COMPLETE.":PRINT"TO
START AGAIN, HIT <ENTER>."
415 IF S=N THEN INPUT X$: GOTO 300
420 IF S<>N THEN N=S:S=0:GOTO 320
```

Do you enjoy this sort of recreation? If so, then you are not only a prime candidate for Magic Math Plus, but also for a new newsletter which I'm pleased to announce here. It is REC, which is an acronym for Recreational and Educational Computing Newsletter, and it is being made available now. I am the editor and publisher of REC, with subscriptions just $16.95 for one year of six issues loaded with goodies such as what I've offered here. However, REC will have much, much more. I will be glad to send any doubting Thomases more information if I receive a standard sized, self-addressed stamped envelope with 39¢ postage affixed. For those who are sold, please send $16.95 U.S. funds made payable to Recreational and Educational Computing Newsletter:

Dr. Michael Ecker, Editor of REC
Recreational and Educational Computing
129 Carol Drive
Clarks Summit, Pennsylvania 18411

When you write, please tell me a bit about yourself (or as much as you want), especially computers owned or used, interests, background, suggestions, etc.

I will try not to make REC machine-specific, although I personally own many Radio Shack and Tandy computers, so Northern Bytes readers may feel just a wee bit more at home than some others at first. (However, I will be translating to other machines, so don't despair if you have an Apple, for instance.)

Remember I said there would be a bonus if you read this far? Well, here it is, although I'm being crazy, according to all the rules of pricing. (I figure I'm earning maybe minimum wage!!!) If you subscribe to REC ($16.95), you may purchase the sampler disk for just $2, or the full MM+ disk for just $16 (TRSDOS) or $13 (MSDOS). Please specify your computer carefully. Note that the sampler disk contains full programs, not demos, and these have the same menu system, etc. These are copyrighted programs, not public domain throwaways!

In other words, you get the newsletter and Magic Math Plus for about the same or quite a few dollars less than the cost of Magic Math Plus alone!

Please take me up on my offer before I come to my senses. Believe it or not, on top of all this, I still intend to answer every letter (provided a SASE is enclosed), just as I always did while writing for Byte, Popular, Creative, etc. There are so many goodies planned that I just can't tell you everything.

Lastly, thanks to Jack Decker and also to Charley Butler of the Alternate Source for their cooperation in allowing ourselves to work together for the computer community. At a time when so many are bailing out because they're not making their fortur it's nice to be part of an enterprise that offers something o than a chance for itself to get rich.

As always, I solicit your questions, improvements, suggestions and so on. Remember that if you'd like a reply for anything other than an order, you need to enclose a SASE, including information about Magic Math Plus or the Recreational and Educational Computing Newsletter. Write me at the address above. Until next issue, happy computing!

[Editor's note for the benefit of our readers outside the U.S.: Unless you have a supply of U.S. postage stamps, you won't be able to send a SASE. So, enclose U.S. coin or currency in the following amounts: Canada and Mexico: 1 oz. letter, 22¢; 2 oz. letter, 40¢. Europe, Asia, Australia, etc.: ½ oz. airmail letter, 44¢. For each additional ½ oz. add 44¢. Thus, if you want more information on the REC newsletter (a 2 oz. mailing), you should enclose at least $1.76 in U.S. funds.]

Dr. Michael W. Ecker, besides being the president of Recreational Mathemagical Software and the editor / publisher of Recreational and Educational Computing Newsletter, is an associate professor of mathematics and computer scie presently at the University of Scranton. He is a known compu columnist and software reviewer, and his reviews and articles are featured monthly in the Computer Shopper.

PATCH FILE FOR DOUBLEDUTY

by Dave Bower [Compuserve 70635,330]

```
.
.                 DoubleDuty Patch
.                  by  Dave Bower
.                     1/19/86
.
. This patch is for DoubleDuty version 02.05.00, by
.   Software Concepts and sold by Radio Shack for the
.   TRS-80 Model 4
.
. It makes four changes:
.    1)  Changes the DDuty cursor (because the DDuty
.         cursor overrides any cursor you may have SYSGEN'd)
.    2)  Changes the prompt in the library area (Partition
.         3) to from TRSDOS Ready to Library Area
.    3)  Clears the screen when DDuty is called up
.    4)  Changes "DoubleDuty is now operational." to
.         "All systems are now a go!"
.
. Apply as many, or as few, as you like, substituting your
.   own cursor, prompt, and message.
.
. Apply this patch using the command:
.
. ==>    PATCH DDUTY/CMD USING DDUTY/PCH
.
. Change the cursor -- 5F = old value, B0 = new value.

D03,04=B0
F03,04=5F
.
. Change the library prompt.
.
D0F,73="Library Area"
F0F,73="TRSDOS Ready"
.
. Clear the screen.
.
D05,68=1C 1F
F05,68=20 20
.
. Change the message.
.
D09,59="All systems are now a go!     "
F09,59="DoubleDuty is now operational."
.
.  End of patch.
```

CP/M HARDWARE CONVERSION FOR THE MODEL I & SYSTEM 80

by Bruce Orr

(System 80 details by Tim Bulluss)

[Reprinted from SYDTRUG NEWS, P.O. Box 297, Padstow, New South Wales 2211, Australia. Note that the SYSTEM 80 computer mentioned in this article was sold under the name PMC-80 here in the U.S.A. (and under the name Video Genie in some other parts of the world).]

This file explains how the Model 1 or System 80 can be adapted to run CP/M. There MIGHT be a problem if you use the tandard Dick Smith expansion unit for the System 80. (I don't know for sure - I use an LNW expansion board.) See note below.

COMPONENTS:

- 1 x 74LS04 hex inverter (see note below)
- 1 x 74LS367 hex tri-state bus driver
- 1 x 1 MΩ resistor
- 1 x 0.022µF greencap capacitor (Model I)
- 1 x 0.056µF greencap capacitor (System 80)
- 1 x SPST toggle switch
- 1 x push button (N.O.)
- PCB or Veroboard

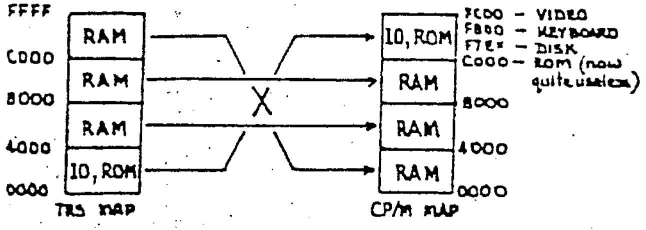
This modification adapts the System 80 for CP/M by:

a) Remapping memory so that RAM commences at 0000H.

b) Adding a reset switch (current reset switch generates an NMI, not a reset to location 0 as required by CP/M).

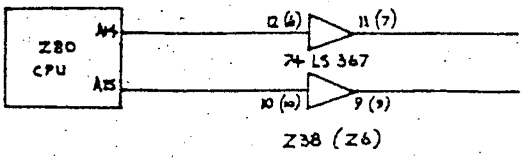
MEMORY REMAPPING

This is achieved by swapping the first 16K block of memory e ROM & I/O) with the last (part of RAM).

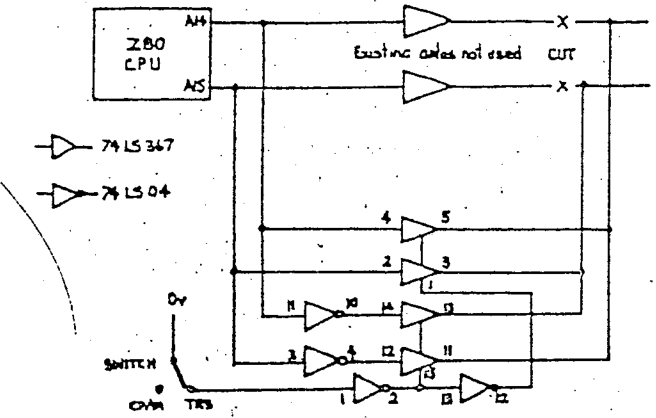


The CP/M memory mapping is achieved by inverting & swapping address bits A14 & A15 from the Z80 before they enter he address decoding circuits. In the following circuits all mbering refers to the Model 1 (System 80 numbering appears in parentheses).

EXISTING CIRCUIT:



ODIFIED CIRCUIT:



I suggest that this be built on a small PCB mounted underneath the main board near the Z80 CPU or by "piggy backing" individual chips on top of existing IC's.

SYSTEM 80 EXPANSION INTERFACE

NOTE : - This modification possibly makes computer unsuitable for use with the standard Sys 80 S100 expansion unit as A14 & A15 can no longer be disabled by the *DODBS/ADDBS line. If this would be a problem, suggest using an LS00 NAND gate in place of the LS04 inverter. Connect *DODBS/ADDBS to one input of both the gates which control the enables of the LS367.

RESET SWITCH

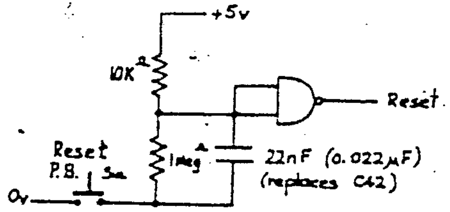
Requirements:

a) Must not disable power up reset.

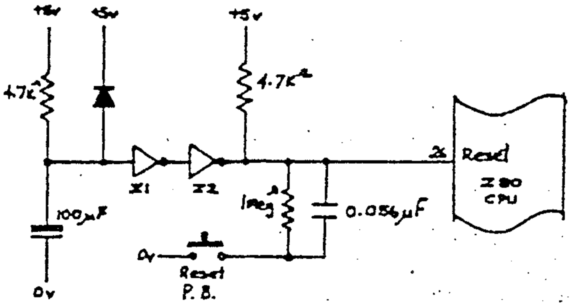
b) Provides sufficient reset pulse to reset the Z80.

c) While the Z80 is held reset, memory refreshing is halted. Thus, reset pulse must not be too long, even if reset switch is held closed.

MODEL 1 RESET



SYSTEM 80 RESET



In the above diagram, the only new components are the push button, the 1MΩ resistor & the 0.056µF greencap. They are connected to pin 26 of the Z80 as shown.

To avoid having two reset switches, suggest using a DPDT toggle for the change over switch, using the second pole to select the function of the existing reset switch.

TEST PROGRAMS

Two programs to test this modification follow the BIOS listing, they are CPMTEST1/ASM & CPMTEST2/ASM. They can be assembled using EDTASM. Operating instructions are included in the source code.

```
00100 ;  * * *  CPM BIOS FOR TRS-80 MODEL 1 & SYSTEM 80
00110 ;
00120 ;        VERSION 3.3
00130 ;        LAST UPDATE 21 AUG 1983
00140 ;
00150 ;        COPYRIGHT (C) 1983
00160 ;
00170 ;BY      BRUCE ORR       VK2F0
00180 ;        8 GLENSIDE STREET,
00190 ;        BALGOWLAH, 2093.
00200 ;        AUSTRALIA.     PH. (02) 94-6520
00210 ;
00220 ;        PERSONAL NON-PROFIT USE OF THIS
00230 ;        SOFTWARE IS AUTHORISED.
00240 ;
00250 ;THIS PROGRAM FORMS THE BIOS (BASIC INPUT/OUTPUT SYSTEM)
00260 ;REQUIRED TO INTERFACE THE MODEL 1 COMPUTER TO THE CP/M
00270 ;OPERATING SYSTEM. BOTH SINGLE & DOUBLE DENSITY FORMATS
```

```
                00280 ;ARE SUPPORTED WITH AUTOMATIC DENSITY RECOGNITION.  THE
                00290 ;SINGLE DENSITY FORMAT IS COMPATIBLE WITH THE OSBORNE
                00300 ;(TM) SINGLE DENSITY FORMAT WHILE THE DOUBLE DENSITY
                00310 ;FORMAT IS BASED ON THE TRS-80 MODEL 3/4 FORMAT.
                00320 ;
                00330 ;USE OF THIS PROGRAM REQUIRES THE FOLLOWING :-
                00340 ;
                00350 ;STANDARD EXPANSION INTERFACE WITH 48K MEMORY.
                00360 ;MEMORY MAPPING MODIFICATION WHICH EXCHANGES FIRST AND
                00370 ;LAST 16K MEMORY BLOCKS UNDER SWITCH CONTROL.
                00380 ;LICENCED COPY OF CP/M 2.2 48K ON THE ABOVE DISK FORMAT.
                00390 ;
                00400 ;THANKS TO TIM BULLUSS, NEIL TREMBLE AND HORST LEYKAM
                00410 ;FOR THEIR COMMENTS AND SUGGESTIONS.
                00420 ;
0004            00430 CDISK   EQU     0004H           ;LAST LOGGED DISK
0003            00440 IOBYTE  EQU     0003H           ;I/O CONTROL BYTE
0040            00450 SEKDSK  EQU     40H             ;SEEK DISK NUMBER
0041            00460 SEKTRK  EQU     41H             ;SEEK TRACK NUMBER
0042            00470 SEKSEC  EQU     42H             ;SEEK SECTOR NUMBER
0043            00480 HSTDSK  EQU     43H             ;HOST DISK NUMBER
0044            00490 HSTTRK  EQU     44H             ;HOST TRACK NUMBER
0045            00500 HSTSEC  EQU     45H             ;HOST SECTOR NUMBER
0046            00510 SEKHST  EQU     46H             ;SEEK SHR SECSHF
0047            00520 HSTACT  EQU     47H             ;HOST ACTIVE FLAG
0048            00530 HSTWRT  EQU     48H             ;HOST WRITTEN FLAG
0049            00540 UNACNT  EQU     49H             ;UNALLOC REC COUNT
004A            00550 UNADSK  EQU     4AH             ;LAST UNALLOC DISK
004B            00560 UNATRK  EQU     4BH             ;LAST UNALLOC TRACK
004C            00570 UNASEC  EQU     4CH             ;LAST UNALLOC SECTOR
004D            00580 ERFLAG  EQU     4DH             ;ERROR REPORTING FLAG
004E            00590 RSFLAG  EQU     4EH             ;READ SECTOR FLAG
004F            00600 READOP  EQU     4FH             ;1 IF READ OPERATION
002B            00610 KBD     EQU     2BH             ;ROM KEYBOARD ROUTINE
0033            00620 DSP     EQU     33H             ;ROM DISPLAY ROUTINE
402D            00630 DOS     EQU     402DH           ;TRSDOS RETURN ADD
00E8            00640 RSCTRL  EQU     0E8H            ;RS232 CONTROL
00E9            00650 RSBAUD  EQU     0E9H            ;RS232 BAUD RATE
00EA            00660 RSSTAT  EQU     0EAH            ;RS232 STATUS
00EB            00670 RSDATA  EQU     0EBH            ;RS232 DATA
F7E8            00680 PRTREG  EQU     0F7E8H          ;PRINTER REGISTER
F7EC            00690 DCSREG  EQU     0F7ECH          ;DISK COMMAND/STATUS REG
F7ED            00700 TRKREG  EQU     0F7EDH          ;DISK TRACK REG
F7EE            00710 SECREG  EQU     0F7EEH          ;DISK SECTOR REG
F7EF            00720 DATREG  EQU     0F7EFH          ;DISK DATA REG
F7E0            00730 DRVREG  EQU     0F7E0H          ;DISK DRIVE REG
0003            00740 NDISKS  EQU     3               ;NUMBER OF DISK DRIVES
002C            00750 NSECTS  EQU     44              ;NUMBER OF SECS TO LOAD
0800            00760 BLKSIZ  EQU     2048            ;CP/M ALLOC SIZE
0100            00770 HSTSIZ  EQU     256             ;HOST SECTOR SIZE
0002            00780 HSTBLK  EQU     2               ;CP/M SECTS/HOST BUFF
0010            00790 CPMSPB  EQU     16              ;CPM SECTORS PER BLOCK
0001            00800 SECMSK  EQU     1               ;SECT MASK (HSTBLK-1)
0000            00810 WRALL   EQU     0               ;WRITE TO ALLOCATED
0001            00820 WRDIR   EQU     1               ;WRITE TO DIRECTORY
0002            00830 WRUAL   EQU     2               ;WRITE TO UNALLOCATED
9F00            00840 CPMB    EQU     09F00H          ;CPM LOAD BASE ADDRESS
A706            00850 BDOS    EQU     0A706H          ;BDOS ENTRY POINT
B500            00860         ORG     0B500H          ;SET BIOS ORIGIN
B500 C368B5     00870 ORIGIN  JP      BOOT            ;COLD START
B503 C373B5     00880 WBOOTE  JP      WBOOT           ;WARM START
B506 C334B6     00890         JP      CONST           ;CONSOLE STATUS
B509 C355B6     00900         JP      CONIN           ;CONSOLE CHAR IN
B50C C391B6     00910         JP      CONOUT          ;CONSOLE CHAR OUT
B50F C3A5B6     00920         JP      LIST            ;LIST CHAR OUT
B512 C3D0B6     00930         JP      PUNCH           ;PUNCH OUT
B515 C3E0B6     00940         JP      READER          ;READER IN
B518 C301B7     00950         JP      HOME            ;RESTORE DISK HEAD POS
B51B C317B7     00960         JP      SELDSK          ;SELECT DISK
B51E C330B7     00970         JP      SETTRK          ;SET TRACK NUMBER
B521 C357B7     00980         JP      SETSEC          ;SET SECTOR NUMBER
B524 C35CB7     00990         JP      SETDMA          ;SET DMA ADDRESS
B527 C361B7     01000         JP      READ            ;READ DISK
B52A C36EB7     01010         JP      WRITE           ;WRITE DISK
B52D C3E1B9     01020         JP      LISTST          ;LIST DEVICE STATUS
B530 C3ECB9     01030         JP      SECTRN          ;SECTOR TRANSLATION
B533 C339B5     01040         JP      NU              ;NOT USED
B536 C339B5     01050         JP      NU
B539 C9         01060 NU      RET                     ;RETURN
B53A F3         01070 START   DI                      ;DISABLE INTERRUPTS
```

