

NORTHERN BYTES



Volume 7 Number 1

THE EXTERMINATOR

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 THE 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 page 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 (906) 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 you don't want to give up the convenience of NEWDOS/80, but can't stand 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 possible.

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:

```
MI/III FRS 20,03 - M4 FRS 25,98: change FE 2E 20 05 to FE 20 20 05
MI/III FRS 21,86 - M4 FRS 26,CE: change FE 2E 20 03 to FE 20 20 03
MI/III FRS 23,66 - M4 FRS 28,7E: change 3E 2E 23 20 0E to 3E 20 BE 23 20 0E
MI/III FRS 28,10 - M4 FRS 33,2B: change 3E 2E BE 20 05 to 3E 20 BE 20 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 Comuserve [72167,161], Delphi [TASIO], or MCI Mail [109-7407] (especially if longer than a couple of paragraphs). If you are NOT using Allwrite (or Newsprint) and your word processor offers the option to save your file in ASCII format, please do so (especially if using SuperScripts!). 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 OFFH, 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 Malo
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 radio license preparation materials that contain all questions, answers, 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 frequencies for a new Computer Hobbyist Radio Service." It was (and 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 FCC's 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 proposal ran to 28 typewritten pages. I called Don when I received 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 their time.

Stoner is the Vice President of Engineering at the Microperipheral 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 μ v
- (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 there 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 be assured that telephone and broadcast interests will oppose the 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 such 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 via 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 killing support for amateur radio!) The FCC has a lot of input from 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 Band radio was so popular for a time. Unfortunately, the immature 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 to 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 0? [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.]

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 81c 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.

[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 (0=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 original 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>
0 <ENTER> (the letter 0, 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 if 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 Warwick 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 MATHEMATICAL RECREATIONS from MAGIC MATH PLUS
by Dr. Michael Ecker
and Recreational Mathematical 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 recreations 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 of the digits (10) from the original number (1234) to get 1224. Mentally 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 nines", but I find that even those who have heard of this process don'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:SS=STRING$(18,42)
110 CLS:PRINT@342,SS:PRINT@406,"*DIGIT
PREDICTION*";PRINT@470,SS:PRINT@960,"RECREATIONAL MATHEMATICAL
SOFTWARE (C) 1985 DR. MICHAEL ECKER"
120 FORJ=1TO1500:NEXTJ
220 CLS
230 PRINT:PRINT TAB(17)"THINK OF ANY WHOLE NUMBER.":PRINT
240 FOR Z=1 TO 1000:NEXT
250 PRINTTAB(13) "ADD UP THE DIGITS OF YOUR NUMBER."
260 PRINT:FOR Z=1 TO 2000:NEXT
270 PRINT TAB(9)"NOW SUBTRACT THE RESULT OF THE
ADDITION":PRINTTAB(16)" FROM THE ORIGINAL NUMBER."
280 PRINT:PRINT:PRINT
290 PRINTTAB(14) "PRESS <ENTER> TO CONTINUE";INPUT X$
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$,I,1))
380 NEXT I
390 IF S>=9 THEN S=S-9:GOTO 390
400 A=9-S
410 FOR Z=1 TO 700:NEXT
420 CLS
430 PRINT "CONCENTRATE NOW ON THE MISSING DIGIT..."
440 FOR Z=1 TO 1000:NEXT
450 PRINT:PRINT "AHA!... THE MISSING DIGIT IS";A;"!"
460 PRINT:PRINT:PRINT:FOR Z=1 TO 1000:NEXT
470 INPUT "PLAY AGAIN (Y OR N)"; RES$
480 IF RES$="Y" OR RES$="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 $1 \times 1 \times 1$, $5 \times 5 \times 5$, and $3 \times 3 \times 3$. 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 inputting 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 Mathematical Software in this case).

```
99 REM TRS-80 VERSION HERE
100 CLEAR500:SS=STRING$(27,42)
110 CLS:PRINT@338,SS:PRINT@402,"*THE REMARKABLE NUMBER
153*";PRINT@466,SS:PRINT@960,"RECREATIONAL MATHEMATICAL
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 fortunes, it's nice to be part of an enterprise that offers something other 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 Mathematical Software and the editor / publisher of Recreational and Educational Computing Newsletter, is an associate professor of mathematics and computer science, presently at the University of Scranton. He is a known computer 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 1 & 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 standard Dick Smith expansion unit for the System 80. (I don't know for sure - I use an LNW expansion board.) See note below.

COMPONENTS:

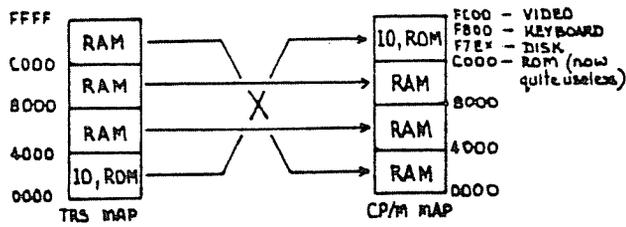
- 1 x 74LS04 hex inverter (see note below)
- 1 x 74LS367 hex tri-state bus driver
- 1 x 1 MΩ resistor
- 1 x 0.022μF greencap capacitor (Model I)
- 1 x 0.056μF greencap capacitor (System 80)
- 1 x SPST toggle switch
- 1 x push button (N.O.)
- PCB or Veroboard

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).

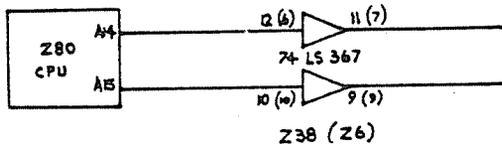
MEMORY REMAPPING

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

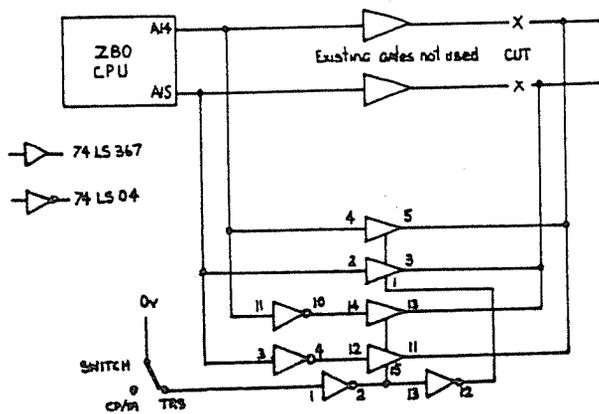


The CP/M memory mapping is achieved by inverting & swapping address bits A14 & A15 from the Z80 before they enter the address decoding circuits. In the following circuits all numbering refers to the Model 1 (System 80 numbering appears in parentheses).

EXISTING CIRCUIT:



MODIFIED 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

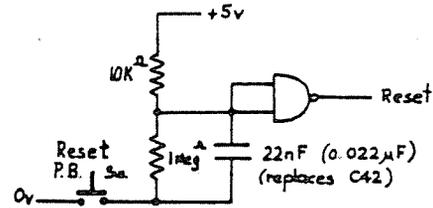
NOTE: - This modification possibly makes computer unsuitable for use with the standard Sys 80 S100 expansion unit as A14 & A15 can no longer be disabled by the *DODBS/ADDBS line. If this would be a problem, suggest using an LS00 NAND gate in place of the LS04 inverter. Connect *DODBS/ADDBS to one input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

RESET SWITCH

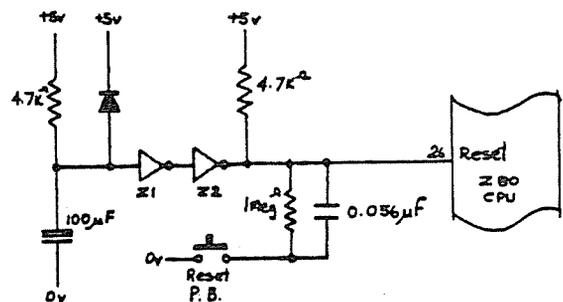
Requirements:

- 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 CPMTST1/ASM & CPMTST2/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 HSTDISK 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
0023 00620 DSP EQU 33H ;ROM DISPLAY ROUTINE
002D 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 PRTRREG EQU 0F7E8H ;PRINTER REGISTER
F7EC 00690 DCSREG EQU 0F7ECH ;DISK COMMAND/STATUS REG
F7ED 00700 TRKREG EQU 0F7E0H ;DISK TRACK REG
F7EE 00710 SECREG EQU 0F7EEH ;DISK SECTOR REG
F7EF 00720 DATREG EQU 0F7EFH ;DISK DATA REG
F7F0 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
0000 00760 BLKSIZ EQU 2040 ;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 SECMASK 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 C3685 00870 ORIGIN JP BOOT ;COLD START
B503 C3785 00880 WBOOT JP WBOOT ;WARM START
B506 C3486 00890 JP CONST ;CONSOLE STATUS
B509 C3586 00900 JP CONIN ;CONSOLE CHAR IN
B50C C39186 00910 JP CONOUT ;CONSOLE CHAR OUT
B50F C3A586 00920 JP LIST ;LIST CHAR OUT
B512 C3DD86 00930 JP PUNCH ;PUNCH OUT
B515 C3E086 00940 JP READER ;READER IN
B518 C30187 00950 JP HOME ;RESTORE DISK HEAD POS
B51B C31787 00960 JP SELDSK ;SELECT DISK
B51E C33087 00970 JP SETTRK ;SET TRACK NUMBER
B521 C35787 00980 JP SETSEC ;SET SECTOR NUMBER
B524 C35C87 00990 JP SETDMA ;SET DMA ADDRESS
B527 C36187 01000 JP READ ;READ DISK
B52A C36E87 01010 JP WRITE ;WRITE DISK
B52D C3E189 01020 JP LISTST ;LIST DEVICE STATUS
B530 C3EC89 01030 JP SECTRn ;SECTOR TRANSLATION
B533 C33985 01040 JP NU ;NOT USED
B536 C33785 01050 JP NU
B539 C9 01060 NU RET ;RETURN
B53A F3 01070 START DI ;DISABLE INTERRUPTS
B538 21098E 01080 LD HL,MESS2 ;DISPLAY MESSAGE 2
B53E C0B08A 01090 CALL SOSP ;SDSP
B541 C01B86 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 8E 01150 CP (HL) ;WRITE SUCCESSFUL?
B54E 20F7 01160 JR NZ,MAPTST ;IF NOT WAIT
B550 0605 01170 LD B,5 ;500MS DEBOUNCE DELAY
B552 C00B89 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 C0B98A 01240 CALL SDISP ;DISPLAY CP/M MESSAGE
B565 C37385 01250 JP WBOOT ;LOAD SYSTEM
B568 310000 01260 BOOT LD SP,0000H ;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 C08988 01300 WBOOT CALL BLINIT ;INIT BLOCKING
B576 310000 01310 LD SP,0000H ;INITIALISE STACK POINTER
B579 0E00 01320 LD C,0 ;SELECT DISK 0
B57B C017B7 01330 CALL SELDSK
B57E C001B7 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 C03087 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 05 01410 PUSH DE ;SAVE NEXT SEC TO READ
B58F 05 01420 PUSH HL ;SAVE DMA ADDRESS
B590 4A 01430 LD C,D ;GET SEC ADD TO REG C
B591 C057B7 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 C05C87 01470 CALL SETDMA ;SET DMA ADDRESS FROM BC
B599 C061B7 01480 ;DRIVE SET TO 0, TRACK SET, SECTOR SET, DMA SET
B59C FE00 01490 CALL READ ;READ LOGICAL SECTOR
B59E C27385 01500 CP 0 ;ANY ERRORS?
B59F 05 01510 JP NZ,WBOOT ;IF SO RE-BOOT
B5A1 E1 01520 ;NO ERROR: MOVE TO NEXT SECTOR
B5A2 118000 01530 POP HL ;RECALL DMA ADD
B5A5 19 01540 LD DE,12B ;DMA=DMA+12B
B5A6 01 01550 ADD HL,DE ;NEW DMA ADD IN HL
B5A7 C1 01560 POP DE ;RECALL SEC ADD
B5A8 05 01570 POP BC ;RECALL SECS REMAINING
B5A9 CAC2B5 01580 DEC B ;SECS=SECS-1
B5AC 14 01590 JP Z,GOCPM ;GOTO CP/M WHEN COMPLETE
B5AD 7A 01600 ;MORE SECTORS TO LOAD, CHECK FOR TRACK CHANGE
B5AE FE14 01610 INC D ;INCREMENT SECTOR COUNT
B5B0 0A80B5 01620 LD A,D ;IF SEC>19 CHANGE TRACK
B5B3 1600 01630 CP Z0 ;ELSE CONTINUE
B5B5 0C 01640 INC C ;TRACK=TRACK+1
B5B8 05 01650 ;END OF CURRENT TRACK, GO TO NEXT
B5B9 0C 01660 LD D,0 ;BEGIN WITH FIRST SEC
B5BA 05 01670 INC C ;TRACK=TRACK+1
B5BB 05 01680 ;SAVE REG STATES AND CHANGE TRACKS
B5BC C5 01690 PUSH BC
B5BD 05 01700 PUSH DE
B5BE 05 01710 PUSH HL
B5BF C03087 01720 CALL SETTRK
B5C0 E1 01730 POP HL
B5C1 01 01740 POP DE
B5C2 01 01750 POP BC
B5C3 0C80B5 01760 JP LOAD1 ;GET NEXT SECTOR
B5C4 C0F686 01770 ;END OF LOAD, SET PARAMETERS AND GO TO CP/M
B5C5 C08988 01780 GOCPM CALL RSNIN ;INITIALISE RS232 PORT
B5C6 C3C3 01790 CALL BLINIT ;INITIALISE BLOCKING
B5C7 3E00 01800 LD A,0C3H ;JUMP INSTRUCTION
B5C8 320000 01810 LD (0),A ;FOR JUMP TO WBOOT
B5C9 210385 01820 LD HL,WBOOT ;WBOOT ENTRY POINT
B5CA 220100 01830 LD (1),HL
B5CB 320500 01840 LD (5),A ;FOR JUMP TO BDOS
B5CC 2106A7 01850 LD HL,BDOS ;BDOS ENTRY POINT
B5CD 220600 01860 LD (6),HL
B5CE 3E00 01870 LD A,0C9H ;RETURN INSTRUCTION CODE

```