```
B53B 21D9BE     01080         LD      HL,MESS2        ;DISPLAY MESSAGE 2
B53E CDB0BA     01090         CALL    SDSP
B541 CD01B6     01100         CALL    CHECK           ;CALC CHECKSUM
B544 3278BC     01110         LD      (CKSUM),A       ;SAVE IT
B547 210000     01120 MAPTST  LD      HL,0            ;FIRST RAM LOCATION
B54A 3E55       01130         LD      A,55H
B54C 77         01140         LD      (HL),A          ;ATTEMPT WRITE
B54D BE         01150         CP      (HL)            ;WRITE SUCCESSFUL?
B54E 20F7       01160         JR      NZ,MAPTST       ;IF NOT WAIT
B550 0605       01170         LD      B,5             ;500MS DEBOUNCE DELAY
B552 CDD3B9     01180         CALL    DELAY
B555 3E00       01190         LD      A,0             ;SELECT DISK A
B557 320400     01200         LD      (CDISK),A
B55A 3E81       01210         LD      A,81H
B55C 320300     01220         LD      (IOBYTE),A      ;INITIALISE I/O BYTE
B55F 219ABE     01230         LD      HL,MESS1
B562 CD89BA     01240         CALL    SDISP           ;DISPLAY CP/M MESSAGE
B565 C373B5     01250         JP      WBOOT           ;LOAD SYSTEM
B568 318000     01260 BOOT    LD      SP,0080H        ;RESET STACK POINTER
B56B 3E00       01270         LD      A,0             ;SELECT DISK A INITIALLY
B56D 320400     01280         LD      (CDISK),A
B570 C3C2B5     01290         JP      GOCPM           ;INITIALIZE AND GOTO CP/M
B573 CD89B8     01300 WBOOT   CALL    BLINIT          ;INIT BLOCKING
B576 318000     01310         LD      SP,0080H        ;INITIALISE STACK POINTER
B579 0E00       01320         LD      C,0             ;SELECT DISK 0
B57B CD17B7     01330         CALL    SELDSK
B57E CD01B7     01340         CALL    HOME            ;GO TO TRACK 00
B581 062C       01350         LD      B,NSECTS        ;NUMBER OF SECS TO LOAD
B583 0E00       01360         LD      C,0             ;SET START TRACK
B585 CD30B7     01370         CALL    SETTRK
B588 1600       01380         LD      D,0             ;SET START SECTOR
B58A 21009F     01390         LD      HL,CPMB         ;SET CPM LOAD BASE
B58D C5         01400 LOAD1   PUSH    BC              ;SAVE SEC CNT, TRACK
B58E D5         01410         PUSH    DE              ;SAVE NEXT SEC TO READ
B58F E5         01420         PUSH    HL              ;SAVE DMA ADDRESS
B590 4A         01430         LD      C,D             ;GET SEC ADD TO REG C
B591 CD57B7     01440         CALL    SETSEC          ;SET SEC ADD FROM REG C
B594 C1         01450         POP     BC              ;RECALL DMA ADD TO BC
B595 C5         01460         PUSH    BC              ;PLACE BACK ON STACK
B596 CD5CB7     01470         CALL    SETDMA          ;SET DMA ADDRESS FROM BC
                01480 ;       DRIVE SET TO 0, TRACK SET, SECTOR SET, DMA SET
B599 CD61B7     01490         CALL    READ            ;READ LOGICAL SECTOR
B59C FE00       01500         CP      0               ;ANY ERRORS?
B59E C273B5     01510         JP      NZ,WBOOT        ;IF SO RE-BOOT
                01520 ;       NO ERROR, MOVE TO NEXT SECTOR
B5A1 E1         01530         POP     HL              ;RECALL DMA ADD
B5A2 118000     01540         LD      DE,128          ;DMA=DMA+128
B5A5 19         01550         ADD     HL,DE           ;NEW DMA ADD IN HL
B5A6 D1         01560         POP     DE              ;RECALL SEC ADD
B5A7 C1         01570         POP     BC              ;RECALL SECS REMAINING
B5A8 05         01580         DEC     B               ;SECS=SECS-1
B5A9 CAC2B5     01590         JP      Z,GOCPM         ;GOTO CP/M WHEN COMPLETE
                01600 ;       MORE SECTORS TO LOAD, CHECK FOR TRACK CHANGE
B5AC 14         01610         INC     D               ;INCREMENT SECTOR COUNT
B5AD 7A         01620         LD      A,D
B5AE FE14       01630         CP      20              ;IF SEC>19 CHANGE TRACK
B5B0 DA8DB5     01640         JP      C,LOAD1         ;ELSE CONTINUE
                01650 ;       END OF CURRENT TRACK, GO TO NEXT
B5B3 1600       01660         LD      D,0             ;BEGIN WITH FIRST SEC
B5B5 0C         01670         INC     C               ;TRACK=TRACK+1
                01680 ;       SAVE REG STATES AND CHANGE TRACKS
B5B6 C5         01690         PUSH    BC
B5B7 D5         01700         PUSH    DE
B5B8 E5         01710         PUSH    HL
B5B9 CD30B7     01720         CALL    SETTRK
B5BC E1         01730         POP     HL
B5BD D1         01740         POP     DE
B5BE C1         01750         POP     BC
B5BF C38DB5     01760         JP      LOAD1           ;GET NEXT SECTOR
                01770 ;       END OF LOAD, SET PARAMETERS AND GO TO CP/M
B5C2 CDF6B6     01780 GOCPM   CALL    RSINIT          ;INITIALISE RS232 PORT
B5C5 CD89B8     01790         CALL    BLINIT          ;INITIALISE BLOCKING
B5C8 3EC3       01800         LD      A,0C3H          ;JUMP INSTRUCTION
B5CA 320000     01810         LD      (0),A           ;FOR JUMP TO WBOOT
B5CD 2103B5     01820         LD      HL,WBOOTE       ;WBOOT ENTRY POINT
B5D0 220100     01830         LD      (1),HL
B5D3 320500     01840         LD      (5),A           ;FOR JUMP TO BDOS
B5D6 2106A7     01850         LD      HL,BDOS         ;BDOS ENTRY POINT
B5D9 220600     01860         LD      (6),HL
B5DC 3EC9       01870         LD      A,0C9H          ;RETURN INSTRUCTION CODE
```

```
B5DE 320800   01880          LD     (08H),A         ;STORE AT INTERRUPTS 1-7
B5E1 321000   01890          LD     (10H),A
B5E4 321800   01900          LD     (18H),A
B5E7 322000   01910          LD     (20H),A
B5EA 322800   01920          LD     (28H),A
B5ED 323000   01930          LD     (30H),A
B5F0 323800   01940          LD     (38H),A
B5F3 018000   01950          LD     BC,80H          ;SET DEFAULT DMA ADD
B5F6 CD5CB7   01960          CALL   SETDMA
B5F9 F3       01970          DI                     ;DISABLE INTERRUPTS
B5FA 3A0400   01980          LD     A,(CDISK)       ;GET CURRENT DRIVE
B5FD 4F       01990          LD     C,A             ;SEND TO THE CCP
B5FE C3009F   02000          JP     CPMB            ;GO TO CP/M
B601 2A0BB9   02010 CHECK    LD     HL,(TRNFER)     ;SAVE TRNFER WORD
B604 E5       02020          PUSH   HL
B605 210000   02030          LD     HL,0
B608 220BB9   02040          LD     (TRNFER),HL     ;ZERO IT
B60B 2100B5   02050          LD     HL,ORIGIN       ;LOAD START ADD
B60E 1139BC   02060          LD     DE,CKEND+1      ;LOAD END ADD
B611 0E00     02070          LD     C,0             ;ZERO SUM
B613 79       02080 CHECKL   LD     A,C             ;GET SUM
B614 86       02090          ADD    A,(HL)          ;ADD NEXT BYTE
B615 4F       02100          LD     C,A             ;SAVE SUM
B616 23       02110          INC    HL              ;POINT TO NEXT BYTE
B617 7C       02120          LD     A,H             ;END REACHED ?
B618 BA       02130          CP     D
B619 20F8     02140          JR     NZ,CHECKL
B61B 7D       02150          LD     A,L
B61C BB       02160          CP     E
B61D 20F4     02170          JR     NZ,CHECKL
B61F 79       02180          LD     A,C             ;YES, PUT RESULT IN A
B620 E1       02190          POP    HL              ;RESTORE TRNFER WORD
B621 220BB9   02200          LD     (TRNFER),HL
B624 C9       02210          RET                    ;RETURN
B625 CD01B6   02220 VERCHK   CALL   CHECK           ;CALCULATE CHECKSUM
B628 217BBC   02230          LD     HL,CKSUM
B62B BE       02240          CP     (HL)            ;SAME AS STORED VALUE?
B62C C8       02250          RET    Z               ;RETURN IF SO
B62D 219ABF   02260          LD     HL,MESS5
B630 CD89BA   02270          CALL   SDISP           ;DISPLAY MESSAGE 5
B633 C9       02280          RET                    ;RETURN
              02290 ;        I/O HANDLER SECTION BEGINS HERE
B634 CD7DB6   02300 CONST    CALL   KEYF            ;CHECK KEYBOARD
B637 326FBC   02310          LD     (CHAR),A        ;SAVE RESULT
B63A A7       02320          AND    A               ;SET FLAGS
B63B 2012     02330          JR     NZ,CONRDY       ;JUMP IF READY
B63D 3A0300   02340          LD     A,(IOBYTE)      ;GET I/O BYTE
B640 E601     02350          AND    1
B642 200E     02360          JR     NZ,CONIDL       ;EXIT ON CON MODES 1,3
B644 DBEA     02370          IN     A,(RSSTAT)      ;GET RS232 STATUS
B646 CB7F     02380          BIT    7,A             ;HAS CHAR ARRIVED ?
B648 2808     02390          JR     Z,CONIDL        ;JUMP IF NOT
B64A DBEB     02400          IN     A,(RSDATA)      ;GET CHAR
B64C 326FBC   02410          LD     (CHAR),A        ;SAVE IT
B64F 3EFF     02420 CONRDY   LD     A,0FFH          ;SET READY INDICATION
B651 C9       02430          RET                    ;RETURN
B652 3E00     02440 CONIDL   LD     A,0             ;SET NOT READY INDICATION
B654 C9       02450          RET                    ;RETURN
              02460 ;        CONSOLE CHARACTER INTO REGISTER A
B655 3A6FBC   02470 CONIN    LD     A,(CHAR)        ;GET CHARACTER
B658 E67F     02480          AND    7FH             ;RESET HIGH BIT
B65A 2808     02490          JR     Z,WAITK         ;JUMP IF NO CHAR
B65C 4F       02500          LD     C,A             ;SAVE CHAR
B65D 3E00     02510          LD     A,0
B65F 326FBC   02520          LD     (CHAR),A        ;CLEAR CHAR REG
B662 79       02530          LD     A,C             ;GET CHAR
B663 C9       02540          RET                    ;RETURN
B664 CD7DB6   02550 WAITK    CALL   KEYF            ;CHECK KEYBOARD
B667 E67F     02560          AND    7FH             ;RESET HIGH BIT
B669 A7       02570          AND    A
B66A C0       02580          RET    NZ              ;RETURN IF KEY PRESSED
B66B 3A0300   02590          LD     A,(IOBYTE)      ;GET I/O BYTE
B66E E601     02600          AND    1
B670 20F2     02610          JR     NZ,WAITK        ;WAIT FOR KEY IF MODE 1,3
B672 DBEA     02620          IN     A,(RSSTAT)      ;ELSE GET RS232 STATUS
B674 CB7F     02630          BIT    7,A
B676 28EC     02640          JR     Z,WAITK         ;WAIT IF NO CHAR READY
B678 DBEB     02650          IN     A,(RSDATA)      ;GET CHAR
B67A E67F     02660          AND    7FH             ;RESET HIGH BIT
B67C C9       02670          RET                    ;RETURN
```

```
B67D 3A02F8   02680 KEYF     LD     A,(0F802H)      ;IF JKL PRESSED
B680 FE1C     02690          CP     1CH
B682 CCC7B6   02700          CALL   Z,PRTSCR        ;THEN PRINT SCREEN
B685 3A10F8   02710          LD     A,(0F810H)      ;IF 123 PRESSED
B688 FE0E     02720          CP     0EH
B68A CC98B8   02730          CALL   Z,TERMIN        ;THEN TERMINAL MODE
B68D CDC4BA   02740          CALL   KEY             ;SCAN KEYBOARD
B690 C9       02750          RET
              02760 ;        CONSOLE CHARACTER OUTPUT FROM REGISTER C
B691 79       02770 CONOUT   LD     A,C             ;PUT CHAR IN A
B692 C5       02780          PUSH   BC
B693 CDF7B9   02790          CALL   DISP            ;DISPLAY IT
B696 C1       02800          POP    BC
B697 3A0300   02810          LD     A,(IOBYTE)      ;GET I/O BYTE
B69A E603     02820          AND    03H             ;MASK CONSOLE FIELD
B69C CAE3B6   02830          JP     Z,RS232O        ;0 - SEND TO RS232 PORT
B69F FE02     02840          CP     2
B6A1 CAA5B6   02850          JP     Z,LIST          ;2 - SEND TO LIST DEVICE
B6A4 C9       02860          RET                    ;1,3 - CRT DISP ONLY
              02870 ;        LIST CHARACTER FROM REGISTER C
B6A5 3A0300   02880 LIST     LD     A,(IOBYTE)      ;GET I/O BYTE
B6A8 07       02890          RLCA                   ;EXTRACT LIST DEV NUMBER
B6A9 07       02900          RLCA
B6AA E603     02910          AND    03H
B6AC CAE3B6   02920          JP     Z,RS232O        ;0 - SEND TO RS232
B6AF FE02     02930          CP     2
B6B1 CAB8B6   02940          JP     Z,PRINT         ;2 - SEND TO PRINTER
B6B4 79       02950          LD     A,C
B6B5 C3F7B9   02960          JP     DISP            ;1,3 - SEND TO CRT DISP
              02970 ;        PRINT CHARACTER FROM REGISTER C
B6B8 3AE8F7   02980 PRINT    LD     A,(PRTREG)      ;GET PRINTER STATUS
B6BB CB7F     02990          BIT    7,A             ;IS IT BUSY ?
B6BD 20E6     03000          JR     NZ,LIST         ;IF SO WAIT
B6BF 79       03010          LD     A,C             ;ELSE GET CHAR
B6C0 FE0A     03020          CP     0AH             ;FILTER OUT LF
B6C2 C8       03030          RET    Z
B6C3 32E8F7   03040          LD     (PRTREG),A      ;PRINT CHAR
B6C6 C9       03050          RET                    ;RETURN
              03060 ;        SCREEN PRINT ROUTINE
B6C7 2100FC   03070 PRTSCR   LD     HL,0FC00H       ;START OF SCREEN
B6CA 0640     03080 PRTLIN   LD     B,64            ;SET LINE LENGTH
B6CC 4E       03090 PRTCHR   LD     C,(HL)          ;GET CHAR
B6CD CDB8B6   03100          CALL   PRINT           ;PRINT IT
B6D0 23       03110          INC    HL              ;ADVANCE POINTER
B6D1 10F9     03120          DJNZ   PRTCHR          ;LOOP TILL EOL
B6D3 0E0D     03130          LD     C,0DH
B6D5 CDB8B6   03140          CALL   PRINT           ;SEND CARR RETURN
B6D8 7C       03150          LD     A,H
B6D9 A7       03160          AND    A               ;END OF SCREEN?
B6DA 20EE     03170          JR     NZ,PRTLIN       ;LOOP IF NOT
B6DC C9       03180          RET                    ;RETURN
              03190 ;        PUNCH CHARACTER OUT
B6DD C3E3B6   03200 PUNCH    JP     RS232O          ;SEND TO RS232 PORT
              03210 ;        READER CHARACTER IN
B6E0 C3EDB6   03220 READER   JP     RS232I          ;GET FROM RS232 PORT
              03230 ;        SEND CHARACTER TO RS232 PORT
B6E3 DBEA     03240 RS232O   IN     A,(RSSTAT)      ;GET UART STATUS
B6E5 CB77     03250          BIT    6,A             ;IS TX REG EMPTY ?
B6E7 28FA     03260          JR     Z,RS232O        ;IF NOT WAIT
B6E9 79       03270          LD     A,C             ;GET CHARACTER
B6EA D3EB     03280          OUT    (RSDATA),A      ;SEND IT
B6EC C9       03290          RET                    ;RETURN
              03300 ;        READ CHARACTER FROM RS232 PORT
B6ED DBEA     03310 RS232I   IN     A,(RSSTAT)      ;GET UART STATUS
B6EF CB7F     03320          BIT    7,A             ;HAS CHAR ARRIVED ?
B6F1 28FA     03330          JR     Z,RS232I        ;IF NOT WAIT
B6F3 DBEB     03340          IN     A,(RSDATA)      ;LOAD CHAR TO A
B6F5 C9       03350          RET                    ;RETURN
              03360 ;        INITIALISE RS232 PORT
B6F6 D3E8     03370 RSINIT   OUT    (RSCTRL),A      ;RESET UART
B6F8 3E55     03380          LD     A,55H           ;SELECT 300 BAUD
B6FA D3E9     03390          OUT    (RSBAUD),A
B6FC 3E6F     03400          LD     A,6FH           ;8 BITS,1 STOP,NO PAR
B6FE D3EA     03410          OUT    (RSSTAT),A
B700 C9       03420          RET                    ;RETURN
              03430 ;        DISK I/O DRIVER SECTION FOLLOWS
              03440 ;        MOVE TO THE TRACK 00 POSITION OF CURRENT DRIVE
B701 3A4000   03450 HOME     LD     A,(SEKDSK)      ;GET SELECTED DISK
B704 CD94B9   03460          CALL   DSKRDY          ;WAIT TILL DISK READY
B707 3E02     03470 HOME1    LD     A,02            ;RESTORE COMMAND
```