```

B5DE 320800 01800 LD (00H),A ;STORE AT INTERRUPTS 1-7
B5E1 321000 01890 LD (10H),A
B5E4 321000 01900 LD (10H),A
B5E7 322000 01910 LD (20H),A
B5EA 322000 01920 LD (20H),A
B5ED 323000 01930 LD (30H),A
B5F0 323000 01940 LD (30H),A
B5F3 018000 01950 LD BC,00H ;SET DEFAULT DMA ADD
B5F6 C05CB7 01960 CALL SETDMA
B5F9 F3 01970 DI ;DISABLE INTERRUPTS
B5FA 3A0400 01980 LD A,(DISK) ;GET CURRENT DRIVE
B5FD 4F 01990 LD C,A ;SEND TO THE CCP
B5FE C3009F 02000 JP CPMB ;GO TO CP/M
B601 2A0899 02010 CHECK LD HL,(TRNFER) ;SAVE TRNFER WORD
B604 55 02020 PUSH HL
B605 210000 02030 LD HL,0
B608 220889 02040 LD (TRNFER),HL ;ZERO IT
B60B 210085 02050 LD HL,ORIGIN ;LOAD START ADD
B60E 11398C 02060 LD DE,END+1 ;LOAD END ADD
B611 0E00 02070 LD C,0 ;ZERO SUM
B613 79 02080 CHECK 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 88 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 220889 02200 LD (TRNFER),HL
B624 C9 02210 RET ;RETURN
B625 C001B6 02220 VERCHK CALL CHECK ;CALCULATE CHECKSUM
B628 217B8C 02230 LD HL,CCKSUM
B62B BE 02240 CP (HL) ;SAME AS STORED VALUE?
B62C CB 02250 RET Z ;RETURN IF 50
B62D 219ABF 02260 LD HL,MESS5
B630 C0B9BA 02270 CALL SD15P ;DISPLAY MESSAGE 5
B633 C9 02280 RET ;RETURN
; I/O HANDLER SECTION BEGINS HERE
B634 C07DB6 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,CONROY ;JUMP IF READY
B63D 3A0300 02340 LD A,(10BYTE) ;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 2008 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 CONROY 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
; CONSOLE CHARACTER INTO REGISTER A
B655 3A6FBC 02470 CONIN LD A,(CHAR) ;GET CHARACTER
B658 E67F 02480 AND 7FH ;RESET HIGH BIT
B65A 2000 02490 LD Z,WAITK ;JUMP IF NO CHAR
B65C 4F 02500 JR 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 C07DB6 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,(10BYTE) ;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,(0F02H) ;IF JKL PRESSED
B680 FE1C 02690 CP 1CH
B682 CCC786 02700 CALL Z,PRTPSCR ;THEN PRINT SCREEN
B685 3A10F8 02710 LD A,(0F10H) ;IF 123 PRESSED
B688 FE0E 02720 CP 0EH
B68A C9988B 02730 CALL Z,TERMIN ;THEN TERMINAL MODE
B68D C0C4BA 02740 CALL KEY ;SCAN KEYBOARD
B690 C9 02750 RET
B691 79 02760 ; CONSOLE CHARACTER OUTPUT FROM REGISTER C
B692 C5 02770 CONOUT LD A,C ;PUT CHAR IN A
B693 C0F7B9 02780 PUSH BC
B696 C1 02790 CALL DISP ;DISPLAY IT
B697 3A0300 02800 POP BC
B699 EA03 02810 LD A,(10BYTE) ;GET I/O BYTE
B69C CAE3B6 02820 AND 03H ;MASK CONSOLE FIELD
B69F FE02 02830 JP Z,RS2320 ;0 - SEND TO RS232 PORT
B6A1 CA58B6 02840 CP 2
B6A4 C9 02850 JP Z,LIST ;2 - SEND TO LIST DEVICE
; 1,3 - CRT DISP ONLY
B6A5 3A0300 02860 LIST LD A,(10BYTE) ;GET I/O BYTE
B6A8 07 02870 RLCA ;EXTRACT LIST DEV NUMBER
B6A9 07 02880 RLCA
B6AA E603 02890 AND 03H
B6AC CAE3B6 02900 JP Z,RS2320 ;0 - SEND TO RS232
B6AF FE02 02910 CP 2
B6B1 CABB86 02920 JP Z,PRINT ;2 - SEND TO PRINTER
B6B4 79 02930 LD A,C
B6B5 C3F7B9 02940 JP DISP ;1,3 - SEND TO CRT DISP
; PRINT CHARACTER FROM REGISTER C
B6B8 3AEBF7 02950 PRINT LD A,(PRTREG) ;GET PRINTER STATUS
B6BB CB7F 02960 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 CB 03030 RET Z
B6C3 32EBF7 03040 LD (PRTREG),A ;PRINT CHAR
B6C6 C9 03050 RET ;RETURN
; SCREEN PRINT ROUTINE
B6C7 2100FC 03070 PRTPSCR 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 C0B8B6 03100 CALL PRINT ;PRINT IT
B6D0 23 03110 INC HL ;ADVANCE POINTER
B6D1 10F9 03120 DJNZ PRTCHR ;LOOP TILL EOL
B6D3 0E00 03130 LD C,00H
B6D5 C0B8B6 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
; PUNCH CHARACTER OUT
B6DD C3E3B6 03190 PUNCH JP RS2320 ;SEND TO RS232 PORT
; READER CHARACTER IN
B6E0 C3E0B6 03200 READER JP RS2321 ;GET FROM RS232 PORT
; SEND CHARACTER TO RS232 PORT
B6E3 DBEA 03210 IN A,(RSSTAT) ;GET UART STATUS
B6E5 CB7F 03220 BIT 6,A ;IS TX REG EMPTY ?
B6E7 20FA 03230 JR Z,RS2320 ;IF NOT WAIT
B6E9 79 03240 LD A,C ;GET CHARACTER
B6EA D3EB 03250 OUT (RSDATA),A ;SEND IT
B6EC C9 03260 RET ;RETURN
; READ CHARACTER FROM RS232 PORT
B6ED DBEA 03270 RS2321 IN A,(RSSTAT) ;GET UART STATUS
B6EF CB7F 03280 BIT 7,A ;HAS CHAR ARRIVED ?
B6F1 20FA 03290 JR Z,RS2321 ;IF NOT WAIT
B6F3 DBEB 03300 IN A,(RSDATA) ;LOAD CHAR TO A
B6F5 C9 03310 RET ;RETURN
; INITIALISE RS232 PORT
B6F6 D3EB 03320 RSINIT OUT (RSCTRL),A ;RESET UART
B6F8 3E55 03330 LD A,55H ;SELECT 300 BAUD
B6FA D3E9 03340 OUT (RSBAUD),A
B6FC 3E6F 03350 LD A,6FH ;8 BITS,1 STOP,NO PAR
B6FE D3EA 03360 OUT (RSSTAT),A ;RETURN
; DISK I/O DRIVER SECTION FOLLOWS
B700 C9 03370 ; MOVE TO THE TRACK 00 POSITION OF CURRENT DRIVE
B701 3A4000 03380 HOME LD A,(SEKDSK) ;GET SELECTED DISK
B704 C094B9 03390 CALL DSKROY ;WAIT TILL DISK READY
B707 3E02 03400 HOME LD A,02 ;RESTORE COMMAND

```

| | | | | | | | | | |
|---------------|-------|------|---------------------------------------|--------------------------|-------------|-------|------|---|--------------------------|
| B709 32ECF7 | 03480 | LD | (DCSREG),A | ;SEND TO CONTROLLER | B7AA 214B00 | 04280 | LD | HL,UNATRK | |
| B70C C0DEB9 | 03490 | CALL | SDELAY | | B7AD BE | 04290 | CP | (HL) | ;SEKTRK = UNATRK ? |
| B70F 3AECF7 | 03500 | LD | A,(DCSREG) | ;GET DISK STATUS | B7AE C203B7 | 04300 | JP | NZ,ALLOC | ;SKIP IF NOT |
| B712 CB47 | 03510 | BIT | 0,A | | | 04310 | | TRACKS ARE THE SAME | |
| B714 20F9 | 03520 | JR | NZ,H0MEW | ;WAIT IF BUSY | B7B1 3A7CBC | 04320 | LD | A,(LOGSEC) | |
| B716 C9 | 03530 | RET | | ;ELSE RETURN | B7B4 214C00 | 04330 | LD | HL,UNASEC | |
| | 03540 | ; | SELECT DISK GIVEN BY REGISTER C | | B7B7 BE | 04340 | CP | (HL) | ;LOGSEC = UNASEC ? |
| B717 210000 | 03550 | LD | HL,0 | ;ERROR CODE | B7B8 C203B7 | 04350 | JP | NZ,ALLOC | |
| B71A 79 | 03560 | LD | A,C | ;GET DRIVE NUMBER | | 04360 | | MATCH, MOVE TO NEXT SECTOR FOR FUTURE REF | |
| B71B FE03 | 03570 | CP | NDISKS | ;TOO LARGE? | B7BB 34 | 04370 | INC | (HL) | ;UNASEC = UNASEC+1 |
| B71D D0 | 03580 | RET | NC | ;RETURN WITH HL=0 IF SO | B7BC 46 | 04380 | LD | B,(HL) | ;END OF TRACK ? |
| B71E 324000 | 03590 | LD | (SEKDSK),A | ;SEEK DISK NUMBER | B7BD 3A73BC | 04390 | LD | A,(CPMSPT) | ;GET SECTORS PER TRACK |
| B721 69 | 03600 | LD | L,C | ;HL=DRIVE NUMBER | B7C0 3D | 04400 | DEC | A | |
| B722 2600 | 03610 | LD | H,0 | | B7C1 B8 | 04410 | CP | B | |
| B724 29 | 03620 | ADD | HL,HL | ;HL=HL*16 | B7C2 D2C8B7 | 04420 | JP | NC,NOOVF | ;SKIP IF NO OVERFLOW |
| B725 29 | 03630 | ADD | HL,HL | | | 04430 | | OVERFLOW TO NEXT TRACK | |
| B726 29 | 03640 | ADD | HL,HL | | B7C5 3600 | 04440 | LD | (HL),0 | ;UNASEC = 0 |
| B727 29 | 03650 | ADD | HL,HL | | B7C7 214B00 | 04450 | LD | HL,UNATRK | |
| B728 1139BC | 03660 | LD | DE,DCB1 | ;DE=DISK CONT BLOCK ADD. | B7CA 34 | 04460 | INC | (HL) | ;UNATRK = UNATRK+1 |
| B72B 19 | 03670 | ADD | HL,DE | ;ADD DRIVE OFFSET | | 04470 | | MATCH FOUND, MARK AS UNNECESSARY READ | |
| B72C 2279BC | 03680 | LD | (CURDCB),HL | ;SAVE CURRENT DCB ADD | B7CB 3E00 | 04480 | LD | A,0 | |
| B72F C9 | 03690 | RET | | ;RETURN | B7CD 324E00 | 04490 | LD | (RSFLAG),A | ;RSFLAG = 0 |
| | 03700 | ; | SET TRACK GIVEN BY REGISTER C | | B7D0 C3DCB7 | 04500 | JP | RWOPER | ;TO PERFORM THE WRITE |
| B730 79 | 03710 | LD | A,C | | | 04510 | | NOT UNALLOCATED RECORD, REQUIRES PRE-READ | |