```
8709 32ECF7   03480        LD    (DCSREG),A     ;SEND TO CONTROLLER
870C CDDEB9   03490        CALL  SDELAY
870F 3AECF7   03500 HOMEW  LD    A,(DCSREG)     ;GET DISK STATUS
8712 CB47     03510        BIT   0,A
8714 20F9     03520        JR    NZ,HOMEW       ;WAIT IF BUSY
8716 C9       03530        RET                  ;ELSE RETURN
              03540 ;      SELECT DISK GIVEN BY REGISTER C
8717 210000   03550 SELDSK LD    HL,0           ;ERROR CODE
871A 79       03560        LD    A,C            ;GET DRIVE NUMBER
871B FE03     03570        CP    NDISKS         ;TOO LARGE?
871D D0       03580        RET   NC             ;RETURN WITH HL=0 IF SO
871E 324000   03590        LD    (SEKDSK),A     ;SEEK DISK NUMBER
8721 69       03600        LD    L,C            ;HL=DRIVE NUMBER
8722 2600     03610        LD    H,0
8724 29       03620        ADD   HL,HL          ;HL=HL*16
8725 29       03630        ADD   HL,HL
8726 29       03640        ADD   HL,HL
8727 29       03650        ADD   HL,HL
8728 11398C   03660        LD    DE,DCB1        ;DE=DISK CONT BLOCK ADD.
872B 19       03670        ADD   HL,DE          ;ADD DRIVE OFFSET
872C 22798C   03680        LD    (CURDCB),HL    ;SAVE CURRENT DCB ADD
872F C9       03690        RET                  ;RETURN
              03700 ;      SET TRACK GIVEN BY REGISTER C
8730 79       03710 SETTRK LD    A,C
8731 324100   03720        LD    (SEKTRK),A     ;SAVE TRACK TO SEEK
8734 C9       03730        RET                  ;RETURN
8735 3A4300   03740 SEEK   LD    A,(HSTDSK)     ;GET CURRENT DISK
8738 CD9489   03750        CALL  DSKRDY         ;READY DISK
873B 3A4400   03760        LD    A,(HSTTRK)     ;GET TRACK TO SEEK
873E 32EFF7   03770        LD    (DATREG),A     ;SEND TO DISK
8741 3E12     03780        LD    A,12H
8743 32ECF7   03790        LD    (DCSREG),A     ;PERFORM TRACK SEEK
8746 CDDEB9   03800        CALL  SDELAY         ;SHORT DELAY
8749 3AECF7   03810 SEEKW  LD    A,(DCSREG)     ;GET DISK STATUS
874C CB47     03820        BIT   0,A
874E 20F9     03830        JR    NZ,SEEKW       ;WAIT TILL FINISHED
8750 3A4500   03840        LD    A,(HSTSEC)     ;GET SECTOR TO SEEK
8753 32EEF7   03850        LD    (SECREG),A     ;SEND TO DISK
8756 C9       03860        RET                  ;THEN RETURN
              03870 ;      SET SECTOR GIVEN BY REGISTER C
8757 79       03880 SETSEC LD    A,C
8758 324200   03890        LD    (SEKSEC),A     ;SAVE SECTOR TO SEEK
875B C9       03900        RET                  ;RETURN
              03910 ;      SET DMA ADDRESS GIVEN BY REGISTER BC
875C ED43698C 03920 SETDMA LD    (DMAADR),BC    ;SAVE DMA ADDRESS
8760 C9       03930        RET                  ;RETURN
              03940 ;      READ THE SELECTED SECTOR
8761 3E01     03950 READ   LD    A,1
8763 324F00   03960        LD    (READOP),A     ;FLAG READ OPERATION
8766 3E02     03970        LD    A,WRUAL
8768 32718C   03980        LD    (WRTYPE),A     ;TREAT AS UNALLOC
876B C3D387   03990        JP    ALLOC          ;TO PERFORM THE READ
876E 3E00     04000 WRITE  LD    A,0
8770 324F00   04010        LD    (READOP),A     ;NOT A READ OPERATION
8773 79       04020        LD    A,C            ;LOAD WRITE TYPE
8774 32718C   04030        LD    (WRTYPE),A     ;STORE IT
8777 FE02     04040        CP    WRUAL          ;WRITE UNALLOCATED ?
8779 2017     04050        JR    NZ,CHKUNA      ;CHECK FOR UNALLOC
              04060 ;      WRITE TO UNALLOCATED, SET PARAMETERS
877B 3E10     04070        LD    A,CPMSPB       ;CPM SECTORS PER BLOCK
877D 324900   04080        LD    (UNACNT),A     ;NEXT UNALLOC RECS
8780 3A4000   04090        LD    A,(SEKDSK)     ;DISK TO SEEK
8783 324A00   04100        LD    (UNADSK),A     ;UNADSK = SEKDSK
8786 3A4100   04110        LD    A,(SEKTRK)
8789 324B00   04120        LD    (UNATRK),A     ;UNATRK = SEKTRK
878C 3A7C8C   04130        LD    A,(LOGSEC)
878F 324C00   04140        LD    (UNASEC),A     ;UNASEC = LOGSEC
              04150 ;      CHECK FOR WRITE TO UNALLOCATED SECTOR
8792 3A4900   04160 CHKUNA LD    A,(UNACNT)     ;ANY UNALLOC REMAIN ?
8795 B7       04170        OR    A
8796 CAD387   04180        JP    Z,ALLOC        ;SKIP IF NOT
              04190 ;      MORE UNALLOCATED RECORDS REMAIN
8799 3D       04200        DEC   A              ;UNACNT = UNACNT-1
879A 324900   04210        LD    (UNACNT),A
879D 3A4000   04220        LD    A,(SEKDSK)     ;SAME DISK ?
87A0 214A00   04230        LD    HL,UNADSK
87A3 BE       04240        CP    (HL)           ;SEKDSK = UNADSK ?
87A4 C2D387   04250        JP    NZ,ALLOC       ;SKIP IF NOT
              04260 ;      DISKS ARE THE SAME
87A7 3A4100   04270        LD    A,(SEKTRK)
```

```
87AA 214B00   04280        LD    HL,UNATRK
87AD BE       04290        CP    (HL)           ;SEKTRK = UNATRK ?
87AE C2D387   04300        JP    NZ,ALLOC       ;SKIP IF NOT
              04310 ;      TRACKS ARE THE SAME
87B1 3A7C8C   04320        LD    A,(LOGSEC)
87B4 214C00   04330        LD    HL,UNASEC
87B7 BE       04340        CP    (HL)           ;LOGSEC = UNASEC ?
87B8 C2D387   04350        JP    NZ,ALLOC
              04360 ;      MATCH, MOVE TO NEXT SECTOR FOR FUTURE REF
87BB 34       04370        INC   (HL)           ;UNASEC = UNASEC+1
87BC 46       04380        LD    B,(HL)         ;END OF TRACK ?
87BD 3A738C   04390        LD    A,(CPMSPT)     ;GET SECTORS PER TRACK
87C0 3D       04400        DEC   A
87C1 B8       04410        CP    B
87C2 D2CB87   04420        JP    NC,NOOVF       ;SKIP IF NO OVERFLOW
              04430 ;      OVERFLOW TO NEXT TRACK
87C5 3600     04440        LD    (HL),0         ;UNASEC = 0
87C7 214B00   04450        LD    HL,UNATRK
87CA 34       04460        INC   (HL)           ;UNATRK = UNATRK+1
              04470 ;      MATCH FOUND, MARK AS UNNECESSARY READ
87CB 3E00     04480 NOOVF  LD    A,0
87CD 324E00   04490        LD    (RSFLAG),A     ;RSFLAG = 0
87D0 C3DC87   04500        JP    RWOPER         ;TO PERFORM THE WRITE
              04510 ;      NOT UNALLOCATED RECORD, REQUIRES PRE-READ
87D3 3E00     04520 ALLOC  LD    A,0
87D5 324900   04530        LD    (UNACNT),A     ;UNACNT = 0
87D8 3C       04540        INC   A
87D9 324E00   04550        LD    (RSFLAG),A     ;RSFLAG = 1
              04560 ;      ENTER HERE TO PERFORM THE READ/WRITE
87DC 3E00     04570 RWOPER LD    A,0
87DE 324D00   04580        LD    (ERFLAG),A     ;NO ERRORS (YET)
87E1 3A4200   04590        LD    A,(SEKSEC)     ;GET SECTOR TO SEEK
87E4 CB3F     04600        SRL   A              ;COMPUTE HOST SECTOR ***
87E6 3C       04610        INC   A
87E7 324600   04620        LD    (SEKHST),A     ;HOST SECTOR TO SEEK
              04630 ;      ACTIVE HOST SECTOR ?
87EA 214700   04640        LD    HL,HSTACT      ;HOST ACTIVE FLAG
87ED 7E       04650        LD    A,(HL)
87EE 3601     04660        LD    (HL),1         ;ALWAYS BECOMES 1
87F0 B7       04670        OR    A              ;WAS IT ACTIVE ALREADY ?
87F1 CA2388   04680        JP    Z,FILHST       ;FILL HOST IF NOT
              04690 ;      HOST BUFFER ACTIVE, SAME AS SEEK BUFFER ?
87F4 3A4000   04700        LD    A,(SEKDSK)
87F7 214300   04710        LD    HL,HSTDSK      ;SAME DISK ?
87FA BE       04720        CP    (HL)           ;SEKDSK = HSTDSK ?
87FB C21C88   04730        JP    NZ,NOMTCH
              04740 ;      SAME DISK, SAME TRACK ?
87FE 3A4100   04750        LD    A,(SEKTRK)
8801 214400   04760        LD    HL,HSTTRK
8804 BE       04770        CP    (HL)           ;SEKTRK = HSTTRK ?
8805 C21C88   04780        JP    NZ,NOMTCH
              04790 ;      SAME DISK, SAME TRACK, SAME BUFFER ?
8808 3A4600   04800        LD    A,(SEKHST)
880B 214500   04810        LD    HL,HSTSEC      ;SECHST = HSTSEC ?
880E BE       04820        CP    (HL)
880F C21C88   04830        JP    NZ,NOMTCH
8812 3A4200   04840        LD    A,(SEKSEC)
8815 217D8C   04850        LD    HL,LSTSEC      ;SEKSEC = LSTSEC ?
8818 BE       04860        CP    (HL)
8819 C24788   04870        JP    NZ,MATCH       ;REDO IF EXACT MATCH
              04880 ;      CORRECT DISK, BUT INCORRECT SECTOR
881C 3A4800   04890 NOMTCH LD    A,(HSTWRT)     ;HOST WRITTEN ?
881F B7       04900        OR    A
8820 C4A288   04910        CALL  NZ,WRTHST      ;CLEAR HOST BUFFER
              04920 ;      MAY HAVE TO FILL HOST BUFFER
8823 3A4000   04930 FILHST LD    A,(SEKDSK)
8826 324300   04940        LD    (HSTDSK),A
8829 3A4100   04950        LD    A,(SEKTRK)
882C 324400   04960        LD    (HSTTRK),A
882F 3A4600   04970        LD    A,(SEKHST)
8832 324500   04980        LD    (HSTSEC),A
8835 3A4200   04990        LD    A,(SEKSEC)
8838 327D8C   05000        LD    (LSTSEC),A
883B 3A4E00   05010        LD    A,(RSFLAG)     ;NEED TO READ ?
883E B7       05020        OR    A
883F C49288   05030        CALL  NZ,RDHST       ;YES, IF 1
8842 3E00     05040        LD    A,0
8844 324800   05050        LD    (HSTWRT),A     ;NO PENDING WRITE
              05060 ;      COPY DATA TO OR FROM BUFFER
8847 3A4200   05070 MATCH  LD    A,(SEKSEC)     ;MASK BUFFER NUMBER
```

```
884A E601    05080          AND   SECMSK          ;LEAST SIG BITS
884C 6F      05090          LD    L,A             ;READY TO SHIFT
884D 2600    05100          LD    H,0
884F 29      05110          ADD   HL,HL           ;SHIFT LEFT 7
8850 29      05120          ADD   HL,HL
8851 29      05130          ADD   HL,HL
8852 29      05140          ADD   HL,HL
8853 29      05150          ADD   HL,HL
8854 29      05160          ADD   HL,HL
8855 29      05170          ADD   HL,HL
     11858C  05180          LD    DE,HSTBUF       ;HOST BUFFER BASE
     19      05190          ADD   HL,DE           ;ADD OFFSET
885A EB      05200          EX    DE,HL           ;PUT RESULT IN DE
885B 2A698C  05210          LD    HL,(DMAADR)     ;CPM DATA ADDRESS
885E 0E80    05220          LD    C,128
8860 0600    05230          LD    B,0             ;128 BYTES TO MOVE
8862 3A4F00  05240          LD    A,(READOP)      ;WHICH WAY ?
8865 B7      05250          OR    A
8866 C26F88  05260          JP    NZ,RWMOVE       ;SKIP IF READ
             05270 ;        WRITE OPERATION, MARK AND SWITCH DIRECTION
8869 3E01    05280          LD    A,1
886B 324800  05290          LD    (HSTWRT),A      ;HSTWRT = 1
886E EB      05300          EX    DE,HL           ;SOURCE/DEST SWAP
             05310 ;        MOVE 128 BYTES FROM DE TO HL
886F EB      05320 RWMOVE   EX    DE,HL
8870 EDB0    05330          LDIR                  ;MOVE DATA BLOCK
             05340 ;        DATA HAS BEEN MOVED TO/FROM HOST BUFFER
8872 3A718C  05350          LD    A,(WRTYPE)      ;WRITE TYPE
8875 FE01    05360          CP    WRDIR           ;TO DIRECTORY ?
8877 3A4D00  05370          LD    A,(ERFLAG)      ;IN CASE OF ERRORS
887A C0      05380          RET   NZ              ;NO FURTHER PROCESSING
             05390 ;        CLEAR HOST BUFFER FOR DIRECTORY WRITE
     B7      05400          OR    A               ;ERRORS ?
887C C0      05410          RET   NZ              ;SKIP IF SO
887D 3E00    05420          LD    A,0
887F 324800  05430          LD    (HSTWRT),A      ;BUFFER WRITTEN
8882 CDA288  05440          CALL  WRTHST
8885 3A4D00  05450          LD    A,(ERFLAG)
8888 C9      05460          RET                   ;RETURN
             05470 ;        INITIALISE BLOCKING FLAGS
8889 3E00    05480 BLINIT   LD    A,0
888B 324700  05490          LD    (HSTACT),A      ;HOST BUFFER INACTIVE
888E 324900  05500          LD    (UNACNT),A      ;CLEAR  UNALLOC COUNT
8891 C9      05510          RET                   ;RETURN
             05520 ;        READ PHYSICAL SECTOR INTO SECBUF
8892 3E1C    05530 RDHST    LD    A,1CH           ;SET ERROR MASK
     32728C  05540          LD    (ERMASK),A
     211A02  05550          LD    HL,021AH        ;SET TRANSFER DIRECTION
889A 220B89  05560          LD    (TRNFER),HL
889D 3E8C    05570          LD    A,8CH           ;SECTOR READ COMMAND
889F C3B288  05580          JP    DISKRW          ;PERFORM READ
             05590 ;        WRITE PHYSICAL SECTOR FROM SECBUF
88A2 3E7C    05600 WRTHST   LD    A,7CH           ;SET ERROR MASK
88A4 32728C  05610          LD    (ERMASK),A
88A7 21120A  05620          LD    HL,0A12H        ;SET TRANSFER DIRECTION
88AA 220B89  05630          LD    (TRNFER),HL
88AD 3EAC    05640          LD    A,0ACH          ;SECTOR WRITE COMMAND
88AF C3B288  05650          JP    DISKRW          ;PERFORM WRITE
             05660 ;        DISK SECTOR READ/WRITE
     32748C  05670 DISKRW   LD    (DSKCOM),A      ;SAVE DISK COMMAND
88B5 CD3587  05680          CALL  SEEK            ;SEEK DISK
88B8 3E06    05690          LD    A,6             ;MAX RETRY COUNT
88BA 32758C  05700          LD    (TRYCNT),A
88BD 21828C  05710          LD    HL,DENTAB       ;DENSITY TABLE BASE
88C0 3A4300  05720          LD    A,(HSTDSK)      ;CURRENT DISK NUMBER
88C3 5F      05730          LD    E,A
88C4 1600    05740          LD    D,0
88C6 19      05750          ADD   HL,DE           ;ADD TO BASE ADD
88C7 22778C  05760          LD    (DENPT),HL      ;SAVE POINTER TO DENSITY
88CA 3A4300  05770 RETRY    LD    A,(HSTDSK)      ;GET CURRENT DISK
88CD CD9489  05780          CALL  DSKRDY          ;READY DISK
88D0 2A778C  05790          LD    HL,(DENPT)      ;GET DENSITY POINTER
88D3 7E      05800          LD    A,(HL)          ;GET DENSITY MODE
88D4 F6FE    05810          OR    0FEH            ;SET BITS 1-7
88D6 21ECF7  05820          LD    HL,DCSREG       ;DISK COMMAND/STAT REG
88D9 77      05830          LD    (HL),A          ;SET DENSITY
88DA 3600    05840          LD    (HL),0D0H       ;FORCE INTERRUPT
88DC 3A4500  05850          LD    A,(HSTSEC)      ;TRANSFER CURRENT SECTOR
88DF 32EEF7  05860          LD    (SECREG),A
88E2 3A4400  05870          LD    A,(HSTTRK)      ;TRANSFER CURRENT TRACK
```

```
88E5 32EDF7   05880          LD    (TRKREG),A
88E8 11EFF7   05890          LD    DE,DATREG       ;DISK DATA REGISTER
88EB 01858C   05900          LD    BC,HSTBUF       ;DISK BUFFER BASE ADD
88EE 3A748C   05910          LD    A,(DSKCOM)      ;GET COMMAND
88F1 77       05920          LD    (HL),A          ;ISSUE COMMAND
88F2 CDDE89   05930          CALL  SDELAY          ;SHORT DELAY
88F5 CB6F     05940          BIT   5,A
88F7 CA0589   05950          JP    Z,WAIT2         ;JUMP IF READ
88FA 7E       05960 WAIT1    LD    A,(HL)          ;WAIT FOR DATA REQUEST
88FB E683     05970          AND   83H
88FD E2FA88   05980          JP    PO,WAIT1
8900 0A       05990          LD    A,(BC)          ;GET FIRST BYTE TO WRT
8901 03       06000          INC   BC              ;INCREMENT POINTER
8902 C30B89   06010          JP    TRNFER          ;BEGIN TRANSFER
8905 7E       06020 WAIT2    LD    A,(HL)          ;GET DISK STATUS
8906 E683     06030          AND   83H             ;READY FOR FIRST BYTE ?
8908 E20589   06040          JP    PO,WAIT2        ;WAIT IF NOT
890B 0A       06050 TRNFER   LD    A,(BC)          ;N.B. READ CASE SHOWN
890C 12       06060          LD    (DE),A          ;INSTUCTS CHANGED FOR WRT
890D 03       06070          INC   BC              ;INC BUFFER POINTER
890E CB4E     06080 WAIT3    BIT   1,(HL)          ;READY FOR NEXT BYTE?
8910 20F9     06090          JR    NZ,TRNFER       ;IF SO JUMP
8912 CB4E     06100          BIT   1,(HL)
8914 20F5     06110          JR    NZ,TRNFER
8916 CB4E     06120          BIT   1,(HL)
8918 20F1     06130          JR    NZ,TRNFER
891A CB46     06140          BIT   0,(HL)          ;OPERATION COMPLETE ?
891C 2812     06150          JR    Z,RWCOMP
891E CB4E     06160          BIT   1,(HL)
8920 20E9     06170          JR    NZ,TRNFER
8922 CB7E     06180          BIT   7,(HL)          ;TIMED OUT ?
8924 28E8     06190          JR    Z,WAIT3         ;LOOP IF NOT
8926 3600     06200          LD    (HL),0D0H       ;FORCE INTERRUPT
8928 215DBF   06210          LD    HL,MESS3
892B CD898A   06220          CALL  SDISP
892E 185E     06230          JR    RWERR
8930 7E       06240 RWCOMP   LD    A,(HL)          ;GET STATUS
8931 32768C   06250          LD    (ERCODE),A      ;SAVE ERROR CODE
8934 5F       06260          LD    E,A
8935 3A728C   06270          LD    A,(ERMASK)
8938 A3       06280          AND   E               ;MASK CODE
8939 324D00   06290          LD    (ERFLAG),A
893C C8       06300          RET   Z               ;RETURN IF NO ERROR
893D 3A758C   06310          LD    A,(TRYCNT)      ;GET RETRY COUNT
8940 3D       06320          DEC   A               ;DECREMENT IT
8941 32758C   06330          LD    (TRYCNT),A      ;SAVE IT
8944 2848     06340          JR    Z,RWERR         ;ERROR IF COUNT = 0
8946 3A748C   06350          LD    A,(DSKCOM)      ;GET DISK COMMAND
8949 CB6F     06360          BIT   5,A
894B C2CA88   06370          JP    NZ,RETRY        ;JUMP IF WRITE
894E 2A778C   06380          LD    HL,(DENPT)      ;LOAD DENSITY POINTER
8951 7E       06390          LD    A,(HL)          ;GET DENSITY MODE
8952 2F       06400          CPL                   ;COMPLEMENT IT
8953 E601     06410          AND   1               ;MASK IT
8955 77       06420          LD    (HL),A          ;SAVE NEW DENSITY
8956 2008     06430          JR    NZ,DOUBLE
8958 01E288   06440 SINGLE   LD    BC,SECTN1       ;SECTOR TRANSLATE TAB 1
895B 111A8C   06450          LD    DE,DPB1         ;DISK PARAM BLOCK 1
895E 1806     06460          JR    SETDEN          ;SET NEW DENSITY
8960 01F688   06470 DOUBLE   LD    BC,SECTN2       ;SECTOR TRANSLATE TAB 2
8963 11298C   06480          LD    DE,DPB2         ;DISK PARAM BLOCK 2
8966 DD2A798C 06490 SETDEN   LD    IX,(CURDCB)     ;CURRENT DCB ADDRESS
896A DD7100   06500          LD    (IX+0),C        ;INSERT NEW TRANSLATE
896D DD7001   06510          LD    (IX+1),B        ;TABLE ADDRESS
8970 DD730A   06520          LD    (IX+0AH),E      ;INSERT NEW DISK
8973 DD720B   06530          LD    (IX+0BH),D      ;PARAMETER BLOCK ADDRESS
8976 ED43D084 06540          LD    (084D0H),BC     ;GIVE BDOS NEW SECTRAN
897A ED53B384 06550          LD    (084B3H),DE     ;GIVE BDOS NEW DPB
897E 1A       06560          LD    A,(DE)          ;GET NEW SECTS PER TRACK
897F 32738C   06570          LD    (CPMSPT),A      ;SAVE IT
8982 21C184   06580          LD    HL,084C1H       ;BDOS WORKING DPB ADD
8985 EB       06590          EX    DE,HL           ;PUT LOCAL DPB ADD IN HL
8986 010F00   06600          LD    BC,0FH
8989 EDB0     06610          LDIR                  ;TRANSFER 15 BYTES
898B C3CA88   06620          JP    RETRY           ;RETRY SECTOR READ
898E 3E01     06630 RWERR    LD    A,1             ;ELSE SET ERROR FLAG
8990 324D00   06640          LD    (ERFLAG),A
8993 C9       06650          RET                   ;AND RETURN
8994 4F       06660 DSKRDY   LD    C,A
8995 3E01     06670          LD    A,1             ;INITIALISE A
```

```
B997 0C       06680         INC     C
B998 0D       06690 SEL1    DEC     C
B999 2803     06700         JR      Z,SEL2
B99B 87       06710         ADD     A,A             ;SHIFT ACCUM LEFT
B99C 18FA     06720         JR      SEL1
B99E 3268BC   06730 SEL2    LD      (DRVCOD),A      ;SAVE DRIVE SELECT CODE
B9A1 3AECF7   06740         LD      A,(DCSREG)      ;GET DISK STATUS
B9A4 CB7F     06750         BIT     7,A             ;IS MOTOR ON?
B9A6 3A68BC   06760         LD      A,(DRVCOD)      ;GET DRIVE SELECT CODE
B9A9 32E0F7   06770         LD      (DRVREG),A      ;SELECT DRIVE
B9AC 2808     06780         JR      Z,MOTON         ;JUMP IF MOTOR ON
B9AE 0605     06790         LD      B,5             ;500MS MOTOR ON DELAY
B9B0 CDD3B9   06800         CALL    DELAY
B9B3 CD25B6   06810         CALL    VERCHK          ;PERFORM CHECKSUM TEST
B9B6 3A68BC   06820 MOTON   LD      A,(DRVCOD)      ;GET DRIVE CODE
B9B9 47       06830         LD      B,A
B9BA 3A6CBC   06840         LD      A,(LSTCOD)      ;GET LAST CODE
B9BD B8       06850         CP      B               ;SAME AS NEW CODE?
B9BE 2808     06860         JR      Z,READY
B9C0 CD07B7   06870         CALL    HOME1           ;HOME HEAD IF NOT
B9C3 0601     06880         LD      B,1             ;AND DELAY 100MS
B9C5 CDD3B9   06890         CALL    DELAY
B9C8 3A68BC   06900 READY   LD      A,(DRVCOD)      ;SELECT DRIVE
B9CB 326CBC   06910         LD      (LSTCOD),A      ;UPDATE LAST DRIVE CODE
B9CE 32E0F7   06920         LD      (DRVREG),A
B9D1 F3       06930         DI                      ;DISABLE INTERRUPTS
B9D2 C9       06940         RET                     ;RETURN
B9D3 11A31A   06950 DELAY   LD      DE,1AA3H        ;SET CORE TIME
B9D6 1B       06960 LOOP    DEC     DE              ;DECREMENT CORE TIME
B9D7 7B       06970         LD      A,E             ;DE=0 ?
B9D8 B2       06980         OR      D
B9D9 20FB     06990         JR      NZ,LOOP         ;LOOP IF NOT
B9DB 10F6     07000         DJNZ    DELAY           ;DECREMENT TIME
B9DD C9       07010         RET                     ;AND RETURN
B9DE C5       07020 SDELAY  PUSH    BC              ;SHORT DELAY
B9DF C1       07030         POP     BC
B9E0 C9       07040         RET                     ;RETURN
              07050 ;       GET LIST DEVICE STATUS
B9E1 3AE8F7   07060 LISTST  LD      A,(PRTREG)      ;GET PRINTER STATUS
B9E4 E680     07070         AND     80H             ;MASK BUSY BIT
B9E6 3E00     07080         LD      A,0
B9E8 C0       07090         RET     NZ              ;RETURN WITH 0 IF BUSY
B9E9 3EFF     07100         LD      A,0FFH          ;ELSE RETURN FFH
B9EB C9       07110         RET
              07120 ;       SECTOR TRANSLATION ROUTINE
B9EC 79       07130 SECTRN  LD      A,C
B9ED 327CBC   07140         LD      (LOGSEC),A      ;SAVE LOGICAL SECTOR
B9F0 69       07150         LD      L,C             ;HL=LOGICAL SECTOR
B9F1 60       07160         LD      H,B
B9F2 19       07170         ADD     HL,DE           ;ADD MAP ADDRESS
B9F3 6E       07180         LD      L,(HL)          ;HL=PHYSICAL SECTOR
B9F4 2600     07190         LD      H,0
B9F6 C9       07200         RET                     ;RETURN
              07210 ;       VIDEO CONSOLE DISPLAY ROUTINE
B9F7 FE80     07220 DISP    CP      80H
B9F9 D0       07230         RET     NC              ;RETURN IF CHAR>80H
B9FA 2A6DBC   07240         LD      HL,(CURPOS)     ;GET CURSOR POSITION
B9FD FE20     07250         CP      20H
B9FF DA40BA   07260         JP      C,CONT          ;JUMP IF CONTROL CHAR
BA02 FE5B     07270         CP      5BH             ;IF CHR=5B,5C,5D,5E
BA04 380E     07280         JR      C,DSPCHR
BA06 FE5F     07290         CP      5FH
BA08 300A     07300         JR      NC,DSPCHR       ;THEN
BA0A EB       07310         EX      DE,HL           ;LOAD CHAR FROM TABLE
BA0B 2183B8   07320         LD      HL,CHRTAB-5BH
BA0E 0600     07330         LD      B,0
BA10 4F       07340         LD      C,A
BA11 09       07350         ADD     HL,BC
BA12 7E       07360         LD      A,(HL)
BA13 EB       07370         EX      DE,HL
BA14 77       07380 DSPCHR  LD      (HL),A          ;DISPLAY CHARACTER
BA15 23       07390         INC     HL              ;ADVANCE CURSOR
BA16 7C       07400 CURSET  LD      A,H
BA17 A7       07410         AND     A               ;SEE IF OFF SCREEN
BA18 CC21BA   07420         CALL    Z,SCROLL        ;IF SO SCROLL DISPLAY
BA1B 365F     07430         LD      (HL),5FH        ;WRITE CURSOR
BA1D 226DBC   07440         LD      (CURPOS),HL     ;SAVE CURSOR POSITION
BA20 C9       07450         RET                     ;RETURN
BA21 1100FC   07460 SCROLL  LD      DE,0FC00H       ;POINT TO TOP OF DISP
BA24 2140FC   07470         LD      HL,0FC40H       ;POINT TO ROW 2
BA27 01C003   07480         LD      BC,03C0H        ;15 ROWS TO MOVE
BA2A EDB0     07490         LDIR                    ;PERFORM SCROLL
BA2C 3E20     07500         LD      A,20H           ;LOAD BLANK
BA2E 32C0FF   07510         LD      (0FFC0H),A      ;STORE AT BEG. ROW 16
BA31 21C0FF   07520         LD      HL,0FFC0H       ;POINT TO FIRST CHAR
BA34 11C1FF   07530         LD      DE,0FFC1H       ;POINT TO SECOND CHAR
BA37 013F00   07540         LD      BC,63           ;63 CHARS TO MOV
BA3A EDB0     07550         LDIR                    ;BLANK ROW 16
BA3C 21C0FF   07560         LD      HL,0FFC0H       ;SET CURSOR POSITION
BA3F C9       07570         RET                     ;RETURN
BA40 FE08     07580 CONT    CP      08H
BA42 CA8CBA   07590         JP      Z,BACKSP        ;BACKSPACE
BA45 FE0A     07600         CP      0AH
BA47 CA82BA   07610         JP      Z,LINEFD        ;LINE FEED
BA4A FE0D     07620         CP      0DH
BA4C CA70BA   07630         JP      Z,CARRET        ;CARRIAGE RETURN
BA4F FE0C     07640         CP      0CH
BA51 CA5ABA   07650         JP      Z,CLEAR         ;CLEAR SCREEN (FF)
BA54 FE07     07660         CP      07H
BA56 C397BA   07670         JP      BELL            ;BELL
BA59 C9       07680         RET                     ;RETURN
BA5A 3E20     07690 CLEAR   LD      A,20H           ;BLANK CHAR. AT
BA5C 3200FC   07700         LD      (0FC00H),A      ;FIRST SCREEN POS.
BA5F 2100FC   07710         LD      HL,0FC00H
BA62 1101FC   07720         LD      DE,0FC01H
BA65 01FF03   07730         LD      BC,03FFH
BA68 EDB0     07740         LDIR                    ;BLANK SCREEN OUT
BA6A 2100FC   07750         LD      HL,0FC00H       ;HOME CURSOR
BA6D C316BA   07760         JP      CURSET          ;SET CURSOR
BA70 CD7BBA   07770 CARRET  CALL    CURDEL          ;DELETE CURSOR
BA73 7D       07780         LD      A,L             ;LOAD LOW BYTE
BA74 E6C0     07790         AND     0C0H            ;GO TO BEG. OF LINE
BA76 6F       07800         LD      L,A
BA77 226DBC   07810         LD      (CURPOS),HL     ;SAVE CURSOR POSITION
BA7A C9       07820         RET                     ;RETURN
BA7B 7E       07830 CURDEL  LD      A,(HL)          ;GET CHAR AT CURSOR
BA7C FE5F     07840         CP      5FH             ;CURSOR CHAR?
BA7E C0       07850         RET     NZ              ;RETURN IF NOT
BA7F 3620     07860         LD      (HL),20H        ;IF SO BLANK IT
BA81 C9       07870         RET                     ;RETURN
BA82 CD7BBA   07880 LINEFD  CALL    CURDEL          ;DELETE CURSOR
BA85 114000   07890         LD      DE,40H
BA88 19       07900         ADD     HL,DE           ;MOVE DOWN ONE L
BA89 C316BA   07910         JP      CURSET          ;SET CURSOR
BA8C 7C       07920 BACKSP  LD      A,H             ;SEE IF CURSOR
BA8D D6FC     07930         SUB     0FCH            ;AT HOME POSITION
BA8F B5       07940         OR      L
BA90 C8       07950         RET     Z               ;IF SO ABORT BACKSP
BA91 3620     07960         LD      (HL),20H        ;ERASE CURSOR
BA93 2B       07970         DEC     HL              ;MOVE CUR. BACK ONE
BA94 C316BA   07980         JP      CURSET          ;SET CURSOR
BA97 21C800   07990 BELL    LD      HL,200          ;200MS LENGTH
BA9A 0644     08000 BELL1   LD      B,68            ;1KHZ FREQ
BA9C 10FE     08010 BELL2   DJNZ    BELL2           ;TIME 1/2 CYCLE
BA9E 3E00     08020         LD      A,0
BAA0 D3FF     08030         OUT     (255),A         ;OUTPUT IT
BAA2 0644     08040         LD      B,68
BAA4 10FE     08050 BELL3   DJNZ    BELL3           ;TIME 1/2 CYCLE
BAA6 3E01     08060         LD      A,1
BAA8 D3FF     08070         OUT     (255),A         ;OUTPUT IT
BAAA 2B       08080         DEC     HL              ;DEC CYCLE COUNT
BAAB 7C       08090         LD      A,H
BAAC B5       08100         OR      L               ;IS IT ZERO?
BAAD 20EB     08110         JR      NZ,BELL1        ;LOOP IF NOT
BAAF C9       08120         RET
              08130 ;       STRING DISPLAY ROUTINES
BAB0 7E       08140 SDSP    LD      A,(HL)          ;GET CHAR TO DISPLAY
BAB1 A7       08150         AND     A               ;TERMINATOR?
BAB2 C8       08160         RET     Z               ;RETURN IF SO
BAB3 CD3300   08170         CALL    DSP             ;ELSE DISPLAY IT
BAB6 23       08180         INC     HL              ;INC STRING POINTER
BAB7 18F7     08190         JR      SDSP            ;LOOP TILL DONE
BAB9 7E       08200 SDISP   LD      A,(HL)          ;GET CHAR TO DISPLAY
BABA A7       08210         AND     A               ;TERMINATOR?
BABB C8       08220         RET     Z               ;RETURN IF SO
BABC E5       08230         PUSH    HL              ;SAVE POINTER
BABD CDF7B9   08240         CALL    DISP            ;DISPLAY CHAR
BAC0 E1       08250         POP     HL              ;RESTORE POINTER
BAC1 23       08260         INC     HL              ;INC STRING POINT
BAC2 18F5     08270         JR      SDISP           ;LOOP TILL DONE
```