| B731 324100 | 03720 | LD | (SEKTRK),A | ;SAVE TRACK TO SEEK | B7D3 3E00 | 04520 | LD | A,0 | |
| B734 C9 | 03730 | RET | | ;RETURN | B7D5 324900 | 04530 | LD | (UNACNT),A | ;UNACNT = 0 |
| B735 3A4300 | 03740 | LD | A,(HSTDISK) | ;GET CURRENT DISK | B7D8 3C | 04540 | INC | A | |
| B738 C094B7 | 03750 | CALL | DKRDY | ;READY DISK | B7D9 324E00 | 04550 | LD | (RSFLAG),A | ;RSFLAG = 1 |
| B73B 3A4400 | 03760 | LD | A,(HSTTRK) | ;GET TRACK TO SEEK | | 04560 | | ENTER HERE TO PERFORM THE READ/WRITE | |
| B73E 32EFF7 | 03770 | LD | (DATREG),A | ;SEND TO DISK | B7DC 3E00 | 04570 | LD | A,0 | |
| B741 3E12 | 03780 | LD | A,12H | | B7DE 324D00 | 04580 | LD | (ERFLAG),A | ;NO ERRORS (YET) |
| B743 32ECF7 | 03790 | LD | (DCSREG),A | ;PERFORM TRACK SEEK | B7E1 3A4200 | 04590 | LD | A,(SEKSEC) | ;GET SECTOR TO SEEK |
| B746 C0DEB9 | 03800 | CALL | SDELAY | ;SHORT DELAY | B7E4 CB3F | 04600 | SRL | A | ;COMPUTE HOST SECTOR *** |
| B749 3AECF7 | 03810 | LD | A,(DCSREG) | ;GET DISK STATUS | B7E6 3C | 04610 | INC | A | |
| B74C CB47 | 03820 | BIT | 0,A | | B7E7 324600 | 04620 | LD | (SEKHST),A | ;HOST SECTOR TO SEEK |
| B74E 20F9 | 03830 | JR | NZ,SEEKW | ;WAIT TILL FINISHED | | 04630 | | ACTIVE HOST SECTOR ? | |
| B750 3A4500 | 03840 | LD | A,(HSTSEC) | ;GET SECTOR TO SEEK | B7EA 214700 | 04640 | LD | HL,HSTACT | ;HOST ACTIVE FLAG |
| B753 32EEF7 | 03850 | LD | (SEKREG),A | ;SEND TO DISK | B7ED 7E | 04650 | LD | A,(HL) | |
| B756 C9 | 03860 | RET | | ;THEN RETURN | B7EE 3601 | 04660 | LD | (HL),1 | ;ALWAYS BECOMES 1 |
| | 03870 | ; | SET SECTOR GIVEN BY REGISTER C | | B7F0 97 | 04670 | OR | A | ;WAS IT ACTIVE ALREADY ? |
| B757 79 | 03880 | LD | A,C | | B7F1 CA23B9 | 04680 | JP | Z,FILHST | ;FILL HOST IF NOT |
| B758 324200 | 03890 | LD | (SEKSEC),A | ;SAVE SECTOR TO SEEK | | 04690 | | HST BUFFER ACTIVE, SAME AS SEEK BUFFER ? | |
| B75B C9 | 03900 | RET | | ;RETURN | B7F4 3A4000 | 04700 | LD | A,(SEKDSK) | |
| | 03910 | ; | SET DMA ADDRESS GIVEN BY REGISTER BC | | B7F7 214300 | 04710 | LD | HL,HSTDISK | ;SAME DISK ? |
| B75C E04369BC | 03920 | LD | (DMAADR),BC | ;SAVE DMA ADDRESS | B7FA BE | 04720 | CP | (HL) | ;SEKDSK = HSTDISK ? |
| B760 C9 | 03930 | RET | | ;RETURN | B7FB C21CB8 | 04730 | JP | NZ,NOMTCH | |
| | 03940 | ; | READ THE SELECTED SECTOR | | | 04740 | | SAME DISK, SAME TRACK ? | |
| B761 3E01 | 03950 | LD | A,1 | | B7FE 3A4100 | 04750 | LD | A,(SEKTRK) | |
| B763 324F00 | 03960 | LD | (READOP),A | ;FLAG READ OPERATION | B801 214400 | 04760 | LD | HL,HSTTRK | |
| B766 3E02 | 03970 | LD | A,WRUAL | | B804 BE | 04770 | CP | (HL) | ;SEKTRK = HSTTRK ? |
| B768 3271BC | 03980 | LD | (WRTYPE),A | ;TREAT AS UNALLOC | B805 C21CB8 | 04780 | JP | NZ,NOMTCH | |
| B76B C303B7 | 03990 | JP | ALLOC | ;TO PERFORM THE READ | | 04790 | | SAME DISK, SAME TRACK, SAME BUFFER ? | |
| B76E 3E00 | 04000 | LD | A,0 | | B808 3A4600 | 04800 | LD | A,(SEKHST) | |
| B770 324F00 | 04010 | LD | (READOP),A | ;NOT A READ OPERATION | B80B 214500 | 04810 | LD | HL,HSTSEC | ;SEKST = HSTSEC ? |
| B773 79 | 04020 | LD | A,C | ;LOAD WRITE TYPE | B80E BE | 04820 | CP | (HL) | |
| B774 3271BC | 04030 | LD | (WRTYPE),A | ;STORE IT | B80F C21CB8 | 04830 | JP | NZ,NOMTCH | |
| B777 FE02 | 04040 | CP | WRUAL | ;WRITE UNALLOCATED ? | B812 3A4200 | 04840 | LD | A,(SEKSEC) | |
| B779 2017 | 04050 | JR | NZ,CHKUNA | ;CHECK FOR UNALLOC | B815 2170BC | 04850 | LD | HL,LSTSEC | ;SEKSEC = LSTSEC ? |
| | 04060 | ; | WRITE TO UNALLOCATED, SET PARAMETERS | | B818 BE | 04860 | CP | (HL) | |
| B77B 3E10 | 04070 | LD | A,CPMSPT | ;CPM SECTORS PER BLOCK | B819 C247B8 | 04870 | JP | NZ,MATCH | ;REDO IF EXACT MATCH |
| B77D 324900 | 04080 | LD | (UNACNT),A | ;NEXT UNALLOC RECS | | 04880 | | CORRECT DISK, BUT INCORRECT SECTOR | |
| B780 3A4000 | 04090 | LD | A,(SEKDSK) | ;DISK TO SEEK | B81C 3A4800 | 04890 | LD | A,(HSTWRT) | ;HOST WRITTEN ? |
| B783 324A00 | 04100 | LD | (UNADSK),A | ;UNADSK = SEKDSK | B81F B7 | 04900 | OR | A | |
| B786 3A4100 | 04110 | LD | A,(SEKTRK) | | B820 C4A2B8 | 04910 | CALL | NZ,WRTHST | ;CLEAR HOST BUFFER |
| B789 324B00 | 04120 | LD | (UNATRK),A | ;UNATRK = SEKTRK | | 04920 | | MAY HAVE TO FILL HOST BUFFER | |
| B78C 3A7CBC | 04130 | LD | A,(LOGSEC) | | B823 3A4000 | 04930 | LD | A,(SEKDSK) | |
| B78F 324C00 | 04140 | LD | (UNASEC),A | ;UNASEC = LOGSEC | B826 324300 | 04940 | LD | (HSTDISK),A | |
| | 04150 | ; | CHECK FOR WRITE TO UNALLOCATED SECTOR | | B829 3A4100 | 04950 | LD | A,(SEKTRK) | |
| B792 3A4900 | 04160 | LD | A,(UNACNT) | ;ANY UNALLOC REMAIN ? | B82C 324400 | 04960 | LD | (HSTTRK),A | |
| B795 B7 | 04170 | OR | A | | B82F 3A4600 | 04970 | LD | A,(SEKHST) | |
| B796 CA03B7 | 04180 | JP | Z,ALLOC | ;SKIP IF NOT | B832 324500 | 04980 | LD | (HSTSEC),A | |
| | 04190 | ; | MORE UNALLOCATED RECORDS REMAIN | | B835 3A4200 | 04990 | LD | A,(SEKSEC) | |
| B799 3D | 04200 | DEC | A | ;UNACNT = UNACNT-1 | B838 3270BC | 05000 | LD | (LSTSEC),A | |
| B79A 324900 | 04210 | LD | (UNACNT),A | | B83B 3A4E00 | 05010 | LD | A,(RSFLAG) | ;NEED TO READ ? |
| B79D 3A4000 | 04220 | LD | A,(SEKDSK) | ;SAME DISK ? | B83E B7 | 05020 | OR | A | |
| B7A0 214A00 | 04230 | LD | HL,UNADSK | | B83F C492B8 | 05030 | CALL | NZ,ROHST | ;YES, IF 1 |
| B7A3 BE | 04240 | CP | (HL) | ;SEKDSK = UNADSK ? | B842 3E00 | 05040 | LD | A,0 | |
| B7A4 C203B7 | 04250 | JP | NZ,ALLOC | ;SKIP IF NOT | B844 324800 | 05050 | LD | (HSTWRT),A | ;NO PENDING WRITE |
| | 04260 | ; | DISKS ARE THE SAME | | | 05060 | | COPY DATA TO OR FROM BUFFER | |
| B7A7 3A4100 | 04270 | LD | A,(SEKTRK) | | B847 3A4200 | 05070 | LD | A,(SEKSEC) | ;MASK BUFFER NUMBER |

| | | | | | | | | | | | |
|-------------|---------|--|-------------|--------------------------|--------------------|-------------|--------|-------------|----------------------------|-------------------------|-------------------------|
| B84A E601 | 05000 | AND | SECMSK | ;LEAST SIG BITS | B8E5 32EDF7 | 05800 | LD | (TRKREG),A | | | |
| B84C 6F | 05090 | LD | L,A | ;READY TO SHIFT | B8E8 11EF77 | 05890 | LD | DE,0ATREG | ;DISK DATA REGISTER | | |
| B84D 2600 | 05100 | LD | H,0 | | B8EB 0185BC | 05900 | LD | BC,HSTBUF | ;DISK BUFFER BASE ADD | | |
| B84F 29 | 05110 | ADD | HL,HL | ;SHIFT LEFT 7 | B8EE 3A74BC | 05910 | LD | A,(DSKCOM) | ;GET COMMAND | | |
| B850 29 | 05120 | ADD | HL,HL | | B8F1 77 | 05920 | LD | (HL),A | ;ISSUE COMMAND | | |
| B851 29 | 05130 | ADD | HL,HL | | B8F2 CDDEB9 | 05930 | CALL | SDELAY | ;SHORT DELAY | | |
| B852 29 | 05140 | ADD | HL,HL | | B8F5 CB6F | 05940 | BIT | 5,A | | | |
| B853 29 | 05150 | ADD | HL,HL | | B8F7 CA05B9 | 05950 | JP | Z,WAIT2 | ;JUMP IF READ | | |
| B854 29 | 05160 | ADD | HL,HL | | B8FA 7E | 05960 | WAIT1 | LD | A,(HL) | ;WAIT FOR DATA REQUEST | |
| B855 29 | 05170 | ADD | HL,HL | | B8FB E603 | 05970 | AND | 03H | | | |
| B856 1185BC | 05180 | LD | DE,HSTBUF | ;HOST BUFFER BASE | B8FD E2FAB0 | 05980 | JP | PO,WAIT1 | | | |
| B859 19 | 05190 | ADD | HL,DE | ;ADD OFFSET | B900 0A | 05990 | LD | A,(BC) | ;GET FIRST BYTE TO WRT | | |
| B85A EB | 05200 | EX | DE,HL | ;PUT RESULT IN DE | B901 03 | 06000 | INC | BC | ;INCREMENT POINTER | | |
| B85B 2A69BC | 05210 | LD | HL,(DMAADR) | ;CPM DATA ADDRESS | B902 C30BB9 | 06010 | JP | TRNFER | ;BEGIN TRANSFER | | |