```
                   ;      KEYBOARD SCAN ROUTINE
BAC4 3AE0F7  KEY     LD    A,(DRVREG)      ;AUTO REPEAT FUNCTION
BAC7 CB7F            BIT   7,A             ;CHECK FOR 25MS PULSE
BAC9 2824            JR    Z,KEY1
BACB 3A7EBC          LD    A,(KEYACT)      ;CHECK FOR ACTIVE KEY
BACE A7              AND   A
BACF 281E            JR    Z,KEY1
BAD1 3A80BC          LD    A,(RPTLAG)      ;GET REPEAT LAG
BAD4 A7              AND   A
BAD5 2806            JR    Z,REPEAT        ;REPEAT IF EXPIRED
BAD7 3D              DEC   A               ;ELSE COUNTDOWN
BAD8 3280BC          LD    (RPTLAG),A      ;AND SAVE
BADB 2012            JR    NZ,KEY1
BADD 3A81BC  REPEAT  LD    A,(RPTRAT)      ;GET REPEAT RATE
BAE0 3D              DEC   A               ;COUNTDOWN
BAE1 3281BC          LD    (RPTRAT),A      ;SAVE IT
BAE4 2009            JR    NZ,KEY1
BAE6 3E03            LD    A,3             ;RESET RATE
BAE8 3281BC          LD    (RPTRAT),A      ;SAVE IT
BAEB 3A7FBC          LD    A,(LSTKEY)      ;GET LAST KEY PRESSED
BAEE C9              RET                   ;AND RETURN
BAEF 2105BE  KEY1    LD    HL,KEYBUF       ;KEYBOARD BUFFER ADD
BAF2 0101F8          LD    BC,0F801H       ;BC=ROW ADD POINT
BAF5 1600            LD    D,0             ;D=ROW COUNTER
BAF7 0A      NXTROW  LD    A,(BC)          ;GET ROW
BAF8 5F              LD    E,A             ;SAVE IN E
BAF9 AE              XOR   (HL)            ;CHANGE IN ROW ?
BAFA 280A            JR    Z,NOCHNG        ;SKIP IF NO CHANGE
BAFC F5              PUSH  AF
BAFD AF              XOR   A
BAFE 327EBC          LD    (KEYACT),A      ;CANCEL ACTIVE FLAG
BB01 F1              POP   AF
BB02 73              LD    (HL),E          ;SAVE BYTE
BB03 A3              AND   E               ;IS KEY PRESSED?
BB04 2008            JR    NZ,PRESS
BB06 14      NOCHNG  INC   D               ;INC ROW COUNTER
BB07 23              INC   HL              ;INC BUFF POINTER
BB08 CB01            RLC   C               ;POINT TO NEXT ROW
BB0A F2F7BA          JP    P,NXTROW        ;LOOP TILL DONE
BB0D C9              RET                   ;RET IF NO KEY PRESSED
BB0E 5F      PRESS   LD    E,A             ;SAVE ROW BYTE
BB0F 3E01            LD    A,1
BB11 327EBC          LD    (KEYACT),A      ;SET KEY ACTIVE FLAG
BB14 3E1E            LD    A,30
BB16 3280BC          LD    (RPTLAG),A      ;SET DELAY BEFORE REPEAT
BB19 7A              LD    A,D             ;GET ROW COUNT
BB1A 07              RLCA
BB1B 07              RLCA
BB1C 07              RLCA
BB1D 57              LD    D,A             ;D=ROW COUNT * 8
BB1E 0E01            LD    C,1             ;SET TESTING MASK
BB20 79      NXTCOL  LD    A,C             ;LOAD MASK
BB21 A3              AND   E               ;TEST FOR KEY
BB22 2005            JR    NZ,FOUND        ;JUMP IF FOUND
BB24 14              INC   D               ;INC COLUMN COUNT
BB25 CB01            RLC   C               ;SHIFT TEST BIT LEFT
BB27 18F7            JR    NXTCOL          ;TEST NEXT COLUMN
BB29 3A80F8  FOUND   LD    A,(0F880H)      ;GET SHIFT BIT
BB2C 47              LD    B,A             ;SAVE IN B
BB2D 7A              LD    A,D             ;A=(ROW*8)+COL
BB2E C640            ADD   A,40H           ;ADD 40H (FOR ASCII LETS)
BB30 FE60            CP    60H             ;TEST FOR NON-LETTER
BB32 3018            JR    NC,NONLET       ;JUMP IF NON-LETTER
BB34 57              LD    D,A             ;SAVE CHAR IN D
BB35 3A40F8          LD    A,(0F840H)      ;GET ROW 6 BITS
BB38 E610            AND   10H             ;TEST FOR DOWN ARROW
BB3A 200B            JR    NZ,CONTRL       ;IF SO CONVERT TO CONTROL
BB3C 3A70BC          LD    A,(SHFLOC)      ;GET SHIFT LOCK STATUS
BB3F B8              CP    B               ;DOES IT MATCH SHIFT KEY?
BB40 283E            JR    Z,KDEL          ;IF SO GO TO DELAY ROUT
BB42 7A              LD    A,D             ;RETRIEVE CHAR
BB43 C620            ADD   A,20H           ;CONVERT TO LOWER CASE
BB45 1838            JR    KDELAY          ;GO TO DELAY ROUT
BB47 7A      CONTRL  LD    A,D             ;RETRIEVE CHAR
BB48 D640            SUB   40H             ;TURN INTO CONTROL CHAR
BB4A 1833            JR    KDELAY          ;GO TO DELAY ROUTINE
BB4C D670    NONLET  SUB   70H             ;TEST FOR LAST ROW
BB4E 3021            JR    NC,LASTRW       ;JUMP IF LAST ROW
BB50 C640            ADD   A,40H           ;ADJUST FOR ROWS 4,5
BB52 2140F8          LD    HL,0F840H       ;CHECK FOR CTRL
BB55 CB66            BIT   4,(HL)
BB57 200E            JR    NZ,EXCONT
BB59 FE3C            CP    3CH             ;IF CHAR < = > ?
BB5B 3802            JR    C,NON1
BB5D EE10            XOR   10H             ;THEN TOGGLE SHIFT
BB5F CB08    NON1    RRC   B               ;GET SHIFT BIT
BB61 301C            JR    NC,KDELAY       ;JUMP IF NO SHIFT
BB63 EE10            XOR   10H             ;ADJUST ASCII
BB65 1818            JR    KDELAY          ;GO TO DELAY ROUTINE
BB67 5F      EXCONT  LD    E,A             ;GENERATE POINTER
BB68 1600            LD    D,0
BB6A 219EBB          LD    HL,CONTAB-30H   ;RELATIVE TABLE BASE
BB6D 19              ADD   HL,DE
BB6E 7E              LD    A,(HL)          ;GET CHAR FROM TABLE
BB6F 180E            JR    KDELAY
BB71 07      LASTRW  RLCA                  ;A=(ROW*8+COL-48)*2
BB72 CB08            RRC   B               ;GET SHIFT BIT
BB74 3001            JR    NC,LAST1        ;JUMP IF NO SHIFT
BB76 3C              INC   A               ;A=COL*2 + 1
BB77 21BEBB  LAST1   LD    HL,KEYTAB       ;POINT TO KEY TABLE
BB7A 4F              LD    C,A             ;GET DISPLACEMENT
BB7B 0600            LD    B,0
BB7D 09              ADD   HL,BC           ;COMPUTE TABLE POSITION
BB7E 7E              LD    A,(HL)          ;GET ASCII CODE
BB7F 57      KDELAY  LD    D,A             ;SAVE CHARACTER
BB80 010015  KDEL    LD    BC,1500H        ;LOAD DELAY COUNT
BB83 0B      KLOOP   DEC   BC              ;DEC DELAY COUNTER
BB84 78              LD    A,B             ;SEE IF ZERO
BB85 B1              OR    C
BB86 20FB            JR    NZ,KLOOP        ;LOOP IF NOT
BB88 7A              LD    A,D             ;RETRIEVE CHARACTER
BB89 327FBC          LD    (LSTKEY),A      ;SAVE IT
BB8C FEFF            CP    0FFH            ;SHIFT LOCK CHAR?
BB8E C0              RET   NZ              ;RETURN IF NOT
BB8F 3A70BC          LD    A,(SHFLOC)      ;GET SHIFT LOCK STATUS
BB92 2F              CPL                   ;COMPLEMENT IT
BB93 E601            AND   1               ;MASK IT
BB95 3270BC          LD    (SHFLOC),A      ;SAVE IT
BB98 3E00            LD    A,0             ;ZERO ACC
BB9A C9              RET                   ;RETURN
                   ;      DATA TERMINAL MODE DRIVER
BB9B 216CBF  TERMIN  LD    HL,MESS4
BB9E CDB9BA          CALL  SDISP           ;DISPLAY MESSAGE 4
BBA1 CDC4BA  TERMI   CALL  KEY             ;SCAN KEYBOARD
BBA4 E67F            AND   7FH
BBA6 2807            JR    Z,TERMO
BBA8 FE05            CP    5               ;CTRL-E PRESSED?
BBAA C8              RET   Z               ;EXIT IF SO
BBAB 4F              LD    C,A
BBAC CDE3B6          CALL  RS232O          ;ELSE SEND CHAR
BBAF DBEA    TERMO   IN    A,(RSSTAT)      ;GET RS232 STATUS
BBB1 CB7F            BIT   7,A             ;CHAR RECEIVED?
BBB3 28EC            JR    Z,TERMI         ;LOOP IF NOT
BBB5 DBE9            IN    A,(RSDATA)      ;GET CHAR
BBB7 E67F            AND   7FH             ;RESET HIGH BIT
BBB9 CDF7B9          CALL  DISP            ;DISPLAY CHARACTER
BBBC 18E3            JR    TERMI           ;AND LOOP
BBBE 0D00    KEYTAB  DEFW  000DH           ;ENTER
BBC0 7FFF            DEFW  0FF7FH          ;CLEAR
BBC2 0303            DEFW  0303H           ;BREAK
BBC4 5E18            DEFW  185EH           ;UP ARROW
BBC6 0000            DEFW  0000H           ;DOWN ARROW
BBC8 0818            DEFW  1808H           ;LEFT ARROW
BBCA 0919            DEFW  1909H           ;RIGHT ARROW
BBCC 2020            DEFW  2020H           ;SPACE BAR
BBCE 1B      CONTAB  DEFB  1BH             ;CTRL 0
BBCF 1C              DEFB  1CH             ;CTRL 1
BBD0 1D              DEFB  1DH             ;CTRL 2
BBD1 1E              DEFB  1EH             ;CTRL 3
BBD2 1F              DEFB  1FH             ;CTRL 4
BBD3 7C              DEFB  7CH             ;CTRL 5
BBD4 7E              DEFB  7EH             ;CTRL 6
BBD5 7F              DEFB  7FH             ;CTRL 7
BBD6 5B              DEFB  5BH             ;CTRL 8
BBD7 5D              DEFB  5DH             ;CTRL 9
BBD8 00              DEFB  0               ;CTRL :
BBD9 00              DEFB  0               ;CTRL ;
BBDA 7B              DEFB  7BH             ;CTRL <
BBDB 5F              DEFB  5FH             ;CTRL =
BBDC 7D              DEFB  7DH             ;CTRL >
```

```
BBDD 5C        09880          DEFB    5CH             ;CTRL ?
BBDE B7        09890 CHRTAB   DEFB    183             ;LEFT SQ BRACKET
BBDF B9        09900          DEFB    B9H             ;BACKSLASH
BBE0 BB        09910          DEFB    187             ;RIGHT SQ BRACKET
BBE1 5B        09920          DEFB    5BH             ;CARET
               09930 ;        SECTOR TRANSLATION TABLE
BBE2 0001      09940 SECTN1   DEFW    0100H
BBE4 0405      09950          DEFW    0504H
BBE6 0809      09960          DEFW    0908H
BBE8 0C0D      09970          DEFW    0D0CH
BBEA 1011      09980          DEFW    1110H
BBEC 0203      09990          DEFW    0302H
BBEE 0607      10000          DEFW    0706H
BBF0 0A0B      10010          DEFW    0B0AH
BBF2 0E0F      10020          DEFW    0F0EH
BBF4 1213      10030          DEFW    1312H
BBF6 0001      10040 SECTN2   DEFW    0100H
BBF8 0405      10050          DEFW    0504H
BBFA 0809      10060          DEFW    0908H
BBFC 0C0D      10070          DEFW    0D0CH
BBFE 1011      10080          DEFW    1110H
BC00 1415      10090          DEFW    1514H
BC02 1819      10100          DEFW    1918H
BC04 1C1D      10110          DEFW    1D1CH
BC06 2021      10120          DEFW    2120H
BC08 0203      10130          DEFW    0302H
BC0A 0607      10140          DEFW    0706H
BC0C 0A0B      10150          DEFW    0B0AH
BC0E 0E0F      10160          DEFW    0F0EH
BC10 1213      10170          DEFW    1312H
BC12 1617      10180          DEFW    1716H
BC14 1A1B      10190          DEFW    1B1AH
BC16 1E1F      10200          DEFW    1F1EH
BC18 2223      10210          DEFW    2322H
               10220 ;        SINGLE DENSITY (OSBORNE) DISK PARAMETER BLOCK
BC1A 1400      10230 DPB1     DEFW    0014H           ;SECTORS/TRACK
BC1C 04        10240          DEFB    04H             ;BLOCK SHIFT
BC1D 0F        10250          DEFB    0FH             ;BLOCK MASK
BC1E 01        10260          DEFB    01H             ;EXTENT MASK
BC1F 2C00      10270          DEFW    002CH           ;DISK SIZE (2K LUMPS)
               10280                                  ;26H=35T, 2CH=40T
BC21 3F00      10290          DEFW    003FH           ;MAX DIR ENTRIES
BC23 80        10300          DEFB    80H             ;ALLOCATION 0
BC24 00        10310          DEFB    00H             ;ALLOCATION 1
BC25 1000      10320          DEFW    0010H           ;CHECKSIZE
BC27 0300      10330          DEFW    0003H           ;TRACK OFFSET
               10340 ;        DOUBLE DENSITY DISK PARAMETER BLOCK
BC29 2400      10350 DPB2     DEFW    0024H           ;SECTORS/TRACK
BC2B 04        10360          DEFB    04H             ;BLOCK SHIFT
BC2C 0F        10370          DEFB    0FH             ;BLOCK MASK
BC2D 01        10380          DEFB    01H,            ;EXTENT MASK
BC2E 5100      10390          DEFW    0051H           ;DISK SIZE (2K LUMPS)
               10400                                  ;46H=35T, 51H=40T
BC30 3F00      10410          DEFW    003FH           ;MAX DIR ENTRIES
BC32 80        10420          DEFB    80H             ;ALLOCATION 0
BC33 00        10430          DEFB    00H             ;ALLOCATION 1
BC34 1000      10440          DEFW    0010H           ;CHECKSIZE
BC36 0300      10450          DEFW    0003H           ;TRACK OFFSET
BC38 00        10460 CKEND    NOP                     ;CHECKSUM FINISHES HERE
               10470 ;        DISK CONTROL BLOCKS
BC39 E2BB      10480 DCB1     DEFW    SECTN1          ;SEC TABLE POINTER
BC3B 0000      10490          DEFW    0               ;ZEROS
BC3D 0000      10500          DEFW    0
BC3F 0000      10510          DEFW    0
BC41 85BD      10520          DEFW    DIRBUF          ;DIRECTORY BUFF ADD
BC43 1ABC      10530          DEFW    DPB1            ;DISK PARAM BLOCK ADD
BC45 0DBE      10540          DEFW    DCS1            ;DIR CHECK SCATCH 1
BC47 3DBE      10550          DEFW    AVA1            ;ALLOC VECT AREA 1
BC49 E2BB      10560 DCB2     DEFW    SECTN1          ;SEC TABLE POINTER
BC4B 0000      10570          DEFW    0               ;ZEROS
BC4D 0000      10580          DEFW    0
BC4F 0000      10590          DEFW    0
BC51 85BD      10600          DEFW    DIRBUF          ;DIRECTORY BUFF ADD
BC53 1ABC      10610          DEFW    DPB1            ;DISK PARAM BLOCK ADD
BC55 1DBE      10620          DEFW    DCS2            ;DIR CHECK SCRATCH 2
BC57 5CBE      10630          DEFW    AVA2            ;ALLOC VECT AREA 2
BC59 E2BB      10640 DCB3     DEFW    SECTN1
BC5B 0000      10650          DEFW    0
BC5D 0000      10660          DEFW    0
BC5F 0000      10670          DEFW    0
```

```
BC61 85BD      10680          DEFW    DIRBUF
BC63 1ABC      10690          DEFW    DPB1
BC65 2DBE      10700          DEFW    DCS3
BC67 7BBE      10710          DEFW    AVA3
               10720 ;        VARIABLE STORAGE AREA
BC69 8000      10730 DMAADR   DEFW    80H             ;DMA ADDRESS
BC6B 00        10740 DRVCOD   DEFB    0               ;DRIVE SELECT CODE
BC6C 00        10750 LSTCOD   DEFB    0               ;LAST DRIVE SELECT CO
BC6D 00FC      10760 CURPOS   DEFW    0FC00H          ;CURSOR POSITION
BC6F 00        10770 CHAR     DEFB    0               ;CHARACTER BUFFER
BC70 01        10780 SHFLOC   DEFB    1               ;SHIFT LOCK STATUS
BC71 00        10790 WRTYPE   DEFB    0               ;WRITE OPERATION TYPE
BC72 00        10800 ERMASK   DEFB    0               ;ERROR MASK
BC73 14        10810 CPMSPT   DEFB    20              ;CP/M SECTORS PER TRACK
BC74 00        10820 DSKCOM   DEFB    0               ;DISK COMMAND
BC75 00        10830 TRYCNT   DEFB    0               ;RETRY COUNT
BC76 00        10840 ERCODE   DEFB    0               ;DISK ERROR CODE
BC77 0000      10850 DENPT    DEFW    0               ;DENSITY POINTER
BC79 0000      10860 CURDCB   DEFW    0               ;CURRENT DCB ADDRESS
BC7B 00        10870 CKSUM    DEFB    0               ;CHECKSUM
BC7C 00        10880 LOGSEC   DEFB    0               ;LOGICAL SECTOR NUMBER
BC7D 00        10890 LSTSEC   DEFB    0               ;LAST SECTOR
BC7E 00        10900 KEYACT   DEFB    0               ;KEY ACTIVE FLAG
BC7F 00        10910 LSTKEY   DEFB    0               ;LAST KEY PRESSED
BC80 00        10920 RPTLAG   DEFB    0               ;KEY REPEAT LAG
BC81 03        10930 RPTRAT   DEFB    3               ;KEY REPEAT RATE
               10940 ;        DISK DRIVE DENSITY TABLE
BC82 00        10950 DENTAB   DEFB    0               ;DRIVE A
BC83 00        10960          DEFB    0               ;DRIVE B
BC84 00        10970          DEFB    0               ;DRIVE C
               10980 ;        BUFFERS AND SCRATCH AREAS
0100           10990 HSTBUF   DEFS    256             ;SECTOR BUFFER
0080           11000 DIRBUF   DEFS    128             ;DIRECTORY BUFFER
0008           11010 KEYBUF   DEFS    8               ;KEYBOARD BUFFER
0010           11020 DCS1     DEFS    16              ;DIR CHECK SCRATCH
0010           11030 DCS2     DEFS    16
0010           11040 DCS3     DEFS    16
001F           11050 AVA1     DEFS    31              ;ALLOC VECTORS AREA
001F           11060 AVA2     DEFS    31
001F           11070 AVA3     DEFS    31
               11080 ;        CONSOLE MESSAGE STORAGE AREA
BE9A 0C        11090 MESS1    DEFB    0CH
BE9B 43        11100          DEFM    'CP/M VERS 2.2A 48K'
     50 2F 4D 20 56 45 52 53 20 32 2E 32 41 20 34 38
     4B
BEAD 0D0A      11110          DEFW    0A0DH
BEAF 43        11120          DEFM    'COPYRIGHT (C), 1981'
     4F 50 59 52 49 47 48 54 20 28 43 29 2C 20 31 39
     38 31
BEC2 0D0A      11130          DEFW    0A0DH
BEC4 44        11140          DEFM    'DIGITAL RESEARCH'
     49 47 49 54 41 4C 20 52 45 53 45 41 52 43 48
BED4 0A0A      11150          DEFW    0A0AH
BED6 0D0A      11160          DEFW    0A0DH
BED8 00        11170          DEFB    0
BED9 1C1F      11180 MESS2    DEFW    1F1CH
BEDB 54        11190          DEFM    'TRS-80 CP/M BIOS 48K VER 3.3'
     52 53 2D 38 30 20 43 50 2F 4D 20 42 49 4F 53 20
     34 38 4B 20 56 45 52 20 33 2E 33
BEF7 0D        11200          DEFB    0DH
BEF8 43        11210          DEFM    'COPYRIGHT (C) BRUCE ORR 1983'
     4F 50 59 52 49 47 48 54 20 28 43 29 20 42 52 55
     43 45 20 4F 52 52 20 31 39 38 33
BF14 0D0D      11220          DEFW    0D0DH
BF16 0D0D      11230          DEFW    0D0DH
BF18 49        11240          DEFM    'INSERT CP/M SYSTEM DISK IN DRIVE 0'
     4E 53 45 52 54 20 43 50 2F 4D 20 53 59 53 54 45
     4D 20 44 49 53 4B 20 49 4E 20 44 52 49 56 45 20
     30
BF3A 0D        11250          DEFB    0DH
BF3B 54        11260          DEFM    'THEN SWITCH MAPPING TO CP/M MODE.'
     48 45 4E 20 53 57 49 54 43 48 20 4D 41 50 50 49
     4E 47 20 54 4F 20 43 50 2F 4D 20 4D 4F 44 45 2E

BF5C 00        11270          DEFB    0
BF5D 0D0A      11280 MESS3    DEFW    0A0DH
BF5F 44        11290          DEFM    'DISK OFFLINE'
     49 53 4B 20 4F 46 46 4C 49 4E 45
BF6B 00        11300          DEFB    0
BF6C 0D0A      11310 MESS4    DEFW    0A0DH
```

```
8F6E 44         11320           DEFM    'DATA TERMINAL MODE - PRESS CTRL-E TO EXIT'
     41 54 41 20 54 45 52 4D 49 4E 41 4C 20 4D 4F 44
     45 20 2D 20 50 52 45 53 53 20 43 54 52 4C 2D 45
     20 54 4F 20 45 58 49 54
8F97 0D0A       11330           DEFW    0A0DH
8F99 00         11340           DEFB    0
8F9A 0D0A       11350 MESSS     DEFW    0A0DH
8F9C 42         11360           DEFM    'BIOS CORRUPTED - RESTART SYSTEM'
     49 4F 53 20 43 4F 52 52 55 50 54 45 44 20 20 20
     52 45 53 54 41 52 54 20 53 59 53 54 45 4D
   3 0D0A       11370           DEFW    0A0DH
  B0 00         11380           DEFB    0
853A            11390           END     START   ;END OF BIOS
00000 TOTAL ERRORS

ALLOC   8703   AVA1    8E3D   AVA2    8E5C   AVA3    8E7B   BACKSP  8A8C
BDOS    A706   BELL    8A97   BELL1   8A9A   BELL2   8A9C   BELL3   8AA4
BLINIT  8889   BLKSIZ  0800   BOOT    8568   CARRET  8A70   CDISK   0004
CHAR    8C6F   CHECK   8601   CHECKL  8613   CHKUNA  8792   CHRTAB  88DE
CKEND   8C38   CKSUM   8C78   CLEAR   8A5A   CONIDL  8652   CONIN   8655
CONOUT  8691   CONRDY  864F   CONST   8634   CONT    8A40   CONTAB  86CE
CONTRL  8847   CPMB    9F00   CPMSPB  0010   CPMSPT  8C73   CURDCB  8C79
CURDEL  8A78   CURPOS  8C60   CURSET  8A16   DATREG  F7EF   DCB1    8C39
DCB2    8C49   DCB3    8C59   DCS1    8E00   DCS2    8E10   DCS3    8E20
DCSREG  F7EC   DELAY   8903   DENPT   8C77   DENTAB  8C82   DIRBUF  8D85
DISKRW  8882   DISP    89F7   DMAADR  8C69   DOS     402D   DOUBLE  8960
DPB1    8C1A   DPB2    8C29   DRVCOD  8C68   DRVREG  F7E0   DSKCOM  8C74
DSKRDY  8994   DSP     0033   DSPCHR  8A14   ERCODE  8C76   ERFLAG  0040
ERMASK  8C72   EXCONT  8867   FILHST  8823   FOUND   8829   GOCPM   85C2
HOME    8701   HOME1   8707   HOMEW   870F   HSTACT  0047   HSTBLK  0002
   BUF  8C85   HSTDSK  0043   HSTSEC  0045   HSTSIZ  0100   HSTTRK  0044
    RT  0048   IOBYTE  0003   KBD     002B   KDEL    8880   KDELAY  887F
KEY     8AC4   KEY1    8AEF   KEYACT  8C7E   KEYBUF  8E05   KEYF    8670
KEYTAB  888E   KLOOP   8B83   LAST1   8877   LASTRW  8871   LINEFD  8A82
LIST    86A5   LISTST  89E1   LOAD1   858D   LOGSEC  8C7C   LOOP    8906
LSTCOD  8C6C   LSTKEY  8C7F   LSTSEC  8C7D   MAPTST  8547   MATCH   8847
MESS1   8E9A   MESS2   8ED9   MESS3   8F5D   MESS4   8F6C   MESS5   8F9A
MOTON   8986   NDISKS  0003   NOCHNG  8806   NOMTCH  881C   NON1    885F
NONLET  884C   NOOVF   87C8   NSECTS  002C   NU      8539   NXTCOL  8820
NXTROW  8AF7   ORIGIN  8500   PRESS   880E   PRINT   8688   PRTCHR  86CC
PRTLIN  86CA   PRTREG  F7E8   PRTSCR  86C7   PUNCH   8600   RDHST   8892
READ    8761   READER  86E0   READOP  004F   READY   89C8   REPEAT  8ACD
RETRY   88CA   RPTLAG  8C80   RPTRAT  8C81   RS232I  86ED   RS232O  86E3
RSBAUD  00E9   RSCTRL  00E8   RSDATA  00EB   RSFLAG  004E   RSINIT  86F6
RSSTAT  00EA   RWCOMP  8930   RWERR   896E   RWMOVE  886F   RWOPER  870C
SCROLL  8A21   SDELAY  890E   SDISP   8A89   SDSP    8A80   SECMSK  0001
    EG  F7EE   SECTN1  88E2   SECTN2  88F6   SECTRN  89EC   SEEK    8735
SEEKW   8749   SEKDSK  0040   SEKHST  0046   SEKSEC  0042   SEKTRK  0041
SEL1    8998   SEL2    899E   SELDSK  8717   SETDEN  8966   SETDMA  875C
SETSEC  8757   SETTRK  8730   SHFLOC  8C70   SINGLE  8958   START   853A
TERM1   88A1   TERMIN  889B   TERMO   88AF   TRKREG  F7ED   TRNFER  8908
TRYCNT  8C75   UNACNT  0049   UNADSK  004A   UNASEC  004C   UNATRK  004B
VERCHK  8625   WAIT1   88FA   WAIT2   8905   WAIT3   890E   WAITK   8664
WBOOT   8573   WBOOTE  8503   WRALL   0000   WRDIR   0001   WRITE   876E
WRTHST  88A2   WRTYPE  8C71   WRUAL   0002
```

```
                00100 ; * * *  CPMTEST1/ASM  * * *
                00110 ;
                00120 ; PROGRAM TO TEST CP/M MEMORY CONVERSION.
                00130 ;
                00140 ; WHILE THIS PROGRAM IS RUNNING, REPEATEDLY OPERATE
                00150 ; THE TRS/CPM SWITCH.  WHILE IN TRS MODE 'T' WILL
                00160 ; FLASH ON THE SCREEN.  WHILE IN CP/M MODE 'C' WILL FLASH
                00170 ; ON THE SCREEN.  THE PROGRAM SHOULD NOT CRASH DUE TO
                00180 ; THE SWITCH OPERATION.
                00190 ;
8000            00200           ORG     8000H
8000 F3         00210 START     DI
8001 3E20       00220           LD      A,' '           ; TURN 'C' & 'T' OFF
8003 32003C     00230           LD      (3C00H),A       ; CLEAR POS (0,0)
8006 3208FC     00240           LD      (0FC08H),A      ; & POS (8,0) OF SCREEN.
8009 10FE       00250           DJNZ    $               ; SHORT DELAY
800B 3E54       00260           LD      A,'T'           ; FLASH 'T' AT POS (0,0)
800D 32003C     00270           LD      (3C00H),A       ; OF TRS-80 SCREEN.
8010 3E43       00280           LD      A,'C'           ; FLASH 'C' AT POS (0,8)
8012 3208FC     00290           LD      (0FC08H),A      ; OF CP/M SCREEN.
8015 10FE       00300           DJNZ    $               ; DELAY AGAIN.
8017 18E7       00310           JR      START           ; ENDLESS LOOP.
8000            00320           END     START
```

```
00000 TOTAL ERRORS

START   8000
```

```
                00100 ; * * *  CPMTEST2/ASM  * * *
                00110 ;
                00120 ; PROGRAM TO TEST FOR DYNAMIC RAM LOSS, DURING
                00130 ; OPERATION OF THE RESET SWITCH.
                00140 ;
                00150 ; START PROGRAM, THEN SET SWITCH TO CP/M MODE.
                00160 ; OPERATE RESET SWITCH. A CHECKSUM OF ALL DYNAMIC
                00170 ; RAM WILL BE DISPLAYED.  REPEAT SEVERAL TIMES
                00180 ; HOLDING SWITCH IN FOR VARIOUS LENGTHS OF TIME.
                00190 ; IF CHECKSUM CHANGES, RESET PULSE IS TOO LONG,
                00200 ; AND REFRESH IS STOPPED FOR TOO LONG.
                00210 ;
C000            00220           ORG     0C000H          ; ADDR 0000H IN CP/M
C000 C30080     00230           JP      CHKSUM
8000            00240           ORG     8000H
8000 F3         00250 CHKSUM    DI
8001 2100FC     00260           LD      HL,0FC00H       ; CLEAR SCREEN
8004 1101FC     00270           LD      DE,0FC01H
8007 01FF03     00280           LD      BC,1023
800A 3620       00290           LD      (HL),' '
800C ED80       00300           LDIR
800E 210000     00310 LOOP1     LD      HL,0            ; GET 16 BIT CKSUM
8011 110040     00320           LD      DE,-0C000H
8014 AF         00330           XOR     A
8015 47         00340           LD      B,A
8016 86         00350 LOOP      ADD     A,(HL)
8017 3001       00360           JR      NC,SKIP
8019 04         00370           INC     B
801A 23         00380 SKIP      INC     HL
801B 1C         00390           INC     E
801C 20F8       00400           JR      NZ,LOOP
801E 14         00410           INC     D
801F 20F5       00420           JR      NZ,LOOP
8021 6F         00430           LD      L,A             ; DISPLAY CKSUM
8022 60         00440           LD      H,B
8023 1100FC     00450           LD      DE,0FC00H
8026 0604       00460           LD      B,4
8028 AF         00470 V         XOR     A
8029 29         00480           ADD     HL,HL
802A 8F         00490           ADC     A,A
802B 29         00500           ADD     HL,HL
802C 8F         00510           ADC     A,A
802D 29         00520           ADD     HL,HL
802E 8F         00530           ADC     A,A
802F 29         00540           ADD     HL,HL
8030 8F         00550           ADC     A,A
8031 27         00560           DAA
8032 FE0A       00570           CP      10
8034 DECF       00580           SBC     A,-'0'-1
8036 12         00590           LD      (DE),A
8037 13         00600           INC     DE
8038 10EE       00610           DJNZ    V
803A 1802       00620           JR      LOOP1
8000            00630           END     CHKSUM
00000 TOTAL ERRORS