| B85E 0E00 | 05220 | LD | C,12B | | B905 7E | 06020 | WAIT2 | LD | A,(HL) | ;GET DISK STATUS | |
| B860 0600 | 05230 | LD | B,0 | ;128 BYTES TO MOVE | B906 E603 | 06030 | AND | 03H | ;READY FOR FIRST BYTE ? | | |
| B862 3A4F00 | 05240 | LD | A,(READOP) | ;WHICH WAY ? | B908 E205B9 | 06040 | JP | PO,WAIT2 | ;WAIT IF NOT | | |
| B865 B7 | 05250 | OR | A | | B90B 0A | 06050 | TRNFER | LD | A,(BC) | ;N.B. READ CASE SHOWN | |
| B866 C26F80 | 05260 | JP | NZ,RWMOVE | ;SKIP IF READ | B90C 12 | 06060 | LD | (DE),A | ;INSTRUCTS CHANGED FOR WRT | | |
| | 05270 ; | WRITE OPERATION; MARK AND SWITCH DIRECTION | | | B90D 03 | 06070 | INC | BC | ;INC BUFFER POINTER | | |
| B869 3E01 | 05280 | LD | A,1 | | B90E CB4E | 06080 | WAIT3 | BIT | 1,(HL) | ;READY FOR NEXT BYTE? | |
| B86B 324000 | 05290 | LD | (HSTWRT),A | ;HSTWRT = 1 | B910 20F9 | 06090 | JR | NZ,TRNFER | ;IF SO JUMP | | |
| B86E EB | 05300 | EX | DE,HL | ;SOURCE/DEST SWAP | B912 CB4E | 06100 | BIT | 1,(HL) | | | |
| | 05310 ; | MOVE 128 BYTES FROM DE TO HL | | | B914 20F5 | 06110 | JR | NZ,TRNFER | | | |
| B86F EB | 05320 | RWMOVE | EX | DE,HL | B916 CB4E | 06120 | BIT | 1,(HL) | | | |
| B870 E0B0 | 05330 | LDIR | | ;MOVE DATA BLOCK | B918 20F1 | 06130 | JR | NZ,TRNFER | | | |
| | 05340 ; | DATA HAS BEEN MOVED TO/FROM HOST BUFFER | | | B91A CB4E | 06140 | BIT | 0,(HL) | ;OPERATION COMPLETE ? | | |
| B872 3A71BC | 05350 | LD | A,(WRTYPE) | ;WRITE TYPE | B91C 2012 | 06150 | JR | Z,RWCOMP | | | |
| B875 FE01 | 05360 | CP | WRDIR | ;TO DIRECTORY ? | B91E CB4E | 06160 | BIT | 1,(HL) | | | |
| B877 3A4D00 | 05370 | LD | A,(ERFLAG) | ;IN CASE OF ERRORS | B920 20E9 | 06170 | JR | NZ,TRNFER | | | |
| B87A C0 | 05380 | RET | NZ | ;NO FURTHER PROCESSING | B922 CB7E | 06180 | BIT | 7,(HL) | ;TIMED OUT ? | | |
| | 05390 ; | CLEAR HOST BUFFER FOR DIRECTORY WRITE | | | B924 20E8 | 06190 | JR | Z,WAIT3 | ;LOOP IF NOT | | |
| B87B B7 | 05400 | OR | A | ;ERRORS ? | B926 3600 | 06200 | LD | (HL),0D0H | ;FORCE INTERRUPT | | |
| B87C C0 | 05410 | RET | NZ | ;SKIP IF SO | B928 2150BF | 06210 | LD | HL,MESS3 | | | |
| B870 3E00 | 05420 | LD | A,0 | | B92B C0B9BA | 06220 | CALL | SDISP | | | |
| B87F 324000 | 05430 | LD | (HSTWRT),A | ;BUFFER WRITTEN | B92E 185E | 06230 | JR | RWERR | | | |
| B882 CDA2B0 | 05440 | CALL | WRTHST | | B930 7E | 06240 | RWCOMP | LD | A,(HL) | ;GET STATUS | |
| B885 3A4D00 | 05450 | LD | A,(ERFLAG) | | B931 3276BC | 06250 | LD | (ERCODE),A | ;SAVE ERROR CODE | | |
| B888 C9 | 05460 | RET | | ;RETURN | B934 5F | 06260 | LD | E,A | | | |
| | 05470 ; | INITIALISE BLOCKING FLAGS | | | B935 3A72BC | 06270 | LD | A,(ERMASK) | | | |
| B889 3E00 | 05480 | BLINIT | LD | A,0 | B938 A3 | 06280 | AND | E | ;MASK CODE | | |
| B88B 324700 | 05490 | LD | (HSTACT),A | ;HOST BUFFER INACTIVE | B939 324D00 | 06290 | LD | (ERFLAG),A | | | |
| B88E 324900 | 05500 | LD | (UNACNT),A | ;CLEAR UNALLOC COUNT | B93C C0 | 06300 | RET | Z | ;RETURN IF NO ERROR | | |
| B891 C9 | 05510 | RET | | ;RETURN | B93D 3A75BC | 06310 | LD | A,(TRYCNT) | ;GET RETRY COUNT | | |
| | 05520 ; | READ PHYSICAL SECTOR INTO SECBUF | | | B940 3D | 06320 | DEC | A | ;DECREMENT IT | | |
| B892 3E1C | 05530 | RDHST | LD | A,1CH | ;SET ERROR MASK | B941 3275BC | 06330 | LD | (TRYCNT),A | ;SAVE IT | |
| B894 3272BC | 05540 | LD | (ERMASK),A | | B944 2848 | 06340 | JR | Z,RWERR | ;ERROR IF COUNT = 0 | | |
| B897 211A02 | 05550 | LD | HL,021AH | ;SET TRANSFER DIRECTION | B946 3A74BC | 06350 | LD | A,(DSKCOM) | ;GET DISK COMMAND | | |
| B89A 220BB9 | 05560 | LD | (TRNFER),HL | | B949 CB6F | 06360 | BIT | 5,A | | | |
| B89D 3E0C | 05570 | LD | A,0CH | ;SECTOR READ COMMAND | B94B C2CABB | 06370 | JP | NZ,RETRY | ;JUMP IF WRITE | | |
| B89F C3B2B0 | 05580 | JP | DISKRW | ;PERFORM READ | B94E 2A77BC | 06380 | LD | HL,(DENPT) | ;LOAD DENSITY POINTER | | |
| | 05590 ; | WRITE PHYSICAL SECTOR FROM SECBUF | | | B951 7E | 06390 | LD | A,(HL) | ;GET DENSITY MODE | | |
| B8A2 3E7C | 05600 | WRTHST | LD | A,7CH | ;SET ERROR MASK | B952 2F | 06400 | CPL | | ;COMPLEMENT IT | |
| B8A4 3272BC | 05610 | LD | (ERMASK),A | | B953 E601 | 06410 | AND | 1 | ;MASK IT | | |
| B8A7 21120A | 05620 | LD | HL,0A12H | ;SET TRANSFER DIRECTION | B955 77 | 06420 | LD | (HL),A | ;SAVE NEW DENSITY | | |
| B8AA 220BB9 | 05630 | LD | (TRNFER),HL | | B956 2000 | 06430 | JR | NZ,DOUBLE | | | |
| B8AD 3EAC | 05640 | LD | A,0ACH | ;SECTOR WRITE COMMAND | B958 01E2B0 | 06440 | SINGLE | LD | BC,SECTN1 | ;SECTOR TRANSLATE TAB 1 | |
| B8AF C3B2B0 | 05650 | JP | DISKRW | ;PERFORM WRITE | B95B 111ABC | 06450 | LD | DE,DPB1 | ;DISK PARAM BLOCK 1 | | |
| | 05660 ; | DISK SECTOR READ/WRITE | | | B95E 1006 | 06460 | JR | SETDEN | ;SET NEW DENSITY | | |
| B882 3274BC | 05670 | DISKRW | LD | (DSKCOM),A | ;SAVE DISK COMMAND | B960 01F6B0 | 06470 | DOUBLE | LD | BC,SECTN2 | ;SECTOR TRANSLATE TAB 2 |
| B885 C035B7 | 05680 | CALL | SEEK | ;SEEK DISK | B963 1129BC | 06480 | LD | DE,DPB2 | ;DISK PARAM BLOCK 2 | | |
| B888 3E06 | 05690 | LD | A,6 | ;MAX RETRY COUNT | B966 D02A79BC | 06490 | SETDEN | LD | IX,(CURDCB) | ;CURRENT DCB ADDRESS | |
| B88A 3275BC | 05700 | LD | (TRYCNT),A | | B96A D07100 | 06500 | LD | (IX+0),C | ;INSERT NEW TRANSLATE | | |
| B88D 2102BC | 05710 | LD | HL,DENTAB | ;DENSITY TABLE BASE | B96D D07001 | 06510 | LD | (IX+1),B | ;TABLE ADDRESS | | |
| B8C0 3A4300 | 05720 | LD | A,(HSTDSK) | ;CURRENT DISK NUMBER | B970 D0730A | 06520 | LD | (IX+0AH),E | ;INSERT NEW DISK | | |
| B8C3 5F | 05730 | LD | E,A | | B973 D0720B | 06530 | LD | (IX+0BH),D | ;PARAMETER BLOCK ADDRESS | | |
| B8C4 1600 | 05740 | LD | D,0 | | B976 ED43D0B4 | 06540 | LD | (0B400H),BC | ;GIVE BOOS NEW SECTRAN | | |
| B8C6 19 | 05750 | ADD | HL,DE | ;ADD TO BASE ADD | B97A E053BBB4 | 06550 | LD | (0B4BBH),DE | ;GIVE BOOS NEW DPB | | |
| B8C7 2277BC | 05760 | LD | (DENPT),HL | ;SAVE POINTER TO DENSITY | B97E 1A | 06560 | LD | A,(DE) | ;GET NEW SECTS PER TRACK | | |
| B8CA 3A4300 | 05770 | RETRY | LD | A,(HSTDSK) | ;GET CURRENT DISK | B97F 3273BC | 06570 | LD | (CPMSPT),A | ;SAVE IT | |
| B8CD C094B9 | 05780 | CALL | OSKRDY | ;READY DISK | B982 21C1B4 | 06580 | LD | HL,0B4C1H | ;BOOS WORKING DPB ADD | | |
| B8D0 2A77BC | 05790 | LD | HL,(DENPT) | ;GET DENSITY POINTER | B985 EB | 06590 | EX | DE,HL | ;PUT LOCAL DPB ADD IN HL | | |
| B8D3 7E | 05800 | LD | A,(HL) | ;GET DENSITY MODE | B986 010F00 | 06600 | LD | BC,0FH | | | |
| B8D4 F6FE | 05810 | OR | 0FEH | ;SET BITS 1-7 | B989 E0B0 | 06610 | LDIR | | ;TRANSFER 15 BYTES | | |
| B8D6 21ECF7 | 05820 | LD | HL,DCSREG | ;DISK COMMAND/STAT REG | B98B C3CAB0 | 06620 | JP | RETRY | ;RETRY SECTOR READ | | |
| B8D9 77 | 05830 | LD | (HL),A | ;SET DENSITY | B98E 3E01 | 06630 | RWERR | LD | A,1 | ;ELSE SET ERROR FLAG | |
| B8DA 3600 | 05840 | LD | (HL),0D0H | ;FORCE INTERRUPT | B990 324D00 | 06640 | LD | (ERFLAG),A | | | |
| B8DC 3A4500 | 05850 | LD | A,(HSTSEC) | ;TRANSFER CURRENT SECTOR | B993 C9 | 06650 | RET | | ;AND RETURN | | |
| B8DF 32EEF7 | 05860 | LD | (SECREG),A | | B994 4F | 06660 | OSKRDY | LD | C,A | | |
| B8E2 3A4400 | 05870 | LD | A,(HSTTRK) | ;TRANSFER CURRENT TRACK | B995 3E01 | 06670 | LD | A,1 | ;INITIALISE A | | |