CHKSUM  8000    LOOP   8016    LOOP1   800E    SKIP    801A    V       8028
```

ZAP TO NEWDOS/80 (MODEL III VERSION) WITH VIDEO4
Provided by Bob Seaborn

The following patch to the LIB command permits the maximum number of DOS commands to be displayed on each screen line under both normal conditions (64 character display, 8 commands on each video display line) and when VIDEO4 is in use (80 character display, 10 commands displayed per line):

SYS1/SYS,3,E1 change 0E 08 06 to 0E 2F 06

TRSDOS 8.2 -- PATCH/CMD

by Dave Bower [Compuserve 70635.330]

PATCH/CMD is a convenient way to make minor changes to a disk file. It changes a byte,. or bytes. from the original values to new values you specify. It can verify that you are changing the proper bytes, if you like.

A patch can be installed two ways, typed in from DOS Ready or by a patch file. Each of these can be in two forms, the "memory load" method, or the "direct modify" method. Each of these two methods can be expressed in two variations, HEX and ASCII. And patches can be removed with two different parameters, YANK and REMOVE.

This may sound a little confusing, but PATCH/CMD is a versatile, easy to use, convenient and very handy little program, for the beginning and seasoned hacker alike.

But it can be a little intimidating at first, and the documentation leaves a little bit to be desire. Maybe I can shed a little bit of light on the subject. We'll take it two steps at a time.

First I want to talk about the two different patch methods, the "memory load" and the "direct modify" method. The difference between the two is WHEN and WHERE they actually patch your file. The "direct modify" method changes your file. on the disk, at the time you apply the patch. When you use this method PATCH changes the bytes to your new values. If you would then disassemble FROM DISK you would see the changed code. This method can also be used on ANY disk file, not just /CMD files.

The "memory load" does not patch your file until you load it into memory. PATCH adds the patch code TO THE END of your file with a loader that changes your program IN MEMORY every time the program is loaded. If you were to disassemble from DISK you would see the old code, but if you disassembled from MEMORY you would see the new code. This type of patch can only be applied to files that can be run directly from DOS (i.e. /CMD files).

The "memory load" patch does not give you the option of specifying a FIND byte to verify that you are in the right area. the "direct modify" patch does.

MEMORY LOAD PATCH

Let's look at how the two different methods are applied from DOS. A "memory load" command looks like this:

```
PATCH FORMAT/CMD.UTILITY (X'3A06'=54 72 73 64 6F 73 36 32)
```

Or, the same patch might look like this:

```
PATCH FORMAT/CMD.UTILITY (X'3A06'="Trsdos62")
```

The part of the command we are talking about is in parentheses. When used directly from DOS it must be. There are two parts. The X'nnnn' is a four digit HEX address which tells PATCH/CMD that this is a "memory load" patch, and where to put it. The information after the equals sign is the data to be loaded at the specified memory address. It can be in the form of HEX bytes, nn nn nn, or it can be an ASCII string enclosed in quotes.

One interesting note about the "memory load" format. With this method you can patch areas past the end of your program. If you specify a load address (X'nnnn') one byte greater than the address of the last byte in your program, patch will add the code to the end of your program. If you can't find room within your program you could do a CALL to a patch tacked to the end. (My compliments to Northern Bytes for this one).

The two examples are legitimate patches and either one will change the default disk name in TRSDOS 6.2, FORMAT/CMD, version 6.2.0 to Trsdos62. (Although it will display upper case unless you've applied the lower case patch to FORMAT/CMD which I'll give later.)

To recap, this is a "memory load" patch. It adds the patch info to the end of your program and loads the patch every time the program is loaded into memory. The program you are patching must be a DOS executable file.

Be careful when applying this patch. It will patch the address you specify, no matter what is there. Make sure you're not patching a later or different version of a program with different addresses!

DIRECT MODIFY PATCH

A "direct modify" patch looks like this:

```
PATCH FORMAT (D11,EA=20 20 20 20:F11,EA=44 49 53 4B)
```

or

```
PATCH FORMAT (D11,EA="    ":F11,EA="DISK")
```

Each of these commands does exactly the same thing. If applied to FORMAT/CMD, TRSDOS 6.2 each will change the letters DISK in the default disk name (DATADISK) to four spaces (DATA   ). Patch will first verify that the old bytes are what you expected before applying the patch.

There are six parts to this type of patch.

Dxx,	Record # within file to directly modify
xx	Relative byte in the record to start
=DATA:	NEW data in HEX or ASCII form
Fxx,	Record # within file to verify (same as Dxx)
xx	Relative byte to start (same as Dxx.xx)
=DATA	Old data, if not found PATCH aborts

When used directly from DOS more than one command can be on the line, but each should be separated by a colon. For example,

```
PATCH FILESPEC (D00,1F=00:F00,1F=1A:D0A,1C=1A:F0A,1C=BB)
```

modifies two different records, 0A and 00.

To recap the "direct modify" patch, it changes the specified byte or bytes within the file ON DISK. A disassembly from disk would show the new code. This method allows the user to specify what the original data should be before applying the patch. And, you can patch any disk file with this method.

PATCH FILES

A better method than applying each individual patch fr DOS would be to put your commands in a patch file and ha PATCH/CMD install, YANK or REMOVE them.

A JCL file is recommended for applying patches, especially with long or numerous patches. You can double check your work and it can be reused.

A patch file is an ASCII list of patch commands. For example, a "direct modify" patch file called FORMAT1/FIX might look like this:

```
. This patch for FORMAT/CMD.UTILITY ver. 6.2.0
. TRSDOS 6.2
. 9/3/85  by Dave Bower
.
. It allows the use of lower case in a, diskette name
. when formatting.
.
. Apply using the command
.         ==> PATCH FORMAT.UTILITY FORMAT1
.
D09,E1=00 00
F09,E1=EE 20
D0D,B4=7B
F0D,B4=5B
.
```

Notes:

1) Documented; PATCH/CMD ignores lines starting with a perioc (Special characters in a comment line, <u>including semicolons</u>, can cause a patch file to fail!)
2) Includes example of command to invoke the patch, lest I should forget.
3) The /CMD is assumed in Filespec 1
4) The /FIX is assumed in Filespec 2
5) Each command should be on a separate line

I apply this patch by issuing the following command from DOS Ready:

```
PATCH FORMAT/CMD.UTILITY FORMAT1/FIX
```

By the way, this is a legitimate patch should you want to use lower case letters in your disk names.

Here's an example, using the "memory load" method that I call FMTNAME/FIX.

```
. Patch to FORMAT/CMD ver 6.2.0, TRSDOS 6.2 to change
. the default name to Trsdos62.
.
. Apply using the command
.
.         ==> PATCH FORMAT/CMD.UTILITY FORMAT1/FIX
.
X'3A06'="Trsdos62
```

I would apply this patch using the command,

PATCH FORMAT/CMD.UTILITY FMTNAME/FIX

REMOVING PATCHES FROM FILES

What if you want to remove a patch from a file? Use the YANK or REMOVE parameter after the identical command you used to install the patch. If you created a patch file it's a piece of cake.

If you used the "memory load" method (X'nnnn'=nn nn nn) then use the YANK parameter.

PATCH FORMAT/CMD.UTILITY FMTNAME/FIX (YANK)

If you used the "direct modify" method (D0A,E1=00) then you use the REMOVE parameter.

PATCH FORMAT/CMD.UTILITY FORMAT1/FIX (REMOVE)

Filespec 2 in each case was the same file you used when you installed the patch.

Using patch files to install patches of either method is much quicker than applying each individual patch via DOS, and safer! And they can be removed with one simple command.

Here is my patch file to modify FORMAT/CMD (version 6.2.0) from TRSDOS 6.2. Two things to notice. When mixing methods ("memory load" and "direct modify") you should put the "direct modify" commands first. And you must YANK and REMOVE the file to remove both kinds of patches, even though PATCH/CMD claims that all patches are removed each time. REMOVE only removes "direct modify" patches and YANK only yanks "memory load" patches. You would need to issue the patch command twice.

PATCH FORMAT/CMD.UTILITY DEFAULTS/FIX (YANK)
PATCH FORMAT/CMD.UTILITY DEFAULTS/FIX (REMOVE)

These are working patches and you can use as many or as as you like to customize your FORMAT/CMD.