```

08280 ; KEYBOARD SCAN ROUTINE
BAC4 3AE0F7 08290 KEY LD A,(DRVREG) ;AUTO REPEAT FUNCTION
BAC7 CB7F 08300 BIT 7,A ;CHECK FOR 25MS PULSE
BAC9 2824 08310 JR Z,KEY1
BACB 3A7EBC 08320 LD A,(KEYACT) ;CHECK FOR ACTIVE KEY
BACE A7 08330 AND A
BACF 281E 08340 JR Z,KEY1
BAD1 3A80BC 08350 LD A,(RPTLAG) ;GET REPEAT LAG
BAD4 A7 08360 AND A
BAD5 2806 08370 JR Z,REPEAT ;REPEAT IF EXPIRED
BAD7 3D 08380 DEC A ;ELSE COUNTDOWN
BAD8 3280BC 08390 LD (RPTLAG),A ;AND SAVE
BADB 2012 08400 JR NZ,KEY1
BADD 3A81BC 08410 REPEAT LD A,(RPTRAT) ;GET REPEAT RATE
BAE0 3D 08420 DEC A ;COUNTDOWN
BAE1 3281BC 08430 LD (RPTRAT),A ;SAVE IT
BAE4 2809 08440 JR NZ,KEY1
BAE6 3E03 08450 LD A,3 ;RESET RATE
BAE8 3281BC 08460 LD (RPTRAT),A ;SAVE IT
BAEB 3A7FBC 08470 LD A,(LSTKEY) ;GET LAST KEY PRESSED
BAEE C9 08480 RET ;AND RETURN
BAEF 2105BE 08490 KEY1 LD HL,KEYBUF ;KEYBOARD BUFFER ADD
BAF2 0101F8 08500 LD BC,(0FB01H) ;BC=ROW ADD POINT
BAF5 1600 08510 LD D,0 ;D=ROW COUNTER
BAF7 0A 08520 NXTROW LD A,(BC) ;GET ROW
BAF8 5F 08530 LD E,A ;SAVE IN E
BAF9 AE 08540 XOR (HL) ;CHANGE IN ROW ?
BAFA 280A 08550 JR Z,NOCHNG ;SKIP IF NO CHANGE
BAFC F5 08560 PUSH AF
BAFD AF 08570 XOR A
BAFE 327EBC 08580 LD (KEYACT),A ;CANCEL ACTIVE FLAG
BB01 F1 08590 POP AF
BB02 73 08600 LD (HL),E ;SAVE BYTE
BB03 A3 08610 AND E ;IS KEY PRESSED?
BB04 2008 08620 JR NZ,PRESS
BB06 14 08630 NOCHNG INC D ;INC ROW COUNTER
BB07 23 08640 INC HL ;INC BUFF POINTER
BB08 CB01 08650 RLC C ;POINT TO NEXT ROW
BB0A F2F7BA 08660 JP P,NXTROW ;LOOP TILL DONE
BB0D C9 08670 RET ;RET IF NO KEY PRESSED
BB0E 5F 08680 PRESS LD E,A ;SAVE ROW BYTE
BB0F 3E01 08690 LD A,1
BB11 327EBC 08700 LD (KEYACT),A ;SET KEY ACTIVE FLAG
BB14 3E1E 08710 LD A,30
BB16 3280BC 08720 LD (RPTLAG),A ;SET DELAY BEFORE REPEAT
BB19 7A 08730 LD A,D ;GET ROW COUNT
BB1A 07 08740 RLCA
BB1B 07 08750 RLCA
BB1C 07 08760 RLCA
BB1D 57 08770 LD D,A ;D=ROW COUNT * 8
BB1E 0E01 08780 LD C,1 ;SET TESTING MASK
BB20 79 08790 NXCOT LD A,C ;LOAD MASK
BB21 A3 08800 AND E ;TEST FOR KEY
BB22 2005 08810 JR NZ,FOUND ;JUMP IF FOUND
BB24 14 08820 INC D ;INC COLUMN COUNT
BB25 CB01 08830 RLC C ;SHIFT TEST BIT LEFT
BB27 18F7 08840 JR NXCOT ;TEST NEXT COLUMN
BB29 3A80F8 08850 FOUND LD A,(0FB00H) ;GET SHIFT BIT
BB2C 47 08860 LD B,A ;SAVE IN B
BB2D 7A 08870 LD A,D ;A=(ROW*8)+COL
BB2E C640 08880 ADD A,40H ;ADD 40H (FOR ASCII LETS)
BB30 FE60 08890 CP 60H ;TEST FOR NON-LETTER
BB32 3018 08900 JR NC,NONLET ;JUMP IF NON-LETTER
BB34 57 08910 LD D,A ;SAVE CHAR IN D
BB35 3A40FB 08920 LD A,(0FB40H) ;GET ROW & BITS
BB38 E610 08930 AND 10H ;TEST FOR DOWN ARROW
BB3A 2809 08940 JR NZ,CONTRL ;IF SO CONVERT TO CONTROL
BB3C 3A70BC 08950 LD A,(SHFLOC) ;GET SHIFT LOCK STATUS
BB3F 88 08960 CP B ;DOES IT MATCH SHIFT KEY?
BB40 283E 08970 JR Z,KDEL ;IF SO GO TO DELAY ROUT
BB42 7A 08980 LD A,D ;RETRIEVE CHAR
BB43 C620 08990 ADD A,20H ;CONVERT TO LOWER CASE
BB45 1838 09000 JR KDELAY ;GO TO DELAY ROUT
BB47 7A 09010 CONTRL LD A,D ;RETRIEVE CHAR
BB48 0640 09020 SUB 40H ;TURN INTO CONTROL CHAR
BB4A 1833 09030 JR KDELAY ;GO TO DELAY ROUTINE
BB4C 0670 09040 NONLET SUB 70H ;TEST FOR LAST ROW
BB4E 3021 09050 JR NC,LASTRW ;JUMP IF LAST ROW
BB50 C640 09060 ADD A,40H ;ADJUST FOR ROWS 4,5
BB52 2140FB 09070 LD HL,(0FB40H) ;CHECK FOR CTRL
BB55 CB66 09080 BIT 4,(HL)
BB57 208E 09090 JR NZ,EXCONT
BB59 FE3C 09100 CP 3CH ;IF CHAR < = ?
BB5B 3082 09110 JR C,NON1
BB5D EE10 09120 XOR 10H ;THEN TOGGLE SHIFT
BB5F CB08 09130 NON1 RRC B ;GET SHIFT BIT
BB61 301C 09140 JR NC,KDELAY ;JUMP IF NO SHIFT
BB63 EE10 09150 XOR 10H ;ADJUST ASCII
BB65 1818 09160 JR KDELAY ;GO TO DELAY ROUTINE
BB68 1600 09180 LD E,A ;GENERATE POINTER
BB6A 219EBB 09190 LD D,0
BB6D 19 09200 LD HL,CONTAB-30H ;RELATIVE TABLE BASE
BB6E 7E 09210 LD A,(HL) ;GET CHAR FROM TABLE
BB6F 180E 09220 JR KDELAY
BB71 07 09230 LASTRW RLCA ;A=(ROW*8+COL-48)*2
BB72 CB08 09240 RRC B ;GET SHIFT BIT
BB74 3001 09250 JR NC,LAST1 ;JUMP IF NO SHIFT
BB76 3C 09260 INC A ;A=COL*2 + 1
BB77 218E9B 09270 LAST1 LD HL,KEYTAB ;POINT TO KEY TABLE
BB7A 4F 09280 LD C,A ;GET DISPLACEMENT
BB7B 0600 09290 LD B,0
BB7D 09 09300 ADD HL,BC ;COMPUTE TABLE POSITION
BB7E 7E 09310 LD A,(HL) ;GET ASCII CODE
BB7F 57 09320 KDELAY LD D,A ;SAVE CHARACTER
BB80 010015 09330 KDEL LD BC,1500H ;LOAD DELAY COUNT
BB83 08 09340 KLOOP DEC BC ;DEC DELAY COUNTER
BB84 78 09350 LD A,B ;SEE IF ZERO
BB85 31 09360 OR C
BB86 20FB 09370 JR NZ,KLOOP ;LOOP IF NOT
BB88 7A 09380 LD A,D ;RETRIEVE CHARACTER
BB89 327FBC 09390 LD (LSTKEY),A ;SAVE IT
BB8C FEFF 09400 CP 0FFH ;SHIFT LOCK CHAR?
BB8E C0 09410 RET NZ ;RETURN IF NOT
BB8F 3A70BC 09420 LD A,(SHFLOC) ;GET SHIFT LOCK STATUS
BB92 F2 09430 CPL ;COMPLEMENT IT
BB93 E601 09440 AND 1 ;MASK IT
BB95 3270BC 09450 LD (SHFLOC),A ;SAVE IT
BB98 3E00 09460 LD A,0 ;ZERO ACC
BB9A C9 09470 RET ;RETURN
09480 ; DATA TERMINAL MODE DRIVER
BB9B 216CBF 09490 TERMIN LD HL,MESS4
BB9E C0B9BA 09500 CALL SDISP ;DISPLAY MESSAGE 4
BBA1 C0C4BA 09510 TERMI CALL KEY ;SCAN KEYBOARD
BBA4 E67F 09520 AND 7FH
BBA6 2807 09530 JR Z,TERMO
BBAB FE05 09540 CP 5 ;CTRL-E PRESSED?
BBAA C8 09550 RET Z ;EXIT IF SO
BBAB 4F 09560 LD C,A
BBAC C0E386 09570 CALL RS2320 ;ELSE SEND CHAR
BBAF DBEA 09580 TERMO IN A,(RSSTAT) ;GET RS232 STATUS
BBB1 C87F 09590 BIT 7,A ;CHAR RECEIVED?
BBB3 28EC 09600 JR Z,TERMI ;LOOP IF NOT
BBB5 DBE9 09610 IN A,(RSDATA) ;GET CHAR
BBB7 E67F 09620 AND 7FH ;RESET HIGH BIT
BBB9 C0F7B9 09630 CALL DISP ;DISPLAY CHARACTER
BBBC 18E3 09640 JR TERMI ;AND LOOP
BBBE 0000 09650 KEYTAB DEFW 0000H ;ENTER
BBC0 7FFF 09660 DEFW 0FF7FH ;CLEAR
BBC2 0303 09670 DEFW 0303H ;BREAK
BBC4 5E18 09680 DEFW 185EH ;UP ARROW
BBC6 0000 09690 DEFW 0000H ;DOWN ARROW
BBC8 0818 09700 DEFW 1808H ;LEFT ARROW
BBCA 0919 09710 DEFW 1909H ;RIGHT ARROW
BBCC 2020 09720 DEFW 2020H ;SPACE BAR
BBCE 18 09730 CONTAB DEFB 18H ;CTRL 0
BBCF 1C 09740 DEFB 1CH ;CTRL 1
BBD0 1D 09750 DEFB 1DH ;CTRL 2
BBD1 1E 09760 DEFB 1EH ;CTRL 3
BBD2 1F 09770 DEFB 1FH ;CTRL 4
BBD3 7C 09780 DEFB 7CH ;CTRL 5
BBD4 7E 09790 DEFB 7EH ;CTRL 6
BBD5 7F 09800 DEFB 7FH ;CTRL 7
BBD6 5B 09810 DEFB 5BH ;CTRL 8
BBD7 5D 09820 DEFB 5DH ;CTRL 9
BBD8 00 09830 DEFB 0 ;CTRL :
BBD9 00 09840 DEFB 0 ;CTRL ;
BBDA 7B 09850 DEFB 7BH ;CTRL <
BBDB 5F 09860 DEFB 5FH ;CTRL =
BBDC 7D 09870 DEFB 7DH ;CTRL >

```

| | | | | |
|-----------|---|--------|-----------------------------|---|
| BB0D 5C | 09800 | DEFB | 5CH | :CTRL ? |
| BB0E B7 | 09800 | CHRTAB | DEFB | 183 ;LEFT SQ BRACKET |
| BB0F 89 | 09900 | DEFB | 89H | :BACKSLASH |
| BB00 BB | 09910 | DEFB | 187 | :RIGHT SQ BRACKET |
| BBE1 58 | 09920 | DEFB | 5BH | :CARET |
| | 09930 ; | | | SECTOR TRANSLATION TABLE |
| BBE2 0001 | 09940 | SECTN1 | DEFW | 0100H |
| BBE4 0405 | 09950 | DEFW | 0504H | |
| BBE6 0009 | 09960 | DEFW | 0900H | |
| BBE8 0C00 | 09970 | DEFW | 000CH | |
| BBEA 1011 | 09980 | DEFW | 1110H | |
| BBEC 0203 | 09990 | DEFW | 0302H | |
| BBEE 0607 | 10000 | DEFW | 0706H | |
| BBF0 0A0B | 10010 | DEFW | 080AH | |
| 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 | 0900H | |
| BBFC 0C00 | 10070 | DEFW | 000CH | |
| 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 | 080AH | |
| 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 00 | 10300 | DEFB | 00H ;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 00 | 10420 | DEFB | 00H ;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 E2B8 | 10480 | DCB1 | DEFW | SECTN1 ;SEC TABLE POINTER |
| BC3B 0000 | 10490 | DEFW | 0 | ;ZEROS |
| BC3D 0000 | 10500 | DEFW | 0 | |
| BC3F 0000 | 10510 | DEFW | 0 | |
| BC41 85B0 | 10520 | DEFW | DIRBUF | ;DIRECTORY BUFF ADD |
| BC43 1ABC | 10530 | DEFW | DPB1 | ;DISK PARAM BLOCK ADD |
| BC45 00BE | 10540 | DEFW | DCS1 | ;DIR CHECK SCRATCH 1 |
| BC47 30BE | 10550 | DEFW | AVA1 | ;ALLOC VECT AREA 1 |
| BC49 E2B8 | 10560 | DCB2 | DEFW | SECTN1 ;SEC TABLE POINTER |
| BC4B 0000 | 10570 | DEFW | 0 | ;ZEROS |
| BC4D 0000 | 10580 | DEFW | 0 | |
| BC4F 0000 | 10590 | DEFW | 0 | |
| BC51 85B0 | 10600 | DEFW | DIRBUF | ;DIRECTORY BUFF ADD |
| BC53 1ABC | 10610 | DEFW | DPB1 | ;DISK PARAM BLOCK ADD |
| BC55 10BE | 10620 | DEFW | DCS2 | ;DIR CHECK SCRATCH 2 |
| BC57 5CBE | 10630 | DEFW | AVA2 | ;ALLOC VECT AREA 2 |
| BC59 E2B8 | 10640 | DCB3 | DEFW | SECTN1 ;SEC TABLE POINTER |
| BC5B 0000 | 10650 | DEFW | 0 | ;ZEROS |
| BC5D 0000 | 10660 | DEFW | 0 | |
| BC5F 0000 | 10670 | DEFW | 0 | |
| BC61 85B0 | 10680 | DEFW | DIRBUF | ;DIRECTORY BUFF ADD |
| BC63 1ABC | 10690 | DEFW | DPB1 | ;DISK PARAM BLOCK ADD |
| BC65 20BE | 10700 | DEFW | DCS3 | ;DIR CHECK SCRATCH 3 |
| BC67 78BE | 10710 | DEFW | AVA3 | ;ALLOC VECT AREA 3 |
| | 10720 ; | | | VARIABLE STORAGE AREA |
| BC69 0000 | 10730 | DMAADR | DEFW | 00H ;DMA ADDRESS |
| BC6B 00 | 10740 | DRVCOD | DEFB | 0 ;DRIVE SELECT CODE |
| BC6C 00 | 10750 | LSTCOD | DEFB | 0 ;LAST DRIVE SELECT CODE |
| BC6D 00FC | 10760 | CURPOS | DEFW | 0FC00H ;CURSOR POSITION |
| BC6F 00 | 10770 | CHAR | DEFB | 0 ;CHARACTER BUFFER |
| BC70 01 | 10780 | SHFLC | 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 |
| 0000 | 11000 | DIRBUF | DEFS | 128 ;DIRECTORY BUFFER |
| 0000 | 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 40 20 56 45 52 53 20 32 2E 32 41 20 34 38 | | | |
| | 4B | | | |
| BEAD 000A | 11110 | DEFW | 0A00H | |
| 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 000A | 11130 | DEFW | 0A00H | |
| BEC4 44 | 11140 | DEFM | | 'DIGITAL RESEARCH' |
| | 49 47 49 54 41 4C 20 52 45 53 45 41 52 43 48 | | | |
| BED4 000A | 11150 | DEFW | 0A0AH | |
| BED6 000A | 11160 | DEFW | 0A00H | |
| BED8 00 | 11170 | DEFB | 0 | |
| BED9 1C1F | 11180 | MESS2 | DEFW | 1F1CH |
| BE0B 54 | 11190 | DEFM | | 'TRS-80 CP/M BIOS 48K VER 3.3' |
| | 52 53 20 30 30 20 43 50 2F 40 20 42 49 4F 53 20 | | | |
| | 34 38 48 20 56 45 52 20 33 2E 33 | | | |
| BEF7 00 | 11200 | DEFB | 00H | |
| 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 0000 | 11220 | DEFW | 0000H | |
| BF16 0000 | 11230 | DEFW | 0000H | |
| BF18 49 | 11240 | DEFM | | 'INSERT CP/M SYSTEM DISK IN DRIVE 0' |
| | 4E 53 45 52 54 20 43 50 2F 40 20 53 59 53 54 45 | | | |
| | 40 20 44 49 53 48 20 49 4E 20 44 52 49 56 45 20 | | | |
| | 30 | | | |
| BF3A 00 | 11250 | DEFB | 00H | |
| BF3B 54 | 11260 | DEFM | | 'THEN SWITCH MAPPING TO CP/M MODE.' |
| | 48 45 4E 20 53 57 49 54 43 48 20 40 41 50 50 49 | | | |
| | 4E 47 20 54 4F 20 43 50 2F 40 20 40 4F 44 45 2E | | | |
| BF5C 00 | 11270 | DEFB | 0 | |
| BF5D 000A | 11280 | MESS3 | DEFW | 0A00H |
| BF5F 44 | 11290 | DEFM | | 'DISK OFFLINE' |
| | 49 53 48 20 4F 46 46 4C 49 4E 45 | | | |
| BF68 00 | 11300 | DEFB | 0 | |
| BF6C 000A | 11310 | MESS4 | DEFW | 0A00H |

```

BF6E 44      11320 DEFM 'DATA TERMINAL MODE - PRESS CTRL-E TO EXIT'
41 54 41 20 54 45 52 40 49 4E 41 4C 20 40 4F 44
45 20 20 20 50 52 45 53 20 43 54 52 4C 20 45
20 54 4F 20 45 58 49 54
BF77 000A    11330 DEFW 0A00H
BF99 00      11340 DEFB 0
BF9A 000A    11350 MESS5 DEFW 0A00H
BF9C 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 40
BFB9 000A    11370 DEFW 0A00H
BFBD 00      11380 DEFB 0
B53A        11390 END START ;END OF BIOS
00000 TOTAL ERRORS

```