```
.       Patches to FORMAT/CMD.UTILITY (TRSDOS 6.2)
.
.       Apply ===>  PATCH FORMAT.UTILITY DEFAULTS/FIX
.
.       Verify that we've got right version
.
D0E,96="Format - 6.2.0"
F0E,96="FORMAT - 6.2.0"
.
.       Change default name to Trsdos62
.
D11,E6="Trsdos62"
F11,E6="DATADISK"
.
.       Allows lower case for disk name.
.
D09,E1=00 00
F09,E1=EE 20
D0D,B4=7B
F0D,B4=5B
.
.       No NAME prompt (use parameter or default)
.
D09,EF=00 E6 00 00
   F=EF CD 65 2A
.
.       No PASSWORD prompt (use parameter or default)
.
D0A,38=00 00 00
F0A,38=CD 95 35
.
.       No DENSITY prompt (use parameter or default)
.
D0A,99=E6 00 00
F0A,99=CD 5D 2A
.
.       No CYLINDERS prompt (use parameter or default)
.
D0B,0F=00 00 00
F0B,0F=CD 5D 2A
.
.       No SIDES prompt (use parameter or default)
.
D0A,E4=E6 00 00
F0A,E4=CD 5D 2A
.
.       Defaults to SIDES=2
.
D0A,E8=12
F0A,E8=16
.
.       Does not check for data when ABS specified.
.       from Northern Bytes
.
X'3492'=CD 16 3A
X'3A16'=3A 27 35 B7 C0 CD 27 2A C9
.
.       9/5/85
.
```

That about wraps it up, except for the Lxx parameter. That deals with patching library commands and I am going to ignore it. You'll see that in patches supplied by Radio Shack and they will tell you where to stick it.

One last little point. You can determine addresses for your own patches by disassembling a file. You can determine load blocks and relative bytes by using the LIST command from DOS Ready with the HEX parameter. The LSI file editor, FED II, is an excellent tool for discerning addresses, blocks, bytes and such for patches.

NORTHERN BYTES INDEX FOR VOLUME 6

Notes on this index: 1) If an asterisk appears next to a name in the author's name field, it means that the person mentioned did not actually write the article, but did provide the pertinent information. 2) The issue and page number are given in the format 3/17 (in this example, the article would be found in NORTHERN BYTES Volume 6, Number 3, page 17). If the symbol "/" appears, it means that a correction or follow-up article appears in THIS ISSUE (we could not give the page number since the layout for this issue had not been completed at the time this index was printed). If a large "5/" appears in front of the issue and page number (i.e. 5/3/17) this means that the referenced article was in Volume 5 rather than Volume 6 (such articles are indexed only when a correction or follow-up article appears in Volume 6). 3) We have tried to index all corrections, follow-up articles, and "letters to the editor" dealing with a given article immediately following the listing for the original article. 4) SOME, but not all, "letters to the editor" have been indexed. The indexed letters usually contained useful information of some sort.

GRIGONIS, ED COMMENTS ON THE PREVIOUS LETTER
On Chuck Tesler's letter to the Adelaide users group 8/7
LOGAN, LES LETTER TO THE EDITOR
Comparison of TRS-80 DOSes, problems with DOSPLUS 6/1
NOONAN, ROBERT F. LETTER TO THE EDITOR
Comments on widespread software piracy in schools 6/3
SOLTOFF, ROY LETTER TO THE EDITOR
Comments on the LDOS/TRSDOS 6 "go to sleep" mode 8/3
TESLER, CHUCK PROSOFT ON PIRACY
Reprint of letter to Adelaide, Australia users group 8/7
COMMENTS ON THE PREVIOUS LETTER 8/7
NORTHERN BYTES EDITOR'S COMMENTS ON THE FOREGOING 8/8
THE EXTERMINATOR */*

FEATURE ARTICLES

BROTHERS, HARDIN LETTER TO THE EDITOR
Comments on TRSDOS 6.2, use of SET, ROUTE, LINK, FILTER 4/2
BRYCE, GARY TWO JCL FILES FOR LDOS USERS
Mirror Image Format & Backup and Dos Disk Creator 7/20
BUXTON, JOHN AN EXPLANATION ABOUT ELECTRICITY
Not exactly what you learned in Physical Science class 8/8
"COMPUTER NUT" OVERSEAS EXPERIENCE
Series on problems in using U.S. made computers overseas
Part 1 - 5/8/4
Part 2 - 1/11
Part 3 - 3/4
Part 4 - 4/19
DADASHZADEH, M. THE PROTECTION SCHEME OF COPYCAT 3
What it is and how to defeat it 2/9
DECKER, JACK ANOTHER FREE SERVICE BITES THE DUST
The demise of ONLINE TELECOMPUTING's free service 1/9
DECKER, JACK CHRISTIAN COMPUTER USERS ASSOCIATION
Information on this organization 1/3
DECKER, JACK CLUBS AND NEWSLETTERS
Clubs and user groups then on our mailing list 1/18
DECKER, JACK CUT THOSE HIGH TELEPHONE BILLS
Information on GTE Telenet's PC PURSUIT service 6/9
TELECOMMUNICATIONS NEWS 7/4
DECKER, JACK NORTHERN BYTES ARTICLE SUBMISSION FORMAT
The closest thing we've had to an "author's guide" 3/9
DECKER, JACK MCI MAIL CHANGES
Price increases and how to access from outside the U.S.A. 1/8
DECKER, JACK MODEL 4 ROM & MODEL 4P ROM IMAGE CHANGES
Documents the changes in the new Model 4/4P ROM (image) 2/6
THE EXTERMINATOR 3/2
DECKER, JACK MODEL III MOTHERBOARDS AND OTHER SURPLUS BARGAINS
If you missed this back then, it's probably too late now! 1/4
DECKER, JACK NORTHERN BYTES INDEX FOR VOLUME 5
Last year's edition of this compilation 1/19
DECKER, JACK UNDERUSED BASIC PROGRAMMING STATEMENTS
New ways to use certain BASIC language statements 7/15
DECKER, JACK TELECOMMUNICATIONS NEWS (2 COLUMNS)
Observations and information of interest to MODEM users 7/4
More of interest to avid telecommunicators 8/18
THE EXTERMINATOR */*
DECKER, JACK TRS-80 ROM ROUTINES DOCUMENTED BUG
Errors in the book corrected! 5/4/3
THE EXTERMINATOR 5/2/1
ADDING YOUR OWN /SYS OVERLAYS TO NEWDOS/80 5/3/3
THE EXTERMINATOR 5/4/2
THE EXTERMINATOR 5/7/1
MODEL 4 ROM & MODEL 4P ROM IMAGE CHANGES 2/6
THE EXTERMINATOR 3/2
DECKER, JACK WHICH LONG DISTANCE CARRIER ARE YOU CONNECTED TO?
Maybe not the one you think. Here's how to tell 5/14
ECKER, MICHAEL W. MAGIC MATH PLUS (SERIES)
Program to draw lines on video display in any direction 1/10
Superblackjack: The Game of 110 2/12
The Triangle Number Trick 4/11
GOOD NEWS B.B.S. CHRISTIAN BULLETIN BOARD SYSTEMS
List downloaded from Good News BBS (313) 459-8375 5/22
GRUENTHER, DON TRANSFERRING A BASIC PROGRAM FROM MOD I TO MS-DOS
The fun(?) of converting a program to MS-DOS BASIC 8/21
HESTER, VERN * MULTIDOS 1.7 CHANGES
Discussion of differences between versions 1.6 and 1.7 4/20
MICHNA, HANS G. PACKET SWITCHING NETWORKS
Little-known info on international packet networks 3/17

TUSCON PC USERS GRP FREE DATA BASE DIRECTORY
Available from GTE Telenet 1/17

HARDWARE

BALAS, RICH EPROM PROGRAMMER
Schematic diagram and software, for Model I 7/11
BROERS/JANSEN/LAMPE KARAKTERSET (CHARACTER SET) IN EPROM
Customize the video character set on your Model I 5/16
BRYCE, GARY AUDIO AMPLIFIER IN YOUR VIDEO
Add high-quality sound to your video display unit. 2/14
DECKER, JACK AN ELECTRONICS INVASION FROM 'DOWNUNDA'
Dick Smith Electronics now in the U.S.A. 6/22
DECKER, JACK LETTER TO THE EDITOR (REPLY)
Hints to help make a CPU speedup mod work in a Model I 5/2
DECKER, JACK MODEL I LEVEL II LOWERCASE ON POWERUP
Replace Model I ROM with EPROM with lowercase driver 8/15
THE EXTERMINATOR */*
LETTER TO THE EDITOR */*
DECKER, JACK WARNING ON ADDING 64K MEMORY IC'S TO THE MODEL 4/4P
Use the ones with proper the refresh method 5/10
KELLER, S./TINNEY, C. REAL TIME CLOCK/CALENDAR
Give your TRS-80 its' own clock, with battery backup 5/9
KENNEDY, DAVE A REAL TIME CLOCK
Improved design with battery backup and software 8/11
KENNEDY, DAVE EXTRA RAM FOR THE MODEL I
Used with Errol Rosser's upgrade, enables 3000H area
KENNEDY, DAVE RANDOM I/O
This circuit doesn't work properly; don't use it! 1/7
THE EXTERMINATOR 3/2
USING 256 CYCLE REFRESH RAMS
(This is the corrected version of this circuit) 7/12
MacAULAY, EEON P BUFF KIT REVIEW
Details of Don McKenzie's print buffer kit 8/20
MORE ON THE PBUFF PRINTER BUFFER KIT */*
MATUG NEWSLETTER EPROM PROGRAMMER KIT
Info on a kit from a Wisconsin firm 7/3
McKENZIE, DON * ATTENTION HARDWARE HACKERS
A 1200 baud MODEM kit, & Don's PBUFF print buffer kit 4/12
P BUFF KIT REVIEW 8/20
MORE ON THE PBUFF PRINTER BUFFER KIT */*
OMER, GUY * ANCHOR SIGNALMAN MARK XII DTR FIX
Add the DTR signal required by most BBS software 2,
ORANGE BYTES N/L TANDON DRIVE DOOR HINGE REPLACEMENTS
A source for a better hinge at a lower price 6/22
OWEN, DAVE FAST FIX FOR MODEL 4 SLOWS
How to fix hardware bug that slows pre-1984 Model 4's 6/15
PETERS, DAVE FOUR DRIVES FOR THE MODEL 4P
Conversion instructions to permit use of four drives 7/7
PHILLIPP, JOHN FIX YOUR BLOOMING SCREEN
Replace unstable Mod 4/4P trim pot R-15 with resistors 8/17
ROSSER, ERROL * EXPANSION UNIT FOR PMC-80/SYSTEM-80/VIDEO GENIE USERS
Has single density disk controller, printer interface
ROSSER, ERROL UPGRADING THE TRS-80 MODEL I TO 64K OF DRAMs
Internal modification puts 64K inside keyboard unit 5/8/18
RANDOM I/O
Circuit by another author; doesn't work properly!! 1/7
THE EXTERMINATOR 3/2
LETTER TO THE EDITOR 5/2
EXTRA RAM FOR THE MODEL I 5/6
USING 256 CYCLE REFRESH RAMS 7/12
ROSSER, ERROL USING 256 CYCLE REFRESH RAMS
Circuit to generate 256 cycle RAM refresh on Model I 7/12
SANTANA, MIKE INSTRUCTIONS FOR THE INSTALLATION OF ADDITIONAL 64K IN THE NEW TRS-80 MODEL 4A
For new Model 4's with single P.C. Board 3/10
SIMON, MIKE L. * EL CHEAPO EPROM ERASER
Information extracted from early article by Mr. Simon 7/3

REVIEWS

DECKER, JACK THE COMPLETE HANDBOOK OF PERSONAL COMPUTER COMMUNICATIONS
Review of Alfred Glossbrenner's newly revised edition 8/10
DECKER, JACK THE EXPLOITED MEDIA, AND HOW NOT TO BE ONE OF THEM
Review of "THE UNABASHED SELF-PROMOTER'S GUIDE" 4/1
SALSBURY, NATE LETTER TO THE EDITOR
Mini-review of Soltoff's "Programmer's Guide to TRSDOS 6" 2/4

SOFTWARE

- BESLER, DARREN R. DATECOMP/ASM
  - An assembly language date compression routine 6/14
- BROTHERTON, MICHAEL WHERE
  - Shows exact location of error within BASIC line 5/7/6
    - THE EXTERMINATOR 3/2
- CALMETTES, SERGE Y. SCRPRT FAST SCREEN PRINT PROGRAM
  - Use with Models III/4 and certain Radio Shack printers 4/15
- ADASHZADEH/BYRNE SAVESUB: THE BASIC VERSION NUMBERING UTILITY
  - Never accidentally overwrite the wrong program again 8/9
- DECKER, JACK MODEL 4 SETDATE UPGRADE
  - Upgrades Model 4 SETDATE from version 6.0 to 6.1 8/6
- DECKER, JACK SETDATE
  - The easy way to enter the date at system startup 5/2/7
    - THE EXTERMINATOR 5/4/1
    - THE EXTERMINATOR 5/5/2
    - LETTER TO THE EDITOR 5/6/4
    - LETTER TO THE EDITOR 5/7/2
    - THE EXTERMINATOR 1/2
    - LETTER TO THE EDITOR 2/4
    - MORE CHANGES TO SETDATE 3/10
    - (See also next entry in this index)
- DECKER, JACK SETDATE - THE MODEL 4 VERSION
  - Version 6.0 of the popular date-setting utility 6/18
    - MODEL 4 SETDATE UPGRADE 8/6
- DOMIGAN, TONY FIXGAT/ASM FOR MODEL III NEWDOS/80 VERSION 2
  - Corrects Granule Allocation Table of NEWDOS/80 disks 3/14
- MIGAN, TONY FIXHIT/ASM
  - Repairs damaged NEWDOS/80 (Model 3) Hash Code Tables 2/19
- DOMIGAN, TONY ROM/ASM
  - Restores old ROM image if it's been changed on Model 4P 5/4
- FORD, CARL JR. SCREEN BLANKING ROUTINE FOR USE WITH NEWDOS/80
  - Blanks the video display after a few minutes of non-use 1/14
- GROMMES / SOLTOFF FKEY/CMD
  - Model 4 utility lets you redefine function keys at will 2/10
- HOLDEN, PHIL GHOST MENU
  - Program-selection menu executes from video memory 5/15
- KEIJZER, RICHARD A. MAAK EENS EEN DOOLHOF (MAKE A MAZE)
  - Programs that produce mazes on your printer 5/19
- KELTERBAUM, JOACHIM A NEW OVERLAY MODULE FOR NEWDOS/80 VERSION 2
  - JKL with TRS-80 graphics for Epson RX80/FX80 5/4/6
    - RANDOM I/O 1/10
- KOEHLER, BOB FINANCIAL PLANNER
  - Determine income potential of an investment 5/13
- LIVINGSTONE, JOHN T.* EASTER
  - Find date of Easter in given year 8/8
- MUMAUGH, JIM BASIC WORD SEARCH PROGRAM
  - An excellent program to create word search puzzles 4/17
- SALSBURY, NATE TRS-80 GRAPHICS ON THE FX-80 PRINTER
  - Program to download TRS-80 block graphics to FX-80 1/11
- SMALL, GREG TRACE/CMD
  - Machine language debugging aid, use with NEWDOS/80 2/18
- NCER, GIL INPUT@ DISK BASIC ENHANCEMENT
  - Add new statements to NEWDOS/80 Disk BASIC 3/19
- VAN DAM, JOOP PROMPT
  - Change the BASIC "READY" prompt to whatever you wish 1/18
- ZAJAC, RON KEYBOARD PROCESS ROUTINE FOR USE FROM BASIC with SAMPLE PROGRAM
  - Machine language routine & a 2-player entrapment game 7/9

TIPS AND TECHNIQUES

- ABRAHAMSON, ALAN ASK ALAN
  - Convert text to random access files; "help" file demo 1/16
- ALLEN, BILL SQUEEZING AN EXTRA GRAN INTO A DATA DISK INTO AN EMERGENCY
  - You can KILL BOOT/SYS from a DATA disk if necessary 7/21
- BAKER, BILL GOOD NEWS DEPARTMENT
  - News of new Cancel Call Waiting telephone feature 5/8
    - TELECOMMUNICATIONS NEWS 8/18
- BRYCE, GARY CREATE A SELF-BOOTING DISK USING LDOS
  - Gary says it's much easier using LDOS than NEWDOS/80 4/22
- DECKER, JACK ALLWRITE! BUG(?)
  - Don't omit carriage returns after control word lines! 3/16
- DECKER, JACK ALLWRITE HINTS
  - Make Allwrite's text editor a little easier to use 5/22
- DECKER, JACK FASTER STRING SORTS/SWAPS IN MODEL I/III BASIC
  - Avoid "garbage collection" when swapping BASIC strings 7/14
- DECKER, JACK MICROSOFT BASIC AND FLOATING-POINT NUMBERS
  - Why number-crunching may yield unexpected results 6/17
    - LETTER TO THE EDITOR 8/4
- DECKER, JACK MODEL I/III SELECTIVE RESTORE
  - Method to RESTORE to other than first DATA statement 6/22
- DOMIGAN, TONY LETTER TO THE EDITOR
  - NEWDOS/80 routines convert bytes to ASCII equivalents 2/2
- DOMIGAN, TONY MODEL III NEWDOS/80 ROUTINES
  - A few useful NEWDOS/80 version 2 subroutines 4/10
- EVANS, ROWAN MODEL 4 SYSTEM STATUS FLAGS
  - What the flags control; accessing them from BASIC 6/7
- FREIFELD, MICHAEL R. MODEL I/III TO MODEL 4 SUPERSCRIPSIT PRINTER DRIVERS
  - Convert Model I/III SuperScripsit drivers to Model 4 5/21
- GAZDAR, JEHANGIR * BASIC PROGRAM FUNCTIONS YOU CAN USE
  - Define BASIC functions to ease string handling 1/13
- GROMMES, BOB LETTER TO THE EDITOR
  - How to get a version of TRSDOS 6 for the Models II/12 2/4
- GROMMES, BOB SET/RESET AND DIRECT VIDEO CONTROL FOR MODEL 4 BASIC
  - Regain control lost when you "moved up" to Mod 4 BASIC 3/5
- GROMMES, BOB * USE OF THE SPOOLER ON TRSDOS 6.2.0
  - Problems in using the spooler and how to fix them 3/4
- HAGERS, A. J. WHERE AM I?
  - Simulate relative CALLs using ROM code at 000BH-000CH 4/3
- HESTER, VERN MULTIDOS MODEL 4 BASIC ENTRY POINTS
  - List of equivalents to Model I/III ROM entry points 5/22
- JAEGER, PAUL LETTER TO THE EDITOR
  - Floating point errors in Microsoft Fortran, Alcor Pascal 8/4
- KELTERBAUM, JOACHIM CREATE A SELF-BOOTING DISKETTE USING NEWDOS/80
  - Make your own self-booting disks 5/4/11
    - THE EXTERMINATOR 5/6/2
    - MODEL III SELF-BOOTING DISK USING NEWDOS/80 1/3
      - THE EXTERMINATOR 3/2
- LEWIS, DARREL DOUBLE PRECISION IN 5 BYTES
  - Store money amounts to $49 million in 5 bytes on disk 7/6
- LSI CUSTOMER SERVICE LDOS, TRSDOS 6 AND 1987 - THE WHOLE STORY
  - Why your disk file dates will be messed up in 1988 6/16
- MARTEN, FRANK A DOUBLE SIDED BOOTING DISK UNDER LDOS
  - Create a double sided LDOS system disk 6/12
- ORR, MIKE INTERFACING BASIC AND TRSDOS 6 SVC ROUTINES
  - How to call TRSDOS 6 SVC routines from within BASIC 3/22
- PHILLIPP, JOHN T. INSTALL TURBO PASCAL ON THE TRS-80 MODEL 4
  - Under Montezuma Micro CP/M 2.2 version 1.4 2/22
- PRICE, TOM MEMDISK AND MODEL 4 SUPERSCRIPSIT
  - How to run Model 4 Superscripsit from the MEMDISK 1/9
- ROHDE, ARNE LETTER TO THE EDITOR
  - How to implement a wildcard search in a monitor program 3/3
- SANDS, WARWICK EXTRA INSTRUCTIONS FOR EDAS 3.5
  - Undocumented EDAS commands used with "alien" files 5/4
- SANDS, WARWICK INTERRUPTS IN THE TRS-80
  - Beware if enabling interrupts in Level II environment 1/17
- SEABORN, BOB * HINTS ON READING TRS-80 COLOR COMPUTER DISKETTES UNDER NEWDOS/80
  - How to read or copy a CoCo disk on the Model I/III/4 6/11
- SHIELDS, LAURIE ACCESSING THE VIDEO ON THE MODEL 3 AND 4
  - Tricky techniques for video display manipulation 1/5
- SHIELDS, LAURIE * NEWDOS/80 TIPS FROM LAURIE SHIELDS
  - Problem with CMD"S=..." and JCL file /./5 format info 1/18
- SMALL, GREG LETTER TO THE EDITOR (REPLY)
  - SuperScripsit EOF error fix, and NEWDOS/80 tables 1/2
- SMALL, GREG * NEWDOS/80 BUG
  - RENAME bug if new filename starts with characters "TO" 2/22
    - LETTER TO THE EDITOR (HUMOROUS) 6/5
- SMALL, GREG (INTERNATIONAL) NEWDOS/80 USERS' GROUP (SERIES)
  - Commented disassembly of BOOT/SYS 2/15
  - ZAP creation, BASIC zap permits more than 15 open files 3/11
  - A cornucopia of useful zaps to NEWDOS/80 5/7
- STEVENSON, ROD CONFIGURATION IN LDOS
  - Tips on installing configurations 8/22
- TATSIG NEWSLETTER * WARNING TO 2400 BAUD MODEM BUYERS
  - The Hayes Smartmodem 2400 is not "Hayes Compatible"?! 6/16

ZAPS AND PATCHES

NORTHERN BYTES

Subscription Information

Northern Bytes is edited by Jack Decker and published on an irregular basis by The Alternate Source Information Outlet. Back issues are available starting with Volume 5, Number 1. Issues prior to that are not available. Some of the most valuable articles from earlier issues may be reprinted in future issues of Northern Bytes. Currently there are eight back issues available for Volume 5, as well as all issues from Volume 6. All back issues are $2 each.

It is very easy to be placed on the Northern Bytes **REGULAR** list. Simply place your address, Visa or MasterCard number and expiration date on file with us. We will start with the issue you request. We do not bill you for **ANY ISSUE** until that issue has been mailed. This way, we can continue to offer you top quality information with absolutely no risk to you. There's no question of what to do about unfulfilled issues if we decide to quit publishing. Unless otherwise requested, we presume your subscription will extend through the month of your expiration date.

Don't have a charge card, huh? We understand the myriad of reasons for not having them and we feel that a "To-Be-Invoiced" policy could help increase the demand for Northern Bytes. If you'll do it for us, we'll do it for you. Would you like to be placed on a regular list **TO BE BILLED** for each issue? You could then send a check for the issues as they are mailed. If you didn't send a check, we would presume that your interest has died and discontinue your subscription. The only requirement for getting onto the list is to pay for the first issue up front; the next will be mailed automatically, if you request. You are insured that you will receive top of the line TRS-80 information as it is released. Ask to be placed on the **NO RISK** "To-Be-Invoiced Northern Bytes List".

Call or write, but SIGN UP TODAY!

The Alternate Source Information Outlet
704 North Pennsylvania Avenue
Lansing, Michigan 48906-5319

Telephone (517) 482-8270, or use our toll-free lines for ORDERS ONLY:
(800) 632-7818 ext. 700 in Michigan, (800) 253-3200 ext. 700 elsewhere in U.S.A.
EMAIL: CompuServe: 72167,161 / Delphi: TASIO / MCI Mail: 109-7407 / Telex: 6501097407 MCI

NORTHERN BYTES

c/o Jack Decker
1804 West 18th Street # 155
Sault Ste. Marie, Michigan 49783-1268
U.S.A.

Bulk Mail
U.S. Postage Paid
Permit 815
Lansing, MI

POSTMASTER: Address Correction Requested.
Send address changes, USPS Form 3579, and undeliverable copies to: The Alternate Source Information Outlet, 704 North Pennsylvania Avenue, Lansing, Michigan 48906-5319

To:

ELECTRONIC MAIL ADDRESSES:

Compuserve EasyPlex:	72167,161
Delphi Mail:	TASIO
MCI Mail:	109-7407
Telex:	6501097407
Answerback:	6501097407 MCI