```

00000 TOTAL ERRORS
START 0000

```

```

ALLOC B703 AVA1 BE3D AVA2 BE5C AVA3 BE7B BACKSP BABC
BDOS A706 BELL BA97 BELL1 BA9A BELL2 BA9C BELL3 BAA4
BLINIT 8889 BLKSIZ 0000 BOOT B568 CARRET BA70 COISK 0004
CHAR BC6F CHECK B601 CHECKL B613 CHKUNA B792 CHRTAB B0DE
CKEND BC30 CKSUM BC7B CLEAR BA5A CONIDL B652 CONIN B655
CONOUT B691 CONROD B64F CONST B634 CONT BA40 CONTAB BBCE
CONTRL BB47 CPMB 9F00 CPMSPB 0010 CPMSPT BC73 CURD09 BC79
CURDEL BA78 CURPOS BC6D CURSET BA16 DATREG F7EF DCB1 BC39
DCB2 BC49 DCB3 BC59 DCS1 BE00 DCS2 BE10 DCS3 BE20
DCSREG F7EC DELAY B9D3 DENPT BC77 DENTAB BC82 DIRBUF B0B5
DISKRW 88B2 DISP B9F7 DMAADR BC69 DOS 402D DOUBLE B960
DPB1 BC1A DPB2 BC29 DRVCOD BC68 DRVREG F7E0 DSKCOM BC74
DSKRDY B994 DSP 0033 DSPCHR BA14 ERCODE BC76 ERFLAG 0040
ERMASK BC72 EXCONT BB67 FILHST BB23 FOUND BB29 GOCPM B5C2
HOME B701 HOME1 B707 HOMEW B70F HSTACT 0047 HSTBLK 0002
HSTBUF BC85 HSTOSK 0043 HSTSEC 0045 HSTSIZ 0100 HSTTRK 0044
HSTWRT 0040 L0BYTE 0003 X0D 002B KDEL B8B0 KDELAY B97F
KEY BAC4 KEY1 BAEF KEYACT BC7E KEYBUF BE05 KEYF B670
KEYTAB B88E KLOOP BB83 LAST1 BB77 LASTRW BB71 LINEFD B0B2
LIST B6A5 LISTST B9E1 LOAD1 B580 LOGSEC BC7C LOOP B9D6
LSTCOD BC6C LSTKEY BC7F LSTSEC BC70 MAPST B547 MATCH BB47
MESS1 BE9A MESS2 BE09 MESS3 BF5D MESS4 BF6C MESS5 BF9A
MOTON B986 NOISKS 0003 NOCHNG B886 NDMTCH B81C NONI B85F
NONLET B84C NOOVF B7CB NSECTS 002C NU B539 NXTCOL B820
NXTR0W BAF7 ORIGIN B500 PRESS B80E PRINT B6B8 PRTRCHR B6CC
PRTLIN B6CA PRTREG F7EB PRTSCR B6C7 PUNCH B6D0 RDHST B892
READ B761 READER B6E0 READOP 004F READY B9C8 REPEAT BADD
RETRY B8CA RPTLAG BC00 RPTRAT BC81 RS2321 B6ED RS2320 B6E3
RSBAUD 00E9 RSCTRL 00E8 RSDATA 00EB RSFLAG 004E RSINIT B6F6
RSSTAT 00EA RWCOMP B930 RWERR B98E RWMOVE B86F RWOPER B7DC
SCROLL BA21 SDELAY B9DE SDISP B8B9 SDSP B8B0 SECMSK 0001
SECREG F7EE SECTN1 B8E2 SECTN2 B8F6 SECTRN B9EC SEEK B735
SEEKW B749 SEKDSK 0040 SEKHSK 0046 SEKSEC 0042 SEKTRK 0041
SELL B998 SEL2 B99E SELDSK B717 SETDEN B966 SETDMA B75C
SETSEC B757 SETTRK B730 SHFLOC BC70 SINGLE B958 START B53A
TERM: B8A1 TERMIN B898 TERMO B8AF TRKREG F7ED TRNFER B90B
TRYCNT BC75 UNACNT 0049 UNADSK 004A UNASEC 004C UNATRK 004B
VERCHK B625 WAIT1 B8FA WAIT2 B905 WAIT3 B90E WAITK B664
WBOOT B573 WBOOTE B503 WRALL 0000 WRDIR 0001 WRITE B76E
WRTHST B8A2 WRTPYE BC71 WRUAL 0002

```

```

00100 ; * * * CPMTST2/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 ;
00220 ORG 0C000H ; ADDR 0000H IN CP/M
00230 JP CHKSUM
00240 ORG 0000H
00250 CHKSUM DI
00260 LD HL,0FC00H ; CLEAR SCREEN
00270 LD DE,0FC01H
00280 LD BC,1023
00290 LD (HL),' '
00300 LDIR
00310 LOOP: LD HL,0 ; GET 16 BIT CKSUM
00320 LD DE,-0C000H
00330 XOR A
00340 LD B,A
00350 LOOP: ADD A,(HL)
00360 JR NC,SKIP
00370 INC B
00380 SKIP: INC HL
00390 INC E
00400 JR NZ,LOOP
00410 INC D
00420 JR NZ,LOOP
00430 LD L,A ; DISPLAY CKSUM
00440 LD H,B
00450 LD DE,0FC00H
00460 LD B,4
00470 V XOR A
00480 ADD HL,HL
00490 ADC A,A
00500 ADD HL,HL
00510 ADC A,A
00520 ADD HL,HL
00530 ADC A,A
00540 ADD HL,HL
00550 ADC A,A
00560 DAA
00570 CP 10
00580 SBC A,'0'-1
00590 LD (DE),A
00600 INC DE
00610 DJNZ V
00620 JR LOOP1
00630 END CHKSUM
00000 TOTAL ERRORS

```

```

00100 ; * * * CPMTST1/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 ;

```

```

0000 00200 ORG 0000H
0000 F3 00210 START DI
0001 3E20 00220 LD A,' ' ; TURN 'C' & 'T' OFF
0003 32003C 00230 LD (3C00H),A ; CLEAR POS (0,0)
0006 3200FC 00240 LD (0FC00H),A ; & POS (0,0) OF SCREEN.
0009 10FE 00250 DJNZ $ ; SHORT DELAY
000B 3E54 00260 LD A,'T' ; FLASH 'T' AT POS (0,0)
000D 32003C 00270 LD (3C00H),A ; OF TRS-B0 SCREEN.
0010 3E43 00280 LD A,'C' ; FLASH 'C' AT POS (0,0)
0012 3200FC 00290 LD (0FC00H),A ; OF CP/M SCREEN.
0015 10FE 00300 DJNZ $ ; DELAY AGAIN.
0017 10E7 00310 JR START ; ENDLESS LOOP.
0000 00320 END START

```

```

CHKSUM 0000 LOOP B016 LOOP1 B00E SKIP B01A V B020

```

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

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 desired. 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,IF=00:F00,IF=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 from DOS would be to put your commands in a patch file and have 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 period. (Special characters in a comment line, including semicolons, 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 you 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 few 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

F09,EF=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.

64K

\$59.95 PPD

INSTALLED IN KEYBOARD

TRS-80* Model I-LII

Send us your Keyboard and we will convert it to full 64K memory (48K RAM). Improved performance with or without Interface. 90 day warranty. Satisfaction guaranteed. Quick return. Free return freight within U.S.A.

ICE

International Carbide & Engineering, Inc.

100 Mill St. • P.O. Box 216

Drakes Branch, VA 23937

(800) 424-3311

(804) 568-3311 TWX: 910-997-8341

*TM TANDY CORP.

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.

EDITORIAL

DECKER, JACK

CALL FOR INFORMATION

We wanted info on Model 4 BASIC routines (we still do!) 5/3

DECKER, JACK

NEWDOS/80 ENHANCEMENTS WANTED

Zaps and patches we'd like to see (we're still waiting!) 5/12

DECKER, JACK

THE FUTURE OF COMPUTER

TELECOMMUNICATIONS

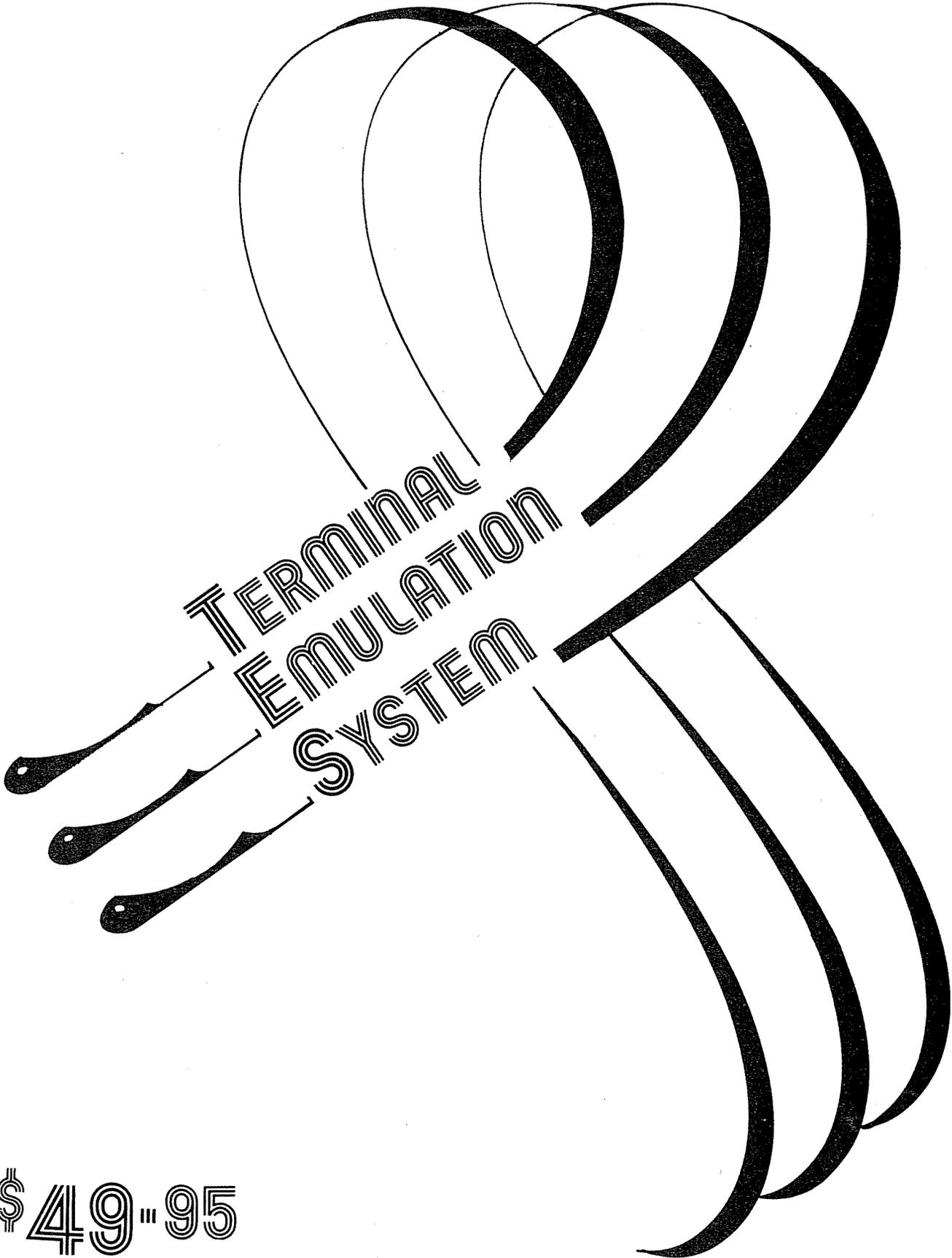
Proposal for cost savings by information utilities 4/13

| | | |
|-------------------------|--|--------|
| 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 | 5/6 |
| 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, EEOON | 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 | 2/22 |
| 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 | 1/9 |
| 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" | 4/10 |
| SALSBUARY, 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. | SCRPRF FAST SCREEN PRINT PROGRAM | | |
| Use with Models III/4 and certain Radio Shack printers | | 4/15 | |
| DADASHZADEH/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 | |
| DOMIGAN, 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 | |
| KELJZER, 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 | |
| SALSBUURY, 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 | |
| SPENCER, 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 | |
| <u>TIPS AND TECHNIQUES</u> | | | |
| 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 SuperScipsit 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 SuperScipsit 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) | | |
| SuperScipsit 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 <u>not</u> "Hayes Compatible"?! | | 6/16 | |

| | | |
|--|---|--------|
| THAYER, RICH * | TRS-80 AMATEUR RADIO NETS | |
| | "Ham" nets devoted to the TRS-80 on Sunday afternoons | 1/13 |
| TRIMBLE, FLOYD C. | LETTER TO THE EDITOR | |
| | Tip regarding use of NEWDOS/80 TI flag with LNW doubler | 6/4 |
| TRS-80 SYSTEM 80 C/G CONVERTING A CALL TO JR | | |
| | Make Model I/III code with JPs, CALLs fully relocatable | 7/19 |
| VARIOUS SOURCES | MORE ON AUTO-BOOTING A MODEL 4P | |
| | Including how to make a double-sided auto-booting disk | 3/13 |
| WEST, ALF | NEWDOS ZAP NUMBER 89 | |
| | You're asking for trouble if you apply it - here's why | 7/22 |
| WEST, ALF | USE OF CMD"J" - NEWDOS/80 VERSION 2 | |
| | Why and how to use CMD"J", includes demo program | 7/17 |
| | ZAPS FOR CMD"J" | 7/18 |
| YATES, LEONARD | MODEL III SELF-BOOTING DISK USING NEWDOS/80 | |
| | Joachim Kelterbaum's techniques applied to Model III | 1/3 |
| | THE EXTERMINATOR | 3/2 |
| ZAJAC, RON | STANDARD SORT ROUTINES | |
| | Heapsort and Shell Sort routines, with demo programs | 5/5 |
| | THE EXTERMINATOR | 8/2 |
| | LETTER TO THE EDITOR | 8/2 |
| ZAPS AND PATCHES | | |
| BIRKS, REG | MORE PATCHES TO TRSDOS | |
| | Zaps let characters other than "/" delimit date entry | 8/17 |
| BOWER, DAVE | DOSPLUS MODELA/III LOAD PATCH | |
| | Make DOSPLUS 3.5 self-booting on the Model 4P | 1/4 |
| | MORE ON AUTO-BOOTING A MODEL 4P | 3/13 |
| BRATE, DON * | TRSDOS 1.3 PATCH PROGRAM | |
| | Installs your choice of zaps to TRSDOS 1.3 | 5/6/19 |
| | THE EXTERMINATOR | 5/8/1 |
| | THE EXTERMINATOR | 2/2 |
| BROTHERS, HARDIN * | A TIP FOR MODEL 4 & 4P HACKERS | |
| | Patch TRSDOS 6.2 to enable the "KILL" command | 1/4 |
| BRUMLEY, BOB | ZAPS FOR USERS OF VIDEO4 WITH NEWDOS/80 | |
| | Zaps to DISASSEM, EDTASM, and the directory display | 3/18 |
| CARLYLE, CRAIG | LETTER TO THE EDITOR | |
| | Patch to Model I version of M-ZAL release 3 | 1/2 |
| CHRISTCHURCH-80. | ZAP TO MODEL I VERSION OF ALAN JOHNSTONE'S NEWDOS/80 MODS | |
| | Permits zero bytes to be sent to the printer | 8/22 |
| CINTUG NEWSLETTER | OPTIONAL NEWDOS/80 VERSION 2.0 ZAPS | |
| | Zaps for PDRIVE, Series I EDTASM, and FORTRAN pkg | 5/6/15 |
| | THE EXTERMINATOR | 1/2 |
| DECKER, JACK | MORE CHANGES TO SETDATE | |
| | For non-U.S. readers. Used with changes in Vol. 6, #2 | 3/10 |
| DECKER, JACK | OBERON READER PATCH FOR MODEM-80 | |
| | Required patch to use the Oberon Reader with MODEM-80 | 5/20 |
| DECKER, JACK | PATCHES TO CHANGE CERTAIN CHARACTERISTICS OF TASMOM | |
| | Three patches that are very handy if you need them | 2/17 |
| DECKER, JACK | ZAP FOR ALLWRITE! TEXT EDITOR | |
| | To keep it from dropping spaces from your text | 7/16 |
| | THE EXTERMINATOR | */* |
| DELAHAYE, H. | TIP FOR VIDEO-GENIE [ALSO PMC-80, SYSTEM-80] | |
| | Zap Tandy EDTASM program for proper printer operation | 1/13 |
| | Also accidentally printed again (we goofed!) | 4/16 |
| DOMIGAN, TONY | LETTER TO THE EDITOR | |
| | Zaps to NEWDOS/80 for use with VIDEO4 | 2/2 |
| DOMIGAN, TONY | LETTER TO THE EDITOR | |
| | Zap to NEWDOS/80 to make CLEAR command honor HIMEM | 5/1 |
| | MODEL I NEWDOS/80 "CLEAR" COMMAND ZAPS | 6/8 |
| DOMIGAN, TONY | MAKING AN AUTO-BOOT-LOADING NEWDOS/80 V2.0 SYSTEM DISK FOR THE MODEL 4P | |
| | So you don't have to load MODELA/III file separately | 5/7/14 |
| | LETTER TO THE EDITOR | 2/2 |
| | LETTER TO THE EDITOR | 2/3 |
| | MORE ON AUTO-BOOTING A MODEL 4P | 3/13 |
| | SELF BOOTING NEWDOS/80 SYSTEM DISK AGAIN | 5/10 |
| | THE TRUTH ABOUT RELATIVE TRACKS | 5/11 |
| | SELF BOOTING NEWDOS/80 SYSTEM DISK FINAL | 6/16 |
| DOMIGAN, TONY | NEWDOS/80 VERSION 2 ZAPS | |
| | Various zaps for Models I/III/4 | 5/4/18 |
| | NEWDOS/80 TIME/DATE PATCH | 4/1 |
| DOMIGAN, TONY | PATCH FOR TMDD/CMD | |
| | Fixes a minor bug in The Memory Disk Driver program | 2/5 |
| DOMIGAN, TONY | SZAP80/CMD | |
| | Patch to SUPERZAP for use with VIDEO4, similar program | 4/19 |
| EREAUT, PHIL | PRINTOUT FROM EDITOR/ASSEMBLER-PLUS | |
| | Zap eliminates title, page # for continuous printing | 5/22 |
| FRANSEN, PAUL | NEWDOS/80 TIME/DATE PATCH | |
| | Change the MM/DD/YY storage format to DD/MM/YY | 4/1 |
| FRANSEN, PAUL * | ZAPS | |
| | Zaps to various programs from the country of Holland | 4/14 |
| | LETTER TO THE EDITOR | 6/5 |
| GIELEN, B. | MODIFICATION TO CHAINBLD/BAS | |
| | Get hardcopy printouts of your NEWDOS/80 chain files | 7/22 |
| HALLGREN, JOHN | TRS-80 TIDBITS, TRASH, TREASURE, AND TRIVIA | |
| | Zaps for TRSDOS v. 1.3, 2.7DD, 2.8, & NEWDOS/80 v.1.0 | 5/4/3 |
| | THE EXTERMINATOR | 1/2 |
| | LETTER TO THE EDITOR | 3/3 |
| HUFFMAN, WILLIAM C. | LETTER TO THE EDITOR | |
| | Zap unprotects BASIC programs saved with the ".P" option | 4/3 |
| KELTERBAUM, JOACHIM | SOUP UP GOOD OL' SCRIPSIT | |
| | Add calculator features to Scripsit's text editor | 4/4 |
| KNAGGS, P. | LETTER TO THE EDITOR | |
| | SuperScripsit zaps remove test for two adjacent spaces | 5/3 |
| LEVINSON, ANDY | ANOTHER TRSDOS 1.3 PATCH | |
| | Prevents long AUTO commands from crashing the system | 6/22 |
| LEWIS, L. & EVANS, R. | MODEL 4 LDOS/TRSDOS 6.2 DATE PATCH | |
| | Changes format of dates from MM/DD/YY to DD/MM/YY | 7/19 |
| MATUG NEWSLETTER | MORE MODEL 4 PATCHES | |
| | Patches to TRSDOS 6.2 and the M-ZAL assembler | 8/22 |
| O'HARE, KEVIN | ZAPS FOR CMD"J" | |
| | Patches to NEWDOS/80 BASIC to let CMD"J" use DD/MM/YY | 7/18 |
| PHILLIPP, JOHN T. | THE TRUTH ABOUT RELATIVE TRACKS | |
| | More on self-booting NEWDOS/80 disks on the Model 4P | 5/11 |
| RASMUSSEN, ART | SELF BOOTING NEWDOS/80 system disk for the MODEL 4P.....AGAIN | |
| | See next entry in this index for the final version | 5/10 |
| RASMUSSEN, ART | SELF BOOTING NEWDOS/80 system disk for the MODEL 4P.....FINAL | |
| | A better way to create a self-booting NEWDOS/80 disk | 6/16 |
| | See listing under DOMIGAN, TONY for related articles. | |
| ROHDE, ARNE | CONVERTING COPYCAT3 INTO /CMD FORMAT | |
| | Also explains the COPYCAT3 protection scheme | 7/13 |
| ROSSER, ERROL | RANDOM I/O | |
| | Mods to Kelterbaum's JKL program; LDOS/NEWDOS patches | 1/10 |
| SALSBURY, NATE | A PATCH OF A PATCH | |
| | Convert cassette EDTASM+ to run on Mod III disk | 5/4/9 |
| | LETTER TO THE EDITOR | 5/6/4 |
| | A PATCH OF A PATCH OF A PATCH | 3/18 |
| SALSBURY, NATE | LETTER TO THE EDITOR | |
| | Better regulation of stepping rate in Model 4 TASMOM | 4/3 |
| SEABORN, BOB | ADAPT MODEL III INFOCOM GAMES TO RUN ON THE MODEL I | |
| | A simple zap makes it possible | 8/22 |
| SEABORN, BOB | MODEL I NEWDOS/80 "CLEAR" COMMAND ZAPS | |
| | Make CLEAR respect HIMEM. Model III zaps also | 6/8 |
| 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 |
| SMITH, HERBERT L. | LETTER TO THE EDITOR | |
| | Patches to Radio Shack CP/M 3.0+ to correct date bugs | 6/5 |
| SMITH, HERBERT L. | SUPER UTILITY+ 3.2 AND THE MODEL 4 | |
| | Patches you may need to run SU+ on a Model 4 | 5/14 |
| SOKOLOWSKI, KAZ | LETTER TO THE EDITOR | |
| | Patch changes track step rate of TRSDOS 6 boot sector | 3/4 |
| SPENCER, GIL * | CHANGE SUPERSCRIPSIT'S CURSOR CHARACTER | |
| | Also where to find patch to eliminate delta symbol | 1/7 |
| SPENCER, GIL | REWRITE OF NEWDOS/80 DISK BASIC AMPERSAND (&) ROUTINE | |
| | Change the default from octal to hexadecimal | 6/5 |
| | (This article appeared with an incorrect listing in 4/22) | |
| | LETTER TO THE EDITOR | 6/5 |
| SWEETMAN/SKARSHOLT | MAKE TRSDOS 1.3 YOUR OYSTER | |
| | Creates directory entries for TRSDOS 1.3 system files | 6/15 |
| THOMPSON, BRENDON | ELECTRIC PENCIL ZAPS | |
| | One mandatory from IJG, and one for NEWDOS/80 users | 6/6 |
| VARIOUS SOURCES | PATCHES FOR MODEL 4 USERS | |
| | Zaps to DOS and various programs for Model 4 users | 6/13 |
| VARIOUS SOURCES | PATCHES FOR TRSDOS 6.2 | |
| | Five sets of useful zaps for TRSDOS 6.2 users | 5/13 |
| WEST, ALF | NEWDOS ZAP NUMBER 89 | |
| | You're asking for trouble if you apply it - here's why | 7/22 |
| WHITE, LAWRENCE C. | A PATCH OF A PATCH OF A PATCH | |
| | Changes to Nate Salsbury's zaps to Editor/Assembler+ | 3/18 |
| WHITTAKER, JIM | MODEL III TRSDOS 1.3 - THE BEST DOS! | |
| | A collection of TRSDOS 1.3 patches and enhancements | 8/5 |

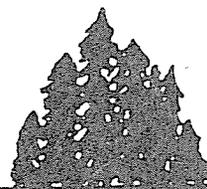


TERMINAL
EMULATION
SYSTEM

\$49.95

TRSDOS 6

